UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MECANISMO EM NUVEM DE MONITORAMENTO A DADOS SENSÍVEIS

RAFAEL TOMÉ DE SOUZA

ORIENTADOR: PROF. DR. SERGIO DONIZETTI ZORZO

São Carlos – SP

Maio/2014

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MECANISMO EM NUVEM DE MONITORAMENTO A DADOS SENSÍVEIS

RAFAEL TOMÉ DE SOUZA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Sistemas Distribuídos e Redes.
Orientador: Prof. Dr. Sergio Donizetti Zorzo

São Carlos – SP

Maio/2014

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária da UFSCar

S729mn

Souza, Rafael Tomé de.
Mecanismo em nuvem de monitoramento a dados sensíveis / Rafael Tomé de Souza. -- São Carlos : UFSCar, 2014.
116 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2014.

1. Sistemas distribuídos. 2. Privacidade. 3. Auditoria. 4. Computação em nuvem. I. Título.

CDD: 005.43 (20ª)

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

"Mecanismo em Nuvem de Monitoramento a Dados Sensíveis"

Rafael Tomé de Souza

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação

Membros da Banca:

Prof. Dr. Sergio Donizetti Zorzo
(Orientador - DC/UFSCar)

Prof. Dr. Antonio Francisco do Prado
(DC/UFSCar)

Prof. Dr. José Viterbo Filho
(UFF)

São Carlos
Maio/2014

Aos meus pais, Antonio e Fátima, ao meu irmão, Sérgio, à minha cunhada, Camila, e às minhas avós, Olímpia e Negrinha.

AGRADECIMENTOS

Agradeço a Deus por sempre iluminar o meu caminho e por colocar pessoas incríveis que me ajudaram durante toda esta jornada.

Agradeço aos meus pais, Antonio e Fátima, ao meu irmão, Sérgio, à minha cunhada, Camila, às minhas avós, Olímpia e Negrinha, pelo apoio incondicional em todos os desafios que me proponho a enfrentar. São pessoas incríveis, com as quais compartilho as tristezas e as alegrias da minha vida.

Agradeço ao meu orientador, Prof. Dr. Sergio D. Zorzo, pela oportunidade de poder participar do corpo discente desta Universidade. Agradeço pelas orientações, pela ajuda, pela confiança, pela amizade, pela compreensão, pela motivação e pelas oportunidades.

Agradeço aos meus queridos amigos Amandia Sá, Augusto César, André Bindilatti, Felipe Giuntini, Isis Caroline, Mirela Cazzolato, Tatiana Alencar e Thiago Vieira pelas risadas e companheirismo durante esses anos de mestrado.

E, por fim, agradeço aos amigos do laboratório: Anderson Kanegae, Leonardo Leite, Paulo Avila, Roan Simões e Tiago Rosa, pelo companheirismo durante o mestrado.

Escolhe um trabalho de que gostes, e não terás que trabalhar nem um dia na tua vida.

Confúcio

RESUMO

A garantia de privacidade de um dado de uma pessoa é entendida como a capacidade desta pessoa gerenciar, armazenar, alterar, restringir ou divulgar para um grupo de indivíduos de sua escolha. O dado compartilhado pode ser sensível revelando algo de teor privado que merece uma proteção no seu compartilhamento, por exemplo a informação financeira pessoal. Nos diversos serviços de computação há muitos dados sensíveis sem qualquer mecanismo que garanta a privacidade de seus proprietários. Este trabalho apresenta um mecanismo que garante a privacidade da pessoa que tem os dados acessados, o proprietário do dado, e da pessoa que acessa o dado. Foi desenvolvido um mecanismo em nuvem de monitoramento a dados que precisam ter o acesso monitorado com cenários de detecção de intrusão disponível para o proprietário do dado. A viabilidade da proposta foi avaliada por testes de tempo de resposta do acesso à página monitorada, sobrecarga do servidor e consumo de recursos do servidor sob o prisma de uma aplicação usando o mecanismo. Tal mecanismo apresenta ser uma solução viável por ter um impacto mínimo nos recursos computacionais e uma solução que auxilia no monitoramento de acesso a dados sensíveis.

Palavras-chave: auditoria, auditoria de pseudônimo, privacidade, dados sensíveis, computação nas nuvens

ABSTRACT

The privacy guarantee of a person's data is understood as the capacity of this person to manage, store, change, restrict or disclose for groups of individual of his choice. The data shared can be sensitive, revealing private content that deserves protection in sharing, for example financial personal information. In many computing services a lot of sensitive data does not have any mechanism that guarantees the owner's privacy. This work shows a mechanism that guarantees the privacy of the person who has the data accessed, the data owner, and the privacy of the person who accesses the data. It was developed a cloud monitoring mechanism for data whose access needs to be monitored with intrusion detection scenario available for the data owner. The propose feasibility was evaluated by response time test of a monitored page access, server overload and the server resource consumption through the prism of an application using the mechanism. Such mechanism has been a viable solution due to its minimal impact in computational resources and a solution that assists in sensitive data access monitoring.

Keywords: auditing, pseudonym auditing, privacy, sensitive data, cloud computing

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

Capítulo 1

INTRODUÇÃO

O capítulo apresenta a motivação, o objetivo e a organização da proposta desenvolvida, considerando a problemática da divulgação de dados que podem revelar dados pessoais, financeiros e localização que, dentro do contexto deste trabalho, são definidos como dados sensíveis. A proposta pressupõe de que é possível conceder o acesso a dados sensíveis, garantindo a transparência do acesso sem interferir na privacidade do indivíduo.

1.1 Contexto

O surgimento de novas tecnologias na área de Tecnologia e Informação (T.I.) propulsionam melhoras nos serviços on-line e nos processos de negócio e reinventa a forma de comunicação e de troca de informações. O rápido acesso à informação e o avanço tecnológico são uma realidade presente em todas as áreas, incluindo a internet.

A área de T.I. ganha mais força com a computação na nuvem com a infraestrutura sob demanda e pela oferta de aplicações como serviço. Espera-se um crescimento da oferta de *Software as a Service* (SaaS) em comparação a pacotes tradicionais de *software*. É possível perceber a queda da receita dos tradicionais pacotes de *softwares*, já que 34% das novas compras de software empresarial serão entregues via SaaS, como citado por Walters (2012).

A facilidade no acesso a dados em sistemas computacionais torna o indivíduo exposto a risco de privacidade e as empresas a terem o seu sigilo violado, em ambos os casos dependendo do tipo de dado acessado. A necessidade de repassar dados pessoais ou confidenciais em troca do acesso a algum tipo de serviço é cada vez mais comum, e com o mundo digital isso torna muito transparente conduzindo a uma perda de controle do dado.

Segundo Ghani e Sidek (2008) o dado tem um papel importante em qualquer tipo de tran-

sação *on-line* ou *off-line*. No entanto, a maioria das pessoas não percebem o valor das suas informações pessoais. Algumas informações pessoais podem ser classificadas como sensíveis e precisa ser mantida privada, já outras informações são sensíveis, mas não tem esta necessidade. Os quatro tipos de dados são o dado pessoal, dado sensível, dado de identificação e dado anônimo, sendo apenas os três primeiros considerados privados.

O dado pessoal refere-se a qualquer dado que pode ser usado para identificar uma pessoa como o nome, endereço, número de telefone. O dado sensível refere-se a qualquer informação liberada sobre raça ou origem étnica, religião, filosófica, ou outras crenças, opiniões políticas, filiação partidária, assim como qualquer dado que revela informação histórico de saúde, raça. O dado de identificação refere-se a dado pessoal que permite uma identificação direta assim como o DNA, número de cartões. O dado anônimo refere-se a qualquer dado que não pode ser associado a nenhuma pessoa ou não identifica gênero, tipo de doença (GHANI; SIDEK, 2008).

Na visão de Gunasekera (2012) a informação pessoal é a classificação do dado conhecido pelo indivíduo e por um número limitado de pessoas dentro do círculo de amizade desta pessoa. A informação pessoal é algo privado, mas o indivíduo pode estar disposto a compartilhar com amigos próximos e familiares. A informação pessoal pode ser número de telefone, endereço e endereço de email. Já a definição de dado sensível é muito mais do que uma informação pessoal, geralmente são informações que não está disposto a compartilhar em nenhuma situação com ninguém. Os dados deste tipo de informação podem ser senhas, credenciais para acesso ao *internet banking*, número de telefone, número do seguro social, ou endereço. A perda ou o comprometimento de informações pessoais pode iniciar situações que podem incomodar o indivíduo. Já a informação sensível comprometida pode ter efeitos físicos ou danos emocionais. Este tipo de informação deve ser protegida o tempo todo seja quando estiver armazenada ou solicitada visualização pelo sistema computacional.

Já Pearson (2009) menciona que a informação pessoal é o tipo de informação que precisa ser protegida. No entanto, o termo "informação pessoal"pode ser definida de forma diferenciada por algumas pessoas. Assim, Pearson (2009) define "informação pessoal"como a privacidade da informação sensível que incluem:

(a) Informação de identificação pessoal referente a qualquer informação usada para localizar/identificar o indivíduo ou informação que pode ser correlacionada para localizar/identificar o indivíduo, i.e., nome, endereço, número de cartão de crédito, código postal e *Internet Protocol Address*;

(b) Informação sensível refere-se a informação considerada privada ou que merece algum tipo

de proteção, i.e, opção religiosa, raça, saúde, orientação sexual, filiação partidária, informação financeira pessoal, informação de desempenho no trabalho;

(c) Informação de identificação pessoal sensível relacionada a informação biométrica e imagem de câmera em locais públicos;

(d) Dados de uso (*Data usage*) relacionado a informação coletada de dispositivos computacionais, assim como impressoras e histórico de visualização de conteúdo do indivíduo aos arquivos; e

(e) Informação de identificação única liberada pelos dispositivos permitindo o rastreamento, e.g., endereço de IP, RFID, identificadores de hardware único.

Com base na definição dos autores (GHANI; SIDEK, 2008; PEARSON, 2009; GUNASEKERA, 2012) para esta dissertação o conceito que mais se aproxima com o que será trabalhado é a definição apresentada por Pearson (2009). Assim, a informação sensível, no contexto a ser tratado por esta dissertação, pode ser considerada como exemplo o nome, local de trabalho, horas de trabalho, remuneração, localização e situação financeira pessoal.

Os autores (PEARSON, 2009; GUNASEKERA, 2012) usam a definição de informação como sendo o conjunto dos dados organizados gerando uma informação. Já no contexto desta dissertação o trabalho visa focar no dado sendo que a organização deste de maneira significativa gera a informação. Ou seja, a premissa é focar no conjunto de dado que pode ser acessado pelo indivíduo e este dado representa uma informação sensível. Por isso sempre será utilizado a definição "dado sensível", pois dado é a matéria bruta que associada com outros dados e devidamente organizado geração a informação.

A contribuição deste trabalho está relacionada à proposta de um serviço de monitoramento disponibilizado na nuvem que permita ao proprietário do dado auditar os logs de acessos dos dados julgados sensíveis. Ou seja, tal mecanismo é acoplado a uma aplicação *web* e, com os *logs* de auditoria gerados, permite a análise dos logs por meio dos cenários de detecção de intrusão. Além disso, tal serviço de auditoria disponível na nuvem mantém a privacidade do usuário que acessa o dado garantindo a transparência de acesso e ao mesmo tempo a identificação no caso de mau uso como previsto na auditoria de pseudônimo (FISCHER-HÜBNER, 2001).

1.2 Motivação

As motivações para esse trabalho é decorrente de quatro análises contemporâneas em relação ao uso de dados sensíveis no contexto de aplicações a serem discutidas a seguir.

A primeira motivação refere-se à divulgação de dados realizada pelo governo federal brasileiro, que usa a Internet para promover a transparência das ações administrativas. O Portal da Transparência do governo federal disponível em `http://www.portaltransparencia.gov.br/` é um exemplo das ações de transparência do Brasil

O Portal da Transparência do governo federal existe desde 2004 e trata-se de uma iniciativa da Controladoria Geral da União (CGU) para assegurar a correta aplicação dos recursos públicos, criado em decorrência da necessidade de divulgação dos dados orçamentários e financeiros, bem como atos administrativos (JORDÃO, 2011).

O portal permite ao cidadão acompanhar de perto a forma como o governo está aplicando o dinheiro, ajudando-o na fiscalização e promovendo a fiscalização pública. Os dados presentes no portal são fornecidos pelo sistema integrado de administração financeira do governo federal e apresenta os gastos diretos do governo federal, as transferências de recursos a estados e municípios, e dos servidores do governo federal (PORTALTRANSPARENCIA.GOV.BR, 2014).

No Brasil, o acesso à informação é um direito do cidadão e um dever do Estado previsto pela Constituição brasileira, agora regulamentada pela lei federal de número 12.527 sancionada em 18 de novembro de 2011, também conhecida como Lei de Acesso à informação (JORDÃO, 2011). Além da Lei de Acesso à Informação, pode ser citado a Lei de Responsabilidade Fiscal (Lei Complementar n. 101/00) e a Lei complementar 131/09 regulamentam atos de transparência no governo.

A lei número 12.527 consolida e define um marco regulatório em relação ao acesso à informação mantida pelo Estado e estabelece processo para que a administração possa atender as solicitações. O Estado adota a posição de que os dados já pertencem aos cidadãos que podem fazer o que querem com os dados (JORDÃO, 2011).

A Lei de acesso à informação, regulamentou o acesso de qualquer cidadão brasileiro aos dados do governo. A implantação bem-sucedida da lei, que é tão comum em mais de 90 nações, produzem cidadãos informados e conscientes sobre os seus direitos e administração transparente (JORDÃO, 2011). Tal lei funciona para regulamentar de forma ampla o acesso a dados do governo e de qualquer documento ou informação específica procurado por um cidadão.

Em 2011, a ação de transparência administrativa se tornou mais forte com a Lei de Acesso à Informação (Lei n. 12.527/11). Esta lei permite o acesso às informações públicas como regra geral, e que o segredo é a exceção. Com a Lei, há a necessidade de publicação de informações de interesse geral ou coletivo, independentemente da solicitação por parte do cidadão aos órgãos governamentais. Tal ação deve ser feito por todos os meios disponíveis, incluindo a

Internet. Entre as informações que precisa ser liberado, há a necessidade de liberar registros de transferência financeira e os registros de despesa (BRASIL, 2014b).

Por se tratar de uma lei federal, a lei estende a várias áreas do governo incluindo estados e municípios. Assim, a implantação da lei de acesso à informação (Lei n. 12.527) chegou a ser entrave na Câmera e no Senado em decorrência da publicação de rendimentos mensais, nome e local de trabalho dos servidores (MILITÃO, 2012).

A liberação deste tipo de dado permite jornalistas e pesquisadores fazerem análises para compreender como o governo está usando recursos públicos, contribuindo para o processo de um governo transparente (MILITÃO, 2012). No entanto, o acesso a dados sensíveis de forma irrestrita pode colocar os servidores públicos federais em situação embaraçosa (SMITH; XU, 2011) e torná-los vulneráveis a ações de cibercrime (TAVANI; GRODZINSKY, 2002) porque os dados sensíveis estão facilmente disponível para qualquer tipo de pessoa.

Ao abordar o aspecto de acesso aos dados sensíveis é possível inferir questões de privacidade em duas perspectivas. A primeira perspectiva é a posição do governo em liberar os dados, considerando que o governo é o proprietário dos dados. E a segunda perspectiva é a do servidor público federal ou servidor público, cujos dados revelam a situação financeira, localização, horas trabalhadas por semana e assim por diante.

De acordo com os estudos (FISCHER-HÜBNER, 2001; WESTIN, 2003), é possível entender que a privacidade é o "direito de ser deixado em paz", conforme citado por Samuel D. Warren e Louis D. Brandeis (WARREN; BRANDEIS, 1890). O direito de decidir se deseja compartilhar ou fornecer alguma informação pessoal que se refere apenas à vida privada. É possível incluir nessa definição o direito do indivíduo, grupos ou instituições de determinar quando, como e que informações sobre eles podem ser conhecidas por outros.

Há alguns países em que o conceito jurídico de privacidade é estendido a grupos e instituições, como a França, Áustria e Dinamarca. Em outros, é restrito aos indivíduos, como Alemanha, EUA e Reino Unido (FISCHER-HÜBNER, 2001).

Nos Estados Unidos, a privacidade é uma preocupação constante devido à ocorrência de ações terroristas e ciberataques. Mesmo assim, a Constituição dos Estados Unidos não fornece uma definição clara do direito à privacidade, mas a constituição explica a limitação de poderes dentro do governo. Contudo, o cidadão norte-americano possue leis que abordam os direitos de privacidade dentro do governo e para os cidadãos, como a Lei Gramm-Leach-Bliley(GLBA) e *Health Insurance Portability and Accountability Act (HIPAA)* (KOWTKO, 2011).

No continente europeu, o direito à privacidade foi estabelecido pela União Europeia que

define as leis que protegem a privacidade entre os cidadãos e interesse social. A diretiva promovida em 2002 define regras e regulamentos, tais como: aprovação do consumidor para coletar dados pessoais; o direito de rever e corrigir os dados por parte dos consumidores; obrigatoriedade das empresas que recolheram informação de registrar suas atividades com o governo; o compartilhamento das informações pessoais podem ser realizadas somente com consentimento individual; e os caixas de lojas não podem pedir o número de telefone da pessoa. Em geral, a privacidade é considerada um direito humano na Europa (KOWTKO, 2011).

No Brasil, o Art. 5, inc. X da Constituição Federal de 1988 (BRASIL, 2014a) o direito à privacidade protege o cidadão em relação à privacidade. O inciso X garante a inviolabilidade da intimidade, da vida privada, da honra e da imagem das pessoas. No entanto, uma lei específica para proteger os dados ainda tramita em Brasília (ACESSOAINFORMACAO.GOV.BR, 2014; GOMES, 2014).

Em outro artigo (SMITH; XU, 2011), Smith ainda define a privacidade como a capacidade de um indivíduo ou um grupo de revelar informações de acordo com as circunstâncias, considerando que a privacidade diz respeito aos domínios físico, informativo, governamental e intelectual que se sobrepõem.

A privacidade no domínio informacional é mais correlacionado com o foco deste trabalho, que está relacionada com a divulgação de informações. A divulgação de informações traz algumas possíveis ameaças como constrangimento pessoal, danos à reputação profissional e discriminação em alguns casos, (SMITH; XU, 2011).

As leis que tratam a transparência e o acesso à informação funcionam como um controlador e libera dados aos cidadãos que requisitam. Ao mesmo tempo, o Governo confia na liberação de dados sensíveis, como as condições financeiras de seus servidores para a promoção da transparência e confiabilidade.

A ação de publicar dados poderia gerar um risco relacionado com os dados liberados como um controle paradoxo relatado em estudos (ACQUISTI; ADJERID; BRANDIMARTE, 2013; BRANDIMARTE; ACQUISTI; LOEWENSTEIN, 2013) que analisa quando alguém tem o controle de liberação de dados e perde o controle sobre os dados liberados.

Lidar com a transparência é uma questão importante para muitos países, incluindo o Brasil. Portanto, usar um serviço que permita compreender o acesso ao dado pelo cidadão conduzirá o proprietário dos dados a criar alguma política e manter algum controle no acesso. Os gastos publicados no portal apresenta dados sensíveis relacionados com a vida financeira de muitos funcionários sendo um bom estudo de caso para a aplicação do mecanismo em nuvem de mo-

nitoramento a dados sensíveis, proposto neste trabalho.

A segunda motivação é a questão do dado acessado e processado por aplicações nas nuvens sem o consentimento de uma empresa pelo empregado (WALTERS, 2012).

Walters (2012) discute a questão do *"Bring Your Own Software"* (BYOS), situação em que os empregados confiam em aplicações nas nuvens para processamento de dados corporativos sem respaldo da equipe de Tecnologia da Informação (TI) da empresa. E ainda há o uso de *"Bring Your Own Device"* (BYOD), tratado pelo autor no mesmo artigo, que diz respeito à adoção das aplicações *web* nos aparelhos dos próprios empregados, já que funcionários podem trabalhar em qualquer lugar a qualquer hora em decorrência dos novos recursos tecnológicos.

Certamente os gerentes de riscos deverão direcionar esforços para descobrir como permitir o acesso a serviços na nuvem enquanto controlam o acesso aos dados e auditam a aplicação usada. Dados corporativos que são acessados por aplicações *web* tendem a sofrer as mesmas ameaças internas e fraudes de computador em comparação com os tradicionais aplicativos corporativos (WALTERS, 2012).

A terceira motivação para esta proposta está relacionada com a sugestão de Spring (2011) de dar ao usuário a possibilidade de banir o acesso de dados sensíveis em áreas públicas, ou seja, o usuário teria a possibilidade de autorizar o que deve ser publicado ou não. O autor menciona que as aplicações em nuvens devem fornecer serviços que são idênticos ou melhores do que as que estão fora delas. Ainda relata sobre os aplicativos do Google que monitoram o comportamento de acesso, como o tempo e endereço de IP, tornam essa informação disponível para o usuário e o notificam de alguma informação anormal.

Entretanto, com o crescimento do compartilhamento de dados surgem novas formas de roubo, o cibercrime. E expor dados de qualquer tipo na Internet pode ser uma brecha para alguma ação maliciosa por alguém mal intencionado. Por exemplo, o fato de que dados sensíveis estão expostos na rede social pode ser uma brecha para o ataque de *phising*, *stalking* e *spamming* (HANNE et al., 2012). Pela combinação de dados, o criminoso pode identificar rotinas e hábitos das pessoas.

A quarta motivação é referente à nova tendência no desenvolvimento de sistemas distribuídos que vem enfrentando alguns desafios, entre eles a falta de um adequado mecanismo de segurança para proteção de dados sensíveis (HAMLEN et al., 2011). A problemática de acesso a dados sensíveis no contexto de aplicações na nuvem é um tema de pesquisa relevante, o qual merece esforços na proposição de mecanismos para aumentar a proteção de acesso a dados sensíveis (WALTERS, 2012; SPRING, 2011), já que trabalhar na nuvem oferece vantagens tais como

infraestrutura, processamento, gerenciamento de memória e custo-eficácia.

A liberação de acesso a dados sensíveis almejado pelo governo federal, a possível falta de conhecimento da empresa em relação a dados que podem ser acessados e processados em aplicações nas nuvens, a possibilidade do usuário controlar a publicação de dados sensíveis e o desafio a ser enfrentado na criação de mecanismos que visam aumentar a proteção dos dados sensíveis são motivações que fortalecem a necessidade de soluções no sentido de monitorar o acesso. Tal monitoramento pode auxiliar na criação de políticas de liberação dados por organizações ou governos.

1.3 Objetivo

O objetivo deste trabalho é propor um mecanismo em nuvem de monitoramento a dado sensível a ser implementado em duas partes. Uma parte implementada como um serviço a ser requisitado por aplicações para realizar o monitoramento do acesso a dados sensíveis. A outra parte a ser implementada como um *plugin* para interceptar os dados definidos como sensíveis das aplicações e comunicar com o serviço na nuvem. O dado sensível, no contexto deste projeto, refere-se a qualquer informação que pode revelar dados tais como situação financeira da pessoa, local de trabalho ou qualquer dado considerado privado ou que merece algum tipo de proteção.

O mecanismo tem a finalidade de monitorar o acesso por meio da geração de logs, captado pelo *plugin*, o qual permite acompanhar os acessos e a tomar decisão sobre políticas para acesso aos dados liberados pelo proprietário do dado. Os acessos são analisados pelo proprietário do dado por meio de cenários de detecção a intrusão que são disponibilizados pelo mecanismo por meio de uma interface web. Os cenários pré-configurados apontam registros dentro de regras evidenciando o possível mau uso em relação aos dados. No entanto, a identidade e a privacidade dos acessos são revelados apenas quando estritamente necessário ao proprietário do dado por meio do mecanismo.

A identidade e a privacidade são protegidos durante o acesso ao dado sensível, através de uma adaptação da auditoria de pseudônimo. Assim, com o uso do mecanismo é possível liberar dados sensíveis e garantir a integridade de quem acessa e do proprietário do dado. A privacidade de quem acessa é garantida com um pseudônimo que só pode ser revelada ao proprietário do dado e a integridade do proprietário do dado é alcançada porque o mesmo consegue identificar mais claramente quem acessa o dado.

A auditoria auxilia na identificação de problemas em sistemas computacionais, violação de sistemas e erros. No contexto deste projeto é aplicada para gerar registros do acesso realizado

pelo indivíduo para identificar possíveis comportamentos de mau uso com base em cenários de detecção previamente configurados. O sistema de auditoria em uma visão geral é composto por um coletor, analisador e notificador, como citado por (BISHOP, 2002).

Em um auditoria a identidade do usuário fica evidente, considerando que cada acesso realizado é capturado um log. Tal log com o registro do usuário pode gerar um invasão de privacidade considerando que a partir do log pode-se entender os acessos realizados pelo usuário. Assim, a identidade refere-se a uma pessoa e a vincula as ações realizadas em um sistema computacional. Já a privacidade salva o direito do indivíduo de revelar dados somente quando ele permitir (WARREN; BRANDEIS, 1890).

A identidade, a privacidade e a auditoria são conceitos fundamentais que norteiam o mecanismo proposto neste trabalho. Tal conjunto de conceitos e práticas visa ser ofertado como um serviço denominado LogCloud, cuja oferta é flexibilizada pela disponibilidade e pelo poder de armazenamento da computação na nuvem.

1.4 Organização do trabalho

A dissertação está organizada em capítulos que visam esclarecer conceitos e detalhar a proposta de trabalho a ser realizada.

No Capítulo 2 é apresentada uma discussão sobre a visão de privacidade no contexto de computação, definições de identidade e os riscos envolvidos quando há compartilhamento de dados.

No Capítulo 3 são explicados os conceitos de auditoria com garantia de privacidade, a visão da proposta do mecanismo para auditar com garantia de privacidade e a adaptação realizada na auditoria de pseudônimo, que visa balancear o conflito segurança e invasão de privacidade. Além disso, apresenta o que a computação em nuvem pode agregar ao trabalho.

No Capítulo 4 são apresentadas propostas relacionadas a este trabalho que tratam do uso da auditoria no auxílio da detecção de intrusão e aplicado na prevenção de mau uso de conteúdo. Neste capítulo, ainda são abordados as possíveis contribuições do trabalho.

O Capítulo 5 apresenta a descrição dos requisitos e a documentação da implementação do mecanismo LogCloud incluindo discussão do *core* do projeto e do *plugin* desenvolvido para comunicar com o LogCloud.

O Capítulo 6 apresenta o estudo de caso incluindo o cenário de *deploy*, os resultados de desempenho com base nos testes executados, análises e limitações do serviço desenvolvido.

Por fim, o Capítulo 7 apresenta a conclusão do trabalho com as contribuições e limitações do mecanismo.

Capítulo 2

PRIVACIDADE

O capítulo apresenta uma visão geral do conceito de privacidade e como a mesma é tratada em diversos trabalhos científicos. Outra questão abordada é o impacto que o mundo computacional infere neste campo considerando que os dados do usuário são gerenciados por aplicações computacionais. Por fim, o capítulo trata a questão diretamente relacionada com acesso a dados sensíveis e possíveis riscos deste compartilhamento.

2.1 Considerações iniciais

Para acessar um sistema ou serviço e definir quais recursos estão disponíveis para o usuário, é necessário informar a identidade (FISCHER-HÜBNER, 2001), o que resulta na identificação de quem acessa o sistema, estabelecendo a validade da identidade informada (FISCHER-HÜBNER, 2001).

A autenticação do usuário pode ser feita usando diferentes tipos de verificação que incluem o uso de alguma coisa que a pessoa sabe (senha), alguma coisa que a pessoa possui (*token*), alguma propriedade física (impressão digital) e algum traço da pessoa (assinatura) (FISCHER-HÜBNER, 2001; SHINDER; CROSS, 2008).

Ao informar a identidade para um sistema computacional, o usuário repassa para ele dados pessoais, facilitando a invasão de privacidade e a segurança. Por esse motivo, têm sido feitas pesquisas para tratar questões de privacidade, explorando-as em diversos cenários, e a forma como os sistemas computacionais lidam com o dado sensível disponibilizado por aplicações, que é o enfoque de estudo deste trabalho.

As publicações científicas pesquisadas abordam várias interpretações possíveis do tema de privacidade no ambiente computacional, revelando diversas facetas sobre a questão da privaci-

dade. Wang, Lee e Wang (1998) mostra que a privacidade é uma preocupação por parte dos usuários em relação ao marketing na Internet e Anton, Earp e Young (2010) mostra uma pesquisa da evolução sobre a preocupação da privacidade entre os anos de 2002 e 2008.

Em 2011, a privacidade se tornou foco de investigação nos serviços públicos do tipo *mashup*, disponibilizados pelo governo britânico, conforme descrito no artigo de Smith e Xu (2011).

A questão da privacidade é presente nas discussões sobre o rastreamento realizado pelos websites que coletam informações do usuário , como apresentado por Mayer e Mitchell (2012). Além disso, ações de engenharia social, tais como *phishing*, podem facilitar o acesso a dados sensíveis invadindo a privacidade do usuário, como tratado por Darwish, Zarka e Aloul (2012).

Outra questão relativa à privacidade que vem sendo abordada é a comunicação entre máquinas, como em Cheng et al. (2012). A proteção da privacidade pode ser aplicada nos aparelhos durante a comunicação, no armazenamento, no processamento, nos sistemas RFID e em serviços baseados em localização. Adicionalmente, novas tecnologias e novas formas de disponibilizar aplicações na Internet trazem novos desafios para a área científica que trabalha com a temática da privacidade.

Os serviços disponibilizados pela computação na nuvem geram questões de indefinição para o usuário em relação ao local onde o conteúdo está armazenado ou se o conteúdo salvo na nuvem está livre de acessos indevidos, visto que está sob a guarda de terceiros.

A computação na nuvem fornece um ambiente de desenvolvimento, *deploy* e execução de aplicações que podem crescer rapidamente em escalabilidade, performance e confiança. Essas características são decorrentes da premissa de que os serviços na nuvem são protegidos de falhas e livram o usuário de preocupações com infraestrutura (ZHOU et al., 2010).

Zhou et al. (2010) explica que, para atingir o nível de segurança adequado, há necessidade de cumprir cinco metas: disponibilidade, confidencialidade, integridade do dado, controle e auditoria. Dentre as diversas publicações podem-se destacar alguns assuntos tais como a privacidade focando a confidencialidade do dado e da integridade do dado (SHAIKH; HAIDER, 2011; ZHOU et al., 2010; SUMTER, 2010).

No decorrer deste capítulo são abordados conceitos que norteiam o mecanismo proposto, tais como a definição da privacidade, riscos no compartilhamento de dados e a privacidade garantida pelo mecanismo focando na temática da liberação de dados sensíveis e considerações finais.

2.2 Definição da privacidade

Ao acessar algum tipo de serviço em um ambiente computacional, torna-se necessário informar um nome de usuário e uma senha. Com essa informação, a aplicação detecta a identidade do usuário e quais serviços ele têm direito de usar.

Uma identidade é uma representação de uma entidade dada um domínio por meio de um conjunto de atributos (SAKURAGUI, 2012). Entidade significa uma pessoa ou organização no mundo real que pode acessar um serviço ou executar qualquer tipo de operação. Os atributos representam a característica de uma entidade em um domínio em particular (SAKURAGUI, 2012). Uma pessoa ou organização que é uma entidade pode ter nenhuma, uma ou mais que uma identidade em um domínio da *web* (JOSANG; POPE, 2005).

Uma identidade é um conjunto de características que é usada para identificar. As características podem ou não ser únicas dentro de uma identidade de domínio e podem diferenciar de acordo com o tipo de entidade do mundo real mapeada (JOSANG; POPE, 2005). Por exemplo, uma característica como data de nascimento é aplicada para a pessoa e não a uma organização (JOSANG; POPE, 2005).

O relacionamento entre entidade, identidade e características/identificadores é descrito na Figura 2.1, ilustrando que uma entidade pessoa ou uma organização pode ter uma ou mais identidades e cada uma pode ter múltiplas características identificadoras, únicas ou não.

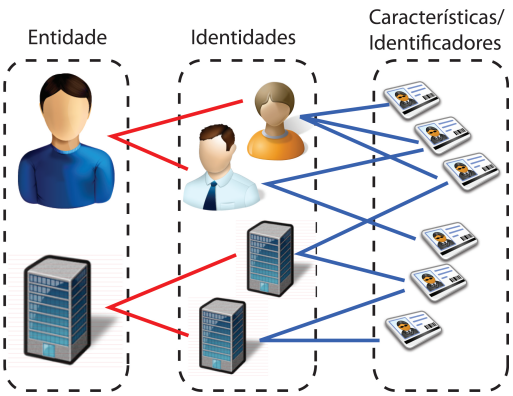

Figura 2.1: Correspondência entre entidade, identidade e atributos. Figura adaptada de (JOSANG; POPE, 2005)

A identidade é única, ao contrário da característica de uma identidade que não é única. Por exemplo, data de nascimento não unifica uma pessoa porque podem existir muitas pessoas com a mesma data de nascimento.

A forma mais comum de gerenciamento de identidade de usuário é o modelo em que cada website ou serviço cria um perfil em que coleta alguns atributos, entre eles nome de usuário

e senha, que devem ser informados em acessos subsequentes. De acordo com Josang e Pope (2005), o modelo de identidade do usuário isolado, apresentado na Figura 2.2, deixa o servidor de serviço agir como um provedor de credencial e provedor de identificação para os clientes. Assim, o usuário tem que separar credenciais, tais como senha associada com um usuário identificador (nome de usuário).

O modelo de identidade do usuário isolado, apresentado na Figura 2.2, é uma abordagem simples para o servidor de serviço, mas tem um custo para o usuário caso considere a quantidade de identificadores e credenciais que deve gerenciar. A sobrecarga de nome de usuário e senha pode conduzir o usuário ao reuso entre os diferentes websites, tornando-se um problema de segurança caso o site seja violado, podendo comprometer todas as outras contas (IVES; WALSH; SCHNEIDER, 2004).

No mecanismo de auditoria proposto neste trabalho, a identificação do usuário que acessa é requisitada quando os dados sensíveis são acessados. O serviço de auditoria construído simplesmente recebe o usuário protegendo a sua identidade com o pseudônimo. Por isso, esse serviço é uma forma de melhorar a segurança ao acesso a dados sensíveis ofertando um novo controle ao proprietário do dado que, ao mesmo tempo, garante a privacidade do usuário do acesso nos logs gerados.

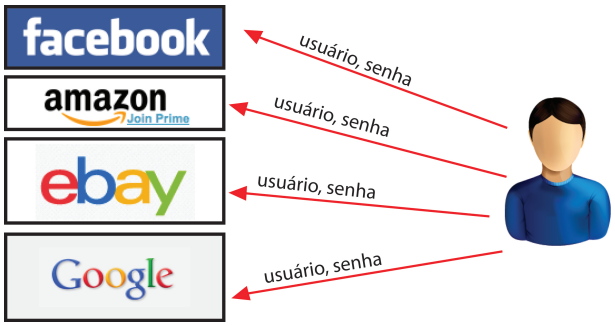

Figura 2.2: Identidade do usuário isolado

A privacidade tem significados diferenciados de uma pessoa para outra e a diferença é percebida ao comparar a visão da privacidade de cada indivíduo. Para algumas pessoas, a divulgação de dados e informações pessoais pode ser completamente normal, mas para outras pessoas a divulgação pode ser considerada uma invasão. Assim, a invasão da privacidade pode ser contextualizada de forma diferenciada considerando a visão pessoal de cada indivíduo em relação à privacidade (ISHITANI, 2003).

Os conceitos dos termos pessoal e privado são usados como sinônimos na maioria das vezes, mas, na abordagem deste texto, entende-se que pessoal é usado para definir informação sobre uma pessoa, a qual pode ser privada ou não. Já o termo privado é o que a pessoa quer

manter em segredo ou confidencial. Geralmente, informações privadas são determinadas como um conjunto de informações pessoais (PRIVACILLA, 2003).

Uma definição de privacidade amplamente discutida e difundida é apresentada por Samuel D. Warren e Louis D. Brandeis no artigo *"The Right to Privacy"* publicado em *Harvard Law Review* em 1890 (FISCHER-HÜBNER, 2001). A privacidade, por esses dois advogados, foi interpretada como sendo *"the right to be alone"*, ou seja, o direito de ser deixado sozinho. Essa publicação foi decorrente do desenvolvimento das novas formas de tecnologia associadas ao desenvolvimento. A fotografia usada pela imprensa, na visão dos autores, foi um ataque à privacidade no sentido do direito de ser deixado sozinho. (FISCHER-HÜBNER, 2001)

Segundo Fischer-Hübner (2001), a definição mais comum em uso de privacidade é a de Alan F. Westin, proposta em 1967. Westin define a privacidade como direito de indivíduos, grupos e instituições de determinar quando, como e qual informação sobre eles pode ser conhecida por outras pessoas (FISCHER-HÜBNER, 2001). A definição também envolve quando tal informação vai ser obtida e qual o uso é realizado por outros.

Para examinar como a privacidade opera em qualquer sociedade, Westin (WESTIN, 2003) propõe rastrear três níveis: o político, o sociocultural e o pessoal.

No nível político, embora a privacidade nas sociedades democráticas sejam valorizadas e institucionalizadas, as democracias devem também liberar informações necessárias para racionalização e responsabilização das condutas de negócios e para apoiar condutas justas nos assuntos de negócios. Os oficiais precisam engajar-se na questão de vigilância para identificar adequadamente atividades antissociais, para controlar atos ilegais ou violentos (WESTIN, 2003).

No nível sociocultural e organizacional, a privacidade é frequentemente determinada pelo poder individual e posição social. O ambiente de cidades populosas e fatores de classe social e a competição moldam as oportunidades das pessoas de reivindicar a liberdade da observação dos outros. Os ricos podem sair da sociedade no momento que quiserem ao contrário dos menos favorecidos (WESTIN, 2003).

Um outro ponto ao analisar a privacidade no nível sociocultural é que os ricos não precisam liberar informações pessoais ao governo em troca de subsídios, ao contrário dos menos favorecidos economicamente e socialmente (WESTIN, 2003).

No nível sociocultural, o modelo comportamental das pessoas socialmente aceitas é mais uma questão privada do que pública, não impedindo o indivíduo de seus direitos, benefícios e oportunidades controladas pelo governo. Quando tal comportamento passa a ser inaceitável, tal questão não é sujeita a uma escolha de privacidade e não é permitido reivindicar a privacidade

(WESTIN, 2003).

No nível pessoal, a privacidade está em função da vida familiar, educação, classe social e composição psicológica. A dimensão da privacidade é refletida na necessidade e desejo individual de cada indivíduo e muda constantemente no decorrer da vida e dos eventos. Quatro possíveis condições psicológicas e estados individuais de privacidade podem ser identificados: solidão, intimidade, anonimato e reservado (WESTIN, 2003).

A necessidade de privacidade do indivíduo está sujeita a mudanças, o que faz a privacidade uma questão tão complexa e de caráter pessoal. A garantia do direito da escolha, seja para o desenvolvimento pessoal do indivíduo, seja para o exercício responsável da cidadania, torna a reivindicação por privacidade um exercício civil fundamental da sociedade democrática (WESTIN, 2003).

No contexto de sistemas de informação, Biskup e Brüggeman (1988) definem a privacidade como sendo o direito do indivíduo de controle sobre a informação. Biskup e Brüggeman (1988) mencionam que o uso intensivo dos sistemas computacionais torna a privacidade um grande desafio para o direito do indivíduo no controle da informação.

Na visão de Biskup e Brüggeman (1988), o conceito de privacidade é mais bem explicado sob o ponto de vista sociológico, considerando os vários papéis que um indivíduo exerce na sociedade. O papel é o padrão de comportamento que o indivíduo exerce dentro de um determinado grupo de pessoas, como por exemplo o papel de um marido diante da sua esposa, do pai diante dos filhos e assim por diante. A privacidade seria a independência do indivíduo de se desfazer desses papéis de acordo com o seu direito de controle e ter a certeza de que os grupos de convívio privado ou público estão respeitando a separação desses papéis (BISKUP; BRÜGGEMAN, 1988).

No ambiente do mercado eletrônico presente na Internet, a privacidade é considerada como o conjunto de informações pessoais, como citado por Wang, Lee e Wang (1998). A Internet é um ambiente revolucionário que mudou a forma como as pessoas lidam com a informação, compras e marketing, tratando-se de um novo paradigma para as relações de negócios e transações. A invasão de privacidade nesse ambiente é caracterizada pela coleta não autorizada da informação decorrente do uso das transações eletrônicas que podem capturar dados estáticos – definidos como informações que nunca mudam – tais como informações referenciais, informações médicas e documentos ou informações dinâmicas, tais como atividades de histórico e conteúdo do usuário (WANG; LEE; WANG, 1998).

Outro conceito sobre a privacidade em relação ao contexto de Internet é apresentado pela

Internet Security (GLOSSARY, 2000), que define a privacidade como sendo o direito de uma entidade, que pode ser geralmente uma pessoa agindo de acordo com o seu interesse, de determinar o grau de interação com outras pessoas dentro do seu ambiente, considerando o grau de disposição de compartilhamento de informação que essa entidade está disposta a partilhar.

Cheng et al. (2012) menciona que, dependendo do contexto, são possíveis três aplicações da definição citada pela *Internet Security* (GLOSSARY, 2000). A primeira é a privacidade do indivíduo ao usar dispositivos que compartilham a localização, tais como celulares. A segunda é a privacidade do indivíduo vigiada por dispositivos, como uma câmera de vigilância, e a terceira, a privacidade do indivíduo acessando dados registrados por dispositivos, por exemplo o acesso da câmera de vigilância revelando detalhes da rotina (CHENG et al., 2012).

Na publicação de Cheng et al. (2012), é mencionado que a privacidade do indivíduo pode ser protegida em dispositivos, comunicação, armazenamento e processamento. A privacidade aplicada ao armazenamento é o que mais se relaciona com o mecanismo deste trabalho que visa oferecer auditoria com garantia de privacidade.

A privacidade no armazenamento, segundo Cheng et al. (2012), é possível através da coleta do mínimo de informações necessárias. As informações pessoais devem ser registradas somente pelo tempo necessário e as informações só devem ser liberadas quando realmente imprescindíveis. Além disso, Cheng et al. (2012) sugere o uso de mecanismo de autenticação de acesso na base, criptografia de dados para o caso de invasão da base de dados e uso de técnicas de anonimato e pseudônimos para esconder a identidade real dos itens associados aos dados.

Humphreys (2011) cita que a tecnologia da informação trouxe muitas questões a considerar sobre a privacidade e vigilância. Quando a informação não pode ser controlada pelo indivíduo, pode ser vista por outra pessoa caracterizando a vigilância.

A vigilância pode ser definida como sendo a coleta ou processamento de dados pessoais usados com finalidades de manipular ou gerenciar aqueles que forneceram os dados, sendo que muitas vezes o indivíduo tem os dados coletados ou observados sem o conhecimento quando ou se está sendo observado (HUMPHREYS, 2011).

O uso dos recursos da tecnologia da informação permite o monitoramento do comportamento e facilita a vigilância por corporações invisíveis e observadores burocráticos que não somente vendem as informações pessoais dos observados, mas também as usam para delatar práticas de controle social e discriminação (HUMPHREYS, 2011) .

A privacidade também é assunto aplicado aos governos de países, como foi recentemente abordado no caso de privacidade inferida nos serviços de arquitetura *mashups* do Reino Unido.

Mashups são aplicações web híbridas construídas por serviços on-line independentes e têm como principal propósito maximizar o acesso e utilização dos dados, proporcionando funcionalidade e conhecimento para a comunidade de interesse, como relatado por Smith e Xu (2011).

A ação Governo Aberto garante ao cidadão britânico o direito de acessar documentos do governo para propósitos de conferência e responsabilização (SMITH; XU, 2011), o que vem se tornando prática comum em diversos países, inclusive o Brasil, conforme mencionado.

No caso do Portal da Transparência disponibilizado pelo governo federal, ele expõe de uma maneira muito direta os dados sensíveis de servidores públicos, expondo situação financeira, órgão de trabalho e jornada.

Ao tratar a questão de privacidade em serviços públicos no Reino Unido, Smith e Xu (2011) aborda a privacidade como sendo a habilidade do indivíduo ou grupo de revelar informações sobre eles mesmos ou comportamento de acordo com as circunstâncias. Smith e Xu (2011) menciona que a privacidade está sob controle do indivíduo ou grupo mais do que as organizações que mantêm o dado. Esse conceito é comum aos indivíduos, mas as organizações restringem a habilidade do indivíduo de controlar a privacidade.

Outro ponto que Smith e Xu (2011) ressaltam é que a ameaça da privacidade e as decisões sobre a privacidade das informações são dependentes do contexto. A proteção da privacidade depende do interesse que se tem em trocar informações. Muitas vezes, há a necessidade da perda da privacidade para obter algo em troca, como algum tipo de serviço.

Um outro detalhe mencionado por Smith e Xu (2011) é que a ameaça à privacidade pode ser altamente dinâmica, pois as mudanças em termos de aceitação de serviço e novos serviços podem causar alterações na base de compartilhamento da informação potencialmente sem o conhecimento ou aceitação do indivíduo. Logo, o controle da privacidade deveria estar nas mãos de quem realmente possui a informação, ao contrário do que acontece. Muitos sistemas de informações têm privacidade baseada nas políticas e práticas da organização que mantém o dado (SMITH; XU, 2011).

Segundo Smith e Xu (2011), questões de privacidade ocorrem em quatro amplos domínios, os quais podem se sobrepor. O primeiro é a privacidade no domínio físico, em que a observação não autorizada é remediada com o uso de artifícios que auxiliam a proteger a identidade do indivíduo. Por exemplo, a possiblidade de evitar o rastreamento do usuário com o uso do anonimato (SMITH; XU, 2011).

O segundo é a privacidade no domínio da informação, em que o acesso não autorizado à informação leva a identificar detalhes da vida do indivíduo prejudicando-o na esfera profissio-

nal ou privada (SMITH; XU, 2011). A privacidade varia de acordo com a pessoa, com o contexto e está mais relacionada às atitudes sociais. Assim, ameaças às informações pessoais podem existir por diversas razões que incluem constrangimento pessoal, comprometimento de relacionamentos, danos a reputação profissional e a possibilidade de discriminação (SMITH; XU, 2011).

A obtenção de informação pessoal pode incluir o rastreamento das atividades da Internet, a liberação de registros médicos, o compartilhamento não desejado de benefícios recebidos com o governo, obtenção de registros de condenações penais, informação sobre o grau de parentesco, informação sobre o vínculo com amigos e o mau uso de conteúdo obtido em redes sociais (SMITH; XU, 2011).

A privacidade no domínio da informação é geralmente mais complicada de proteger em comparação à privacidade no domínio físico. A dificuldade é decorrente da falta de familiaridade com a privacidade no domínio da informação e a complexidade de mapeamento. Como existem muitas formas pelas quais a informação pode ser liberada, torna-se muito difícil conseguir mapear os objetos, preocupações e ameaças diretamente (SMITH; XU, 2011).

O terceiro é a privacidade no domínio governamental, em decorrência de ações e de informações que o governo precisa manter em segredo, gerando conflito de interesses entre governo e população (SMITH; XU, 2011).

A ameaça à privacidade no domínio do governo pode incluir a mineração de dados de indivíduos de múltiplas fontes do departamento governamental, vigilância de indivíduos pelo governo, o desejo do governo controlar as atividades da Internet, divulgação da posição política e religiosa do cidadão. No domínio governamental há pouco a se fazer individualmente para proteger a privacidade, pois a coleta do dado pessoal pode ser forçada por legislações (SMITH; XU, 2011).

O quarto domínio é a privacidade no domínio intelectual, em que as ameaças estão relacionados com a proteção do material produzido a partir de esforço intelectual. Fazem parte desse domínio a situação do discurso livre e do direito de manter a conversação e registros das atividades em particular (SMITH; XU, 2011).

Segundo Smith e Xu (2011), a privacidade é um direito fundamental que pode ser suprimido por governo e outras instituições. No entanto, ações legais podem prevenir a liberação de informação, embora acordos de interesse público possam anular essas medidas (SMITH; XU, 2011).

Conclui-se, com base em todas definições apresentadas, que a privacidade é o direito do indivíduo de controlar o dado ou a informação que revele alguma informação privada. Assim,

a informação privada pode ser sensível ou não ao indivíduo, dependendo do conteúdo revelado pelo dado ou informação. Por sua vez, o controle da privacidade permite adequar o nível de privacidade desejado ou simplesmente controlar a liberação. Em todas as definições apresentadas, é possível perceber a necessidade do indivíduo de tomar decisão de liberar ou não o dado ou a informação sensível.

Com a era da Internet, há necessidade de ter mais controle sobre a informação pessoal, no entanto, grande controle sobre a publicação da informação resulta em liberação demasiada de dados pessoais expondo o indivíduo e evidenciando um controle paradoxo, como citado por Brandimarte, Acquisti e Loewenstein (2013).

O controle paradoxo baseia-se na revelação de mais informações sensíveis do que o normal através da noção de controle sobre a publicação do indivíduo. O controle da liberação proporcionado ao indivíduo faz com que ele preste menos atenção aos riscos e consequente uso dessa liberação por outras pessoas. A situação ocorre até mesmo quando a invasão da privacidade é decorrente do acesso ao dado e uso deste dado, e não simplesmente como é publicado (ACQUISTI; ADJERID; BRANDIMARTE, 2013).

Por outro lado, a falta de percepção de controle resulta em liberação mais baixa de informações pessoais, mesmo quando os riscos são menores. Pois a preocupação de privacidade gerada não está na forma como o dado é liberado, mas na falta de controle sobre ele (ACQUISTI; ADJERID; BRANDIMARTE, 2013).

As evidências levantadas por Acquisti, Adjerid, Brandimarte e Loewenstein (ACQUISTI; ADJERID; BRANDIMARTE, 2013; BRANDIMARTE; ACQUISTI; LOEWENSTEIN, 2013) podem ser analogamente relacionadas ao tema desta monografia e auxiliam na justificativa do mecanismo desenvolvido, uma vez que o mecanismo visa monitorar o dado já publicado.

2.3 Compartilhamento de dados

Vive-se e trabalha-se em um mundo conectado, onde é possível realizar uma conversa casual em um ambiente de bate-papo ou realizar transações financeiras de forma rápida e de valor ilimitado (SHINDER; CROSS, 2008).

Uma grande quantidade de dados pessoais são coletadas por meio de inúmeras aplicações na Internet com diversas finalidades, entre elas o uso de dados pessoais em sistemas de recomendação que indicam produtos ou serviços ao usuário conforme os dados coletados por esses programas. Mas nem tudo que é coletado visa o bem-estar do usuário, ou seja, esses dados

podem ser coletadas e usados de forma a prejudicar o usuário.

Cibercrime é um termo genérico que pode ser categorizado de acordo com delitos cometidos que, dependendo da natureza, podem ser enquadrados em categorias usadas para identificar crimes comuns(SHINDER; CROSS, 2008).

Os computadores podem estar envolvidos nos crimes de diferentes maneiras, tais como: a vítima (o computador ou rede pode ser o alvo), a ferramenta (usada para cometer o crime) ou podem ser usado para manter informações do crime tais como manter registros de vendas de drogas ilegais (SHINDER; CROSS, 2008).

Muitos cibercrimes podem ser enquadrados como "crimes do colarinho branco", em que há o envolvimento de atividades de negócio motivadas pelo benefício financeiro e envolvem roubo, trapaça e fraudes. Assim como a pornografia infantil, que se enquadra no crime de pedofilia, considerada como violenta ou potencialmente violenta (SHINDER; CROSS, 2008).

Os cibercrimes podem ser classificados como violentos, potencialmente violentos ou não violentos. Entre eles podem-se enquadrar o ciberterrorismo, o *assault by threat*, o *cyberstalking* e a pornografia infantil (SHINDER; CROSS, 2008).

O *assault by threat* e o *cyberstalking* estão mais relacionados com descoberta de dados da vítima que gera assédios e ameaças. *Assault by threat* é um delito que pode ser cometido pelo envio de email ameaçando o indivíduo e familiares próximos, instaurando medo e ameaçando a tranquilidade (SHINDER; CROSS, 2008). E o *cyberstalking* trata-se de uma forma eletrônica de assédio envolvendo ameaças implícitas ou explícitas gerando medo na vítima, que pode ser repassada para a vida real gerando violência (SHINDER; CROSS, 2008). Esta forma está mais relacionada à descoberta de dados, informações da vítima muitas vezes disponíveis on-line como sugerido por Tavani e Grodzinsky (2002).

O compartilhamento de informação facilitada pela tecnologia e pelos inúmeros aplicativos gera o questionamento sobre até que ponto a divulgação de certos dados fere a privacidade do usuário. Pode-se perceber que dados que não caracterizam o indivíduo não apresentam um grande risco, ao contrário de dados que indicam a localização, hábitos, preferências e situação financeira.

A exposição dos dados sensíveis do usuário o deixa suscetível a ações de *hackers*, que podem atuar por meio da engenharia social ou suas variações, tais como *phishing* (SHINDER; CROSS, 2008). A engenharia social é a arte de habilmente manipular o ser humano para agir em algum aspecto da vida (HADNAGY, 2011; HONG, 2013).O *phishing* é uma variação em que o *hacker* pode usar o *email*, por exemplo, para adquirir mais informações, solicitando dados

ao destinatário, que passa a acreditar que a solicitação veio de uma fonte confiável (SHINDER; CROSS, 2008) e cai em armadilhas, fornecendo mais informações, tais como conta bancária, cartão de crédito e endereços.

Dessa forma, pode-se entender que dados sensíveis ajudam a identificar de alguma forma o indivíduo ou identificar dados vitais para uma empresa. A vulnerabilidade que o acesso a dados sensíveis proporciona abre possibilidade de ações de algum cibercrime. Com este trabalho, a auditoria ao acesso ao dado permite o monitoramento, ajudando a visualizar cenários mais claros do acesso aos dados e auxiliando o proprietário do dado a identificar situações de frequência e origem de acesso.

2.4 A privacidade garantida pelo mecanismo

A Lei de Acesso à Informação fortalece o acesso do cidadão à informação do governo (JORDÃO, 2011) e tem como regra liberar o acesso às informações com exceção de informações pessoais e dados classificados pelas autoridades como confidenciais (JORDÃO, 2011).

Considerando o Brasil e a Lei de Acesso à Informação, informações pessoais podem ser entendidas como aqueles dados relacionados à identificação natural da pessoa ou identificável, íntimo, vida privada, honra e imagem. Esses dados podem ser liberados ou acessados por terceiros diante da previsão legal ou consentimento expresso do seu proprietário (PLANALTO.GOV.BR, 2012).

O governo implantou a lei com o objetivo de aumentar a confiança do público, pois, através da cultura de acesso a esses dados, os agentes públicos se conscientizarão de que as informações públicas pertencem ao cidadão e é uma obrigação do Estado entregar esse tipo de informação para a população (JORDÃO, 2011).

No contexto deste trabalho, dados sensíveis podem ser compreendidos como um conjunto de atributos que revelam hábitos, situação econômica ou qualquer outro dado cujo acesso deve ser auditado. Os dados sensíveis podem revelar algum aspecto que pode ameaçar o proprietário do dado facilitando a ocorrência de atos ilícitos.

O Portal da Transparência do governo federal é um excepcional exemplo de promoção da transparência. Por outro lado, expõe dados sensíveis tais como rendimento mensal e lugares onde o servidor público trabalha, expondo demasiadamente o servidor.

No mecanismo proposto neste trabalho, a proteção da privacidade é aplicada na proteção do usuário que consulta o dado sensível. Como resultado, esta técnica de proteção caracteriza

a auditoria de pseudônimo, pois os logs possuem pseudônimos da identidade real do acesso e permitem identificar em situações somente quando há necessidade.

Assim, a garantia de transparência ao usuário que acessa o dado sensível é obtida com uso de pseudônimos que protegem a identidade. E, ao mesmo tempo, o proprietário do dado tem noção dos acessos sofridos pelo dado, garantindo a integridade, pois sabe quem acessa e mantém a transparência do acesso.

2.5 Considerações finais

A identidade e a privacidade são conceitos importantes para o mecanismo. A identidade determina quem é o sujeito que acessa o dado sensível e deve ser protegida para manter a privacidade e o direito de visualizar dados cujo acesso lhe é concedido.

Capítulo 3

AUDITAR COM GARANTIA DE PRIVACIDADE

O capítulo apresenta que o uso da auditoria incide sobre a acumulação de registros gerados pela auditoria que podem provocar a invasão da privacidade do usuário. E possibilitar uma auditoria com garantia de privacidade é um ponto chave para conduzir a uma análise imparcial e garantir a liberdade do usuário. A computação em nuvem entra neste projeto como um ambiente que oferece recursos de processamento e armazenamento vantajosos que potencializa a ideia de criação de um mecanismo em nuvem de monitoramento a dados sensíveis que permite auditoria e prestação de contas com garantia de privacidade.

3.1 Considerações iniciais

A segurança da autenticação pode ser melhorada com a associação de diferentes técnicas que podem usar a biometria, a senha e outros meios disponíveis. Ainda assim, aumentar a proteção da autenticação com a combinação de diferentes recursos para verificação do usuário não é sinônimo de total proteção contra acessos ilegais.

A tarefa de tornar um ambiente mais seguro pode ser complementada com a auditoria, que ajuda a identificar mau uso de recursos e os acessos para garantir conformidade com a política liberada ao usuário. A auditoria garante uma cobertura completa, registrando quem está acessando e qual tipo de operação a pessoa executa.

São apresentados neste capítulo os componentes comuns em um sistema de auditoria; como um sistema de detecção de intrusão que auxilia na análise de registros de acessos e a visão da auditoria de pseudônimo, que mantém a privacidade das ações do usuário nas trilhas de auditoria.

No decorrer deste capítulo, são apresentados: a) os componentes de um sistema de audi-

toria; b) um sistema de detecção de intrusão que auxilia na análise das trilhas de auditoria; c) como é possível manter a privacidade das trilhas de auditoria com o uso de pseudônimos; d) as modificações realizadas na auditoria de pseudônimo em comparação com a proposta original de Fischer-Hübner (2001) e e) uma visão geral do ambiente de computação na nuvem que vai possibilitar a oferta desse mecanismo como serviço.

3.2 Componentes de um sistema de auditoria

O desenvolvimento da técnica de auditoria foi decorrente da necessidade de rastrear o acesso à informação sensível importante armazenada em sistemas de computadores, assim como o acesso a sistemas de computadores.

Auditar tem como objetivo identificar vulnerabilidades, melhorar controles internos, verificar implementações e fornecer conclusões para a gerência, de acordo com Koch (1979). É uma função básica de um sistema de segurança que fornece registros e análise de todos os eventos de segurança importantes.

A auditoria adiciona um nível de garantia em que o mau uso não é indetectável e serve para deter a invasão de sistemas de computador. As trilhas de auditoria são técnicas para determinar a violação de segurança e um ponto importante para qualquer sistema de segurança (BISHOP, 2002).

Os registros gravam eventos, ações, fornecendo logs de atividade e permitindo uma autópsia do que aconteceu e como. Os logs fornecem um mecanismo para analisar o estado de segurança do sistema ou mesmo determinar se uma simples requisição coloca o sistema em um estado não seguro (BISHOP, 2002).

Os registros de todos os eventos de atividade permitem a reconstrução do estado do sistema em qualquer momento e a gravação permite determinar possíveis causas de problemas de segurança e fornece novos pontos para mais análises (BISHOP, 2002).

Mecanismos de auditoria podem ser usados na tentativa de revisar padrões de uso, melhorando a efetividade de mecanismos de proteção. Os padrões estabelecem uso de recursos críticos para alguns sistemas de detecção de intrusão. O ataque é percebido pelos registros e análises, fornecendo garantia de violação da política de segurança (BISHOP, 2002).

Desse modo, rastrear o usuário com o uso da auditoria levanta dois problemas distintos: qual o tipo de informação é coletada para construir o log e o que precisa ser analisado para fazer a auditoria. A coleta do que vai ser auditado depende do conhecimento em relação às políticas

de segurança do sistema em questão (BISHOP, 2002).

No entanto, é possível destacar alguns eventos que um mecanismo de auditoria deve ser capaz de registrar: autenticação, identificação, objetos manipulados pelo usuário, exclusão de objetos, ações feitas pelo usuário e outros eventos importantes pertencentes ao sistemas de computador.

Na análise de um registro de auditoria de um evento, deve ser possível identificar: data, horário, usuário do evento, tipo de evento e sucesso ou falha do evento (FISCHER-HÜBNER, 2001).

Em um registro de auditoria criado a partir de um evento de autenticação/identificação, é necessário identificar a origem de sua requisição. E para eventos que apaguem ou acessem objetos, o mecanismo de auditoria precisa registrar o nome do objeto (FISCHER-HÜBNER, 2001).

O mínimo de impacto de um mecanismo de auditoria é importante para manter o desempenho do sistema e o alto grau de confiabilidade (FISCHER-HÜBNER, 2001). A confidencialidade e a integridade devem ser conservadas pela organização considerando o significado da informação que pode ser obtido pela trilha de auditoria (FISCHER-HÜBNER, 2001).

No caso de altos requisitos de segurança, um papel de auditor pode ser inserido para impor a separação de deveres. E recursos técnicos podem ser fornecidos para evitar a perda de dados da auditoria, considerando a sobrecarga de trilhas de auditoria gerada pelo registros constantes de dados (FISCHER-HÜBNER, 2001).

De acordo com Bishop (2002), existem três componentes que são base para um sistema de auditoria, denominados coletor, analisador e notificador, detalhados a seguir (BISHOP, 2002).

O componente coletor é responsável pela coleta do dado e pela criação das informações de log. O tipo, a quantidade, o dado necessário nessa coleta são ditados pelo sistema ou alguns parâmetros de configuração. O dado coletado pode ser gravado na forma original, na forma binária ou enviado a um mecanismo de análise (BISHOP, 2002).

O componente analisador é responsável pela obtenção das informações contidas no log e pela análise (BISHOP, 2002).

O componente notificador é responsável por informar ao analista ou outra entidade o resultado da auditoria. Essa ação pode iniciar uma série de atitudes preventivas (BISHOP, 2002).

A auditoria usualmente gera um grande número de dados auditáveis, o que pode atrasar a detecção de intrusão e uma percepção tardia do ataque (FISCHER-HÜBNER, 2001).

Um sistema de detecção de intrusão identifica ataques analisando os registros de *logs* para

atividades não esperadas ou para atividades que podem comprometer o sistema. O mecanismo de análise de um sistema de detecção de intrusão é um exemplo de analisador de um mecanismo de auditoria (BISHOP, 2002), sendo o sistema de detecção de intrusão detalhado na seção 3.3.

Este trabalho tem como finalidade registrar eventos de acessos a dados sensíveis por um determinado usuário, sendo os dados repassados a um coletor que registra o log de acesso. Na sequência o componente responsável pela análise, que é construído com base em um conjunto de regras pré-configuradas, consegue determinar comportamentos tais como acessos mais frequentes, registros mais acessados, entre outras regras. No final, o componente notificador apresenta ao proprietário do dado as análises dos acessos auxiliando-o a entender os acessos, algum tipo de possível mau uso e colaborando para melhorar as políticas de liberação de dados.

3.3 Sistema de detecção de intrusão

Auditoria é vital para obter evidências para auxiliar a análise do que aconteceu (FISCHER-HÜBNER, 2001). Associada a um sistema de detecção de intrusão, fornece detecção em tempo real de violação de segurança identificando quem prejudica o sistema ou abusa dos privilégios cedidos pelo próprio sistema (FISCHER-HÜBNER, 2001). O sistema de detecção de intrusão possui diferentes modelos tais como modelos de anomalia, modelos de mau uso e a modelagem de especificação (BISHOP, 2002).

O modelo de anomalia detecta a violação de segurança analisando os registros de auditoria, procurando por padrões anormais de uso do sistema. Nesse modelo, um perfil anterior de comportamento de assuntos em relação ao objeto é fornecido na forma de métricas, modelos estatísticos e regras para dar base para aquisição de conhecimento e comportamento anômalo vindo dos registros de auditoria. Detecção de anomalia é responsável por comparar o comportamento normal com parâmetros da sessão do usuário atual. Um desvio importante em relação ao comportamento normal precisa ser informado pelo responsável.

O modelo de mau uso requer conhecimento de vulnerabilidades do sistema ou potenciais vulnerabilidades que os atacantes tentam explorar. O sistema de detecção de intrusão incorpora esse conhecimento como um conjunto de regras e, quando o dado é passado para o sistema de detecção de intrusão, é aplicado um conjunto de regas ao dado para determinar se qualquer sequência do dado combina com qualquer uma das regras.

A modelagem de especificação tem o caminho oposto dos outros dois modelos. Enquanto o modelo de anomalia tem o enfoque de procurar estados não comuns e o modelo de mau uso tende a procurar por estados conhecidos como sendo mau uso, a modelagem de especificação

procura por estados conhecidos por não serem bons. Assim, quando o sistema encontra esse estado, é denominado como uma possível intrusão.

Um sistema de detecção de intrusão recebe dados auditáveis de qualquer parte de um sistema. Assim, a responsabilidade é analisá-lo e compará-lo com métricas, perfis estatísticos ou com banco de dados que possuem dados sobre o mau uso para determinar se o comportamento é suspeito. A fonte de dados para um sistema de detecção de intrusão vem das trilhas de auditoria do sistema, log de sistemas ou pacotes da rede de computador, como ilustrado na Figura 3.1.

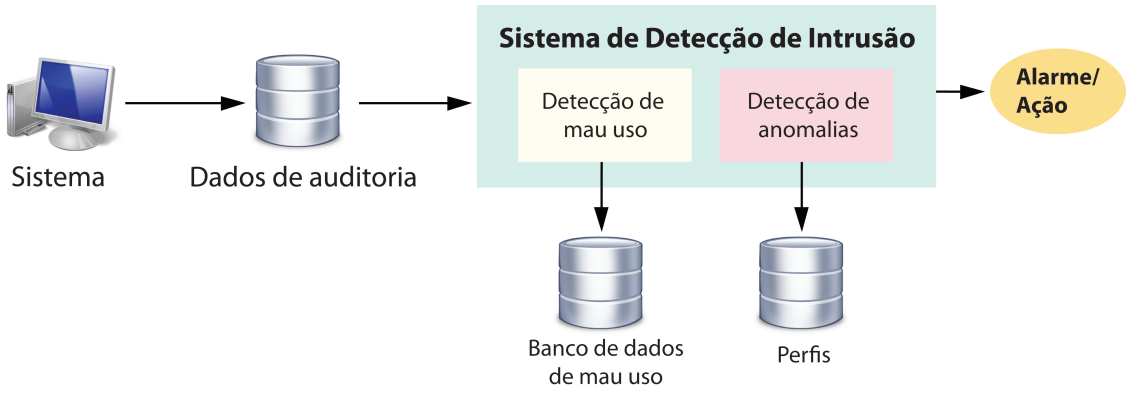

Figura 3.1: Arquitetura de um sistema de detecção de intrusão (FISCHER-HÜBNER, 2001)

O mecanismo em nuvem de monitoramento a dados capta os logs de acesso ao dado definido como sensível pelo proprietário. Os logs que alimentam a base do mecanismo são analisados por um conjunto de regras pré-estabelecidas que define os cenários de detecção de intrusão, em um processo semelhante ao executado por um sistema de detecção de intrusão baseado no modelo de mau uso.

Na realidade, o mecanismo fornece um cenário que apresenta uma análise dos logs que conduz o proprietário a tomar suas decisões e identificar comportamentos de mau uso com base em regras pré-definidas nesses cenários.

A criação dos cenários tornou-se necessária porque, pela quantidade de logs gerados pelo mecanismo, deve existir uma forma mais simples de analisar o tráfego dos acessos. Assim, um sistema de detecção de intrusão, via de regra, responde a uma análise imediata de tudo o que é gerado pelo log, apontando de uma forma clara os problemas ocorridos.

No entanto, o cenário fornecido pelo mecanismo de monitoramento a dados sensíveis se baseia no conceito de um sistema de detecção de intrusão, porém fornece as análises na forma de consultas com que o proprietário do dado pode interagir, podendo analisar o comportamento dos acessos.

A identificação do mau uso representa um comportamento fora do padrão dentro da regra

geral estabelecida, sendo necessário identificar o responsável pelo acesso. Assim, o uso de auditoria de pseudônimo se tornou necessário porque, ao mesmo tempo que protege a identidade real do usuário, impede a invasão de privacidade possibilitando uma forma de descobrir a nova identidade.

3.4 Auditoria de pseudônimo

Auditoria fornece a chance de recriar os passos até quando os problemas aconteceram, ajudando a detectar o mau uso ou melhorar o sistema de proteção contra violações de segurança. Além de já ter a autenticação ou controle de acesso, usar auditoria pode ser uma forma de melhorar a prevenção contra violações de segurança.

Com a auditoria, todas as ações feitas pelo usuário são coletadas usando uma aplicação ou usando um sistema operacional. Assim, as trilhas de auditoria têm muitos dados sobre os hábitos do usuário. E de alguma forma esse tipo de operação invade a privacidade do usuário.

A auditoria de pseudônimo é uma técnica de segurança que aumenta a privacidade através da adição de um pseudônimo ao usuário que está sendo auditado. Quando usado em um sistema de detecção de intrusão, ajuda a oferecer uma abordagem mais sociável e legalmente aceitável (FISCHER-HÜBNER, 2001).

Assim, o pseudônimo vai ser aplicado para proteger a identidade do usuário que acessa o dado sensível. E a identidade real somente é descoberta se forem identificadas anomalias no acesso aos dados sensíveis. Um pseudônimo é usado para proteger um nome real, pois se trata de um nome falso que pode identificar o usuário através de um domínio, podendo ser o remetente ou o destinatário real (PFITZMANN; HANSEN, 2009).

Existe um conflito no sistema de segurança que, ao mesmo tempo que protege a privacidade do dado e dos usuários, pode ser usado para invadi-los (FISCHER-HÜBNER, 2001). Esse conflito pode ser contornado com o uso da auditoria de pseudônimo.

Na auditoria de pseudônimo, o identificador de assunto e o usuário identificador de dados nos registros de auditoria são apresentados sob um pseudônimo depois de sua criação (FISCHER-HÜBNER, 2001).

O uso de pseudônimos deve cobrir campos, tais como: identificadores de usuários, identificadores de localização, subdiretórios e nomes de objetos referindo-se a nomes de usuários (FISCHER-HÜBNER, 2001). Com a auditoria de pseudônimo é possível proteger a privacidade do usuário que executa a ação e responsabilizá-lo, caso seja necessário.

A auditoria de pseudônimo é uma técnica de auditoria pela qual objetos identificadores e dados identificadores do usuário nos registros de auditoria são substituídos por pseudônimos logo após a sua criação e os registros auditados podem ser analisados por um sistema de detecção de intrusão, como apresenta a Figura 3.2 (FISCHER-HÜBNER, 2001).

Quando os dados de auditoria são analisados, não há necessidade de saber a real identidade do usuário monitorado, pois é suficiente saber que a identidade real pode ser determinada em caso de suspeita ou um comportamento intrusivo de um usuário agindo sob um pseudônimo. Assim, a re-identifição do usuário é feita a fim de desmascará-lo como um intruso, por exemplo (FISCHER-HÜBNER, 2001).

A auditoria de pseudônimo fornece responsabilidade ao usuário assim como pseudo-anonimato. Em princípio, a auditoria de pseudônimo é também aplicada em outras formas de auditoria, assim como auditoria de aplicações (FISCHER-HÜBNER, 2001).

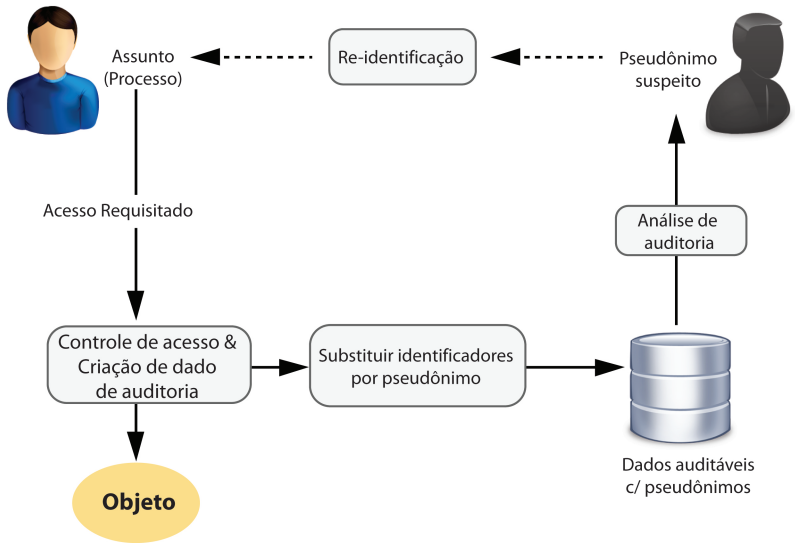

Figura 3.2: Funcionalidade do pseudônimo da auditoria de sistema operacional (FISCHER-HÜBNER, 2001)

De acordo com Fischer-Hübner (2001), a auditoria de pseudônimo é a habilidade de associar pseudônimos aos usuários e vinculá-los aos dados coletados durante o seu monitoramento. Assim, pseudônimos são usados e ao longo do tempo a quantidade de informações armazenadas sobre um pseudônimo pode ser usada para propósito de re-identificação. Então, passa a ser necessário modificar o pseudônimo depois de um período para proteger contra a re-identificação.

Os pseudônimos podem ser implementados através de referência de pseudônimos ou pseudônimos criptográficos. Para a implementação do pseudônimo de referência, os pseudônimos no banco de dados precisam ser mantidos e administrados, o que pode causar considerável sobrecarga.

Os pseudônimos criptográficos podem, por exemplo, ser gerados por chaves de criptografia simétrica do usuário que identifica o dado no registro de auditoria. A descriptografia pode ficar sob responsabilidade de duas pessoas, uma que protege o interesse da privacidade da pessoa e a outra o administrador, sendo possível identificar o usuário apenas quando as duas pessoas entrem em acordo (FISCHER-HÜBNER, 2001).

No contexto deste trabalho, o proprietário do dado precisa auditar o acesso ao dado. Entretanto, há necessidade de manter a transparência entre a necessidade de monitorar através de logs o acesso e a necessidade de dar uma visão geral sobre quem acessa ao proprietário do dado. Logo, a aplicação de auditoria de pseudônimo soluciona questões de privacidade do acesso do usuário porque o pseudônimo refere-se a um usuário e a revelação da identidade real é atribuída ao proprietário do dado com a quebra do pseudônimo.

3.5 Adaptação do auditoria de pseudônimo para o mecanismo de auditoria em nuvem

A necessidade de dar ao usuário a garantia de que não vai ser coagido sem um julgamento justo e que ao mesmo tempo promova um serviço com operações que possam apurar a responsabilidade de um determinado acesso, foram alguns dos pontos explorados neste trabalho.

A auditoria de pseudônimo é uma técnica em que o nome real de identificadores de objeto e identificadores de dados de usuário são substituídos por um pseudônimo diretamente após a criação dos registros de auditoria. E o pseudônimo precisa ser modificado depois de um período de tempo para evitar a re-identificação baseada na quantidade de pseudônimos dos arquivos de *logs*.

A auditoria de pseudônimo (FISCHER-HÜBNER, 2001) permite melhorar a privacidade do usuário e dá garantia ao administrador do sistema de recuperar a identidade real quando necessário, como mencionado anteriormente.

O conceito por trás da auditoria de pseudônimo é garantir a privacidade do usuário através de um pseudônimo. Esse tipo de solução contribui para a solução do conflito entre sistemas de segurança e privacidade, evitando a invasão de privacidade e protegendo o usuário ao mesmo tempo (FISCHER-HÜBNER, 2001). Um pseudônimo, nesse cenário, pode ser usado para mascarar os dados sensíveis do usuário, evitando a identificação sem uma motivação real.

A vantagem de uma auditoria de pseudônimo é a possibilidade de responsabilizar algum usuário por suas ações, porque o mecanismo de auditoria prevê uma forma de recuperar os

dados reais sobre o usuário, assim como apresentado na Figura 3.2.

As trilhas de auditoria criadas que usam estes pseudônimos são analisadas por sistemas de detecção de intrusão ou outro mecanismo, garantindo a privacidade do usuário (FISCHER-HÜBNER, 2001). Os autores em Fischer-Hübner (2001) ainda mencionam que, se algum ato suspeito é identificado pelo administrador de segurança, o mesmo só pode desmascarar a identidade real com a cooperação do escritório protetor de dados.

O pseudônimo é a arte de mascarar a identidade real e dar apelidos ao indivíduo através de um sistema (PFITZMANN; HANSEN, 2009). Adotar um pseudônimo pode resultar na queda de desempenho do sistema em decorrência do uso de criptografias ou pseudônimos baseados em referência. Para um pseudônimo de referência é preciso manter e administrar uma base de dados de pseudônimo, provocando um overhead.

Outra solução seria criptografar os dados identificadores do usuário com chaves simétricas para gerar um pseudônimo. Com essa abordagem, a chave de descriptografia seria dividida em duas metades; cada metade poderia ser redirecionada para o escritório de segurança e a pessoa que protege o dado, como proposto pelos autores em (FISCHER-HÜBNER, 2001).

A Figura 3.2 mostra uma aplicação de auditoria de pseudônimo dentro do contexto de sistemas operacionais pelo autores em (SOBIREY; RANNENBERG, 1997; FISCHER-HÜBNER, 2001). Entretanto, para a abordagem deste trabalho, algumas adaptações na auditoria de pseudônimo precisaram ser feitas.

Tais adaptações permitiram que um mecanismo a ser ofertado como um serviço na nuvem protegesse a identidade do usuário nos logs gerados. As modificações atingem a área de criação do pseudônimo e a forma como o proprietário identifica a identidade real do usuário que realizou o acesso.

Nesse esquema para auditar o acesso a dados com garantia de privacidade, espera-se a invocação de um serviço dentro da aplicação, aumentando a versatilidade da proposta. Para tornar essa integração possível, o proprietário do dado tem de definir o conjunto de dados sensíveis a cada acesso feito a ele. O conjunto de dados sensíveis, o usuário que fez o acesso e o *token* – criado quando o proprietário do dado junta-se ao serviço que identifica se o conjunto de dados pertence a algum proprietário – serão enviados para a nuvem.

A cada vez que um usuário acessa um dado sensível, será verificado se ele não está cadastrado no banco de dados hospedado no serviço disponível na nuvem. Em caso negativo, o usuário que acessa o dado sensível é salvo. Isto é necessário porque em algum momento o proprietário do dado visualiza a verdadeira identidade de quem fez o acesso, então, a única ligação

é o repositório do usuário disponível no serviço.

Apesar de existir um repositório de usuário na nuvem, isso não significa que a relação do que o usuário acessa com a trilha de auditoria é criada toda vez que o serviço de geração de log é chamado. Muito pelo contrário, quando o usuário é registrado pela primeira vez na nuvem pela invocação do serviço de geração de log, chaves criptográficas são geradas com base em chaves assimétricas. Assim, a cada novo acesso feito por esse usuário ao dado dentro do aplicativo, é verificada a sua chave pública e a nuvem vai salvar um log com um nome de usuário diferente com base na criptografia de e-mail informado usando a chave pública, como mostrado na Figura 3.3.

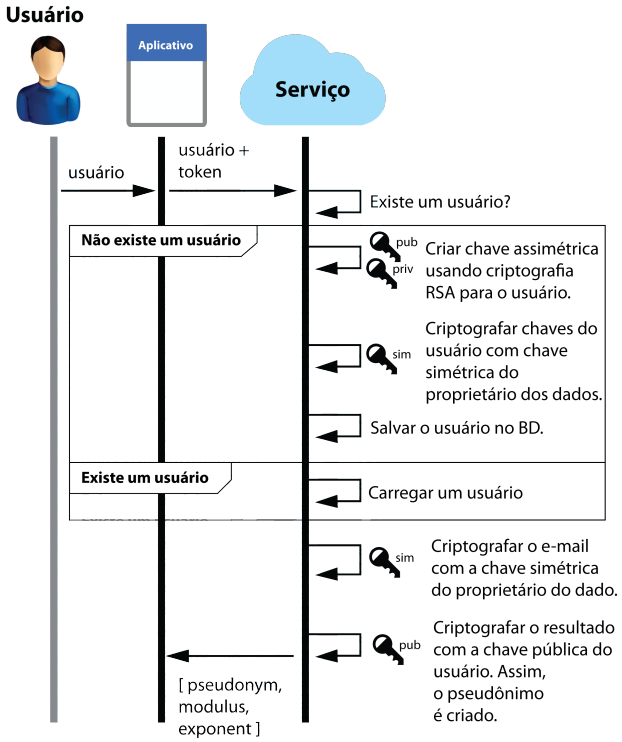

Figura 3.3: Gerando um pseudônimo para proteger a privacidade do usuário.

A adaptação da auditoria de pseudônimo é notada quando o pseudônimo é criado com base em chaves assimétricas de criptografia. No mecanismo em nuvem de monitoramento a dados sensíveis, foi adotada a criptografia RSA, sendo uma das principais diferenças em comparação com a abordagem original da auditoria de pseudônimo. Essa modelagem garante um novo pseudônimo em cada acesso e o vínculo de quem realizou o acesso que gerou o log é o pseudônimo, o módulo de chave privada e o expoente de chave privada registrada no log.

Outra diferença é em relação à descoberta da identidade real do usuário. No serviço disponibilizado pelo mecanismo, a identificação da identidade real do usuário só é possível com o uso da chave privada do usuário composta pelo módulo e pelo expoente, gravado no log. Além

disso, a chave simétrica do proprietário do dado deve ser utilizada também, como é mostrado na Figura 3.4.

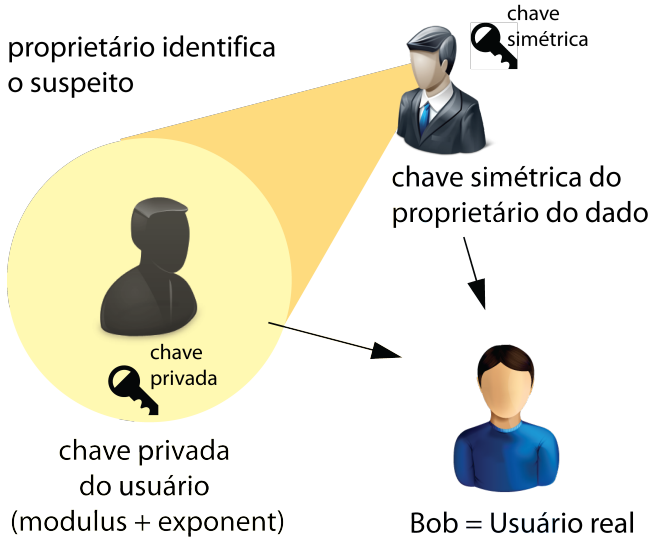

Figura 3.4: O proprietário do dado identifica um suspeito com base no cenário de detecção

Ao criar o pseudônimo, a primeira criptografia é feita pela chave simétrica do proprietário do dado e a última criptografia usa a chave pública do usuário para criar o pseudônimo, encriptando o resultado da criptografia da chave simétrica do proprietário do dado, como mostrado na Figura 3.3.

Para revelar a verdadeira identidade do usuário, é necessário o uso de duas chaves para descriptografar. Assim, independentemente de armazenar os pedaços da chave no log, isso não torna mais fácil revelar a verdadeira identidade do usuário, pois a chave simétrica AES do proprietário do dado precisa ser acessada.

Outro diferença em relação à abordagem original é a necessidade de registrar o módulo e o exponente no log. Os dois campos são necessários porque auxiliam os cenários de detecção a calcular a quantidade de acesso, entendendo que se trata de uma pessoa mas sem identificar quem é. Isso garante o anonimato e identifica que todos os registros de auditoria estão relacionados a algum usuário com uma coleção diferente de pseudônimos.

A adaptação apresentada é aplicada em um contexto diferente, em comparação à abordagem original que foi aplicada em sistema operacional. A adaptação de auditoria de pseudônimo mantém a privacidade do usuário quando acessado diretamente a base de dados. Dessa forma, a privacidade do acesso do usuário é garantida mesmo a partir do provedor de serviço na nuvem, onde o log é registrado.

É importante salientar que a adaptação da auditoria de pseudônimo foi aplicada na identi-

ficação do usuário nos logs guardados na nuvem. Só que os dados que revelam informações sobre a localização e dispositivo não foram anonimizadas para alimentar os cenários de detecção disponíveis no serviço. O serviço desenvolvido no trabalho se propõe explorar a integração com aplicações, tornando um *Software* como um serviço (SaaS) a ser requisitado por qualquer aplicação instalada na nuvem ou em outro servidor.

3.6 Computação em nuvem

A computação na nuvem relacionada a temas tais como privacidade e acesso a dados sensíveis torna-se uma área com diversas oportunidades de investigações e criação de novas propostas, como é discutido nos trabalhos relacionados do Capítulo 4, e representa uma nova forma de desenvolvimento e disponibilização de software presente na indústria de Tecnologia de Informação.

A computação na nuvem é uma área concentrada de recursos computacionais, armazenamento, infraestrutura colaborativa, escalável e sob-demanda. Negócios e aplicações podem ser oferecidos ao público de uma empresa ou para milhares de usuários no mundo como um serviço (HILL et al., 2013).

Uma grande vantagem da abordagem de computação na nuvem é que o cliente só paga pela quantidade de recursos que usar, e pode solicitar mais poder de processamento ou menos, mais poder de armazenamento ou menos, mais memória ou menos de forma rápida. E o uso extra só é considerado pelo período utilizado. Isso torna desnecessário o investimento em massa de *hardware*, que muitas vezes é feito para suprir a demanda em horas intensas de uso, sendo, que na maior parte do tempo, esse *hardware* adicional pode ficar sem uso (HILL et al., 2013).

A computação na nuvem está mudando a forma como as organizações lidam com os seus recursos tecnológicos (*hardware/software*) (SRINIVASAN et al., 2012). Nas nuvens, os clientes não precisam ter conhecimento dos detalhes específicos de tecnologia enquanto hospeda a sua aplicação, o serviço é completamente gerenciado pelo servidor de computação nas nuvens e o cliente consome o serviço conforme suas necessidades (SRINIVASAN et al., 2012). O servidor lida com toda a complexidade em nome do cliente e oferece os recursos necessários para execução e gerenciamento das aplicações (SRINIVASAN et al., 2012).

A nuvem foi escolhida para este projeto pela facilidade na aquisição de recursos computacionais e suporte escalável, além de possuir diversos desafios a serem enfrentados em relação à privacidade e auditoria. A computação na nuvem possui cinco características essenciais definidas pelo *National Institute of Standards and Technology* (NIST) (HILL et al., 2013), como

discutido a seguir:

(a) **Aprovisionamento automático:** A nuvem trouxe a possibilidade de terceirização de infraestrutura de recursos computacionais, assim, a compra de *hardware* é minimizada. Um exemplo disso seria a requisição de mais área para armazenamento na nuvem. Isso não requer a compra de *hardware* extra por parte do cliente. Por outro lado, após a solicitação de mais recursos, não pode haver demora de resposta da nuvem. Logo, a computação na nuvem deve incorporar agilidade suficiente e autonomia para provisionar recursos em tempo real de forma automática e dinâmica, quando solicitado, sem a intervenção humana (HILL et al., 2013; SRINIVASAN et al., 2012).

(b) **Acesso amplo à rede:** A computação na nuvem deve fornecer acesso a redes de computadores tais como Internet, através de protocolos. E acesso aos diversos dispositivos existentes tais como computadores pessoais, computadores portáteis, *tablet* e celulares (HILL et al., 2013; SRINIVASAN et al., 2012).

(c) ***Pool* de recursos:** Relacionado aos aspectos de Computação em Grade e Virtualização de *Hardware*, permite a virtualização dos recursos da nuvem em uma grande camada virtual. Isso garante eficiência com o gerenciamento dinâmico de *hardware* e recursos virtualizados, deixando a oferta de recursos transparente ao cliente, já que não há distinção entre local físico ou granularidade de recursos (HILL et al., 2013; SRINIVASAN et al., 2012).

(d) **Rápida elasticidade:** A demanda de novos recursos é auto-gerenciada, automática e escalável. Ou seja, é capaz de adaptar-se conforme a necessidade do cliente, para mais ou menos recursos, a fim de suprir ou reduzir a demanda. Na visão do cliente, os recursos da nuvem são sempre ilimitados (HILL et al., 2013; SRINIVASAN et al., 2012).

(e) **Medição de serviço:** A computação na nuvem deve fornecer uma forma de monitorar, controlar e reportar o fornecimento do serviço em um nível de compreensão importante ao cliente, para facilitar o pagamento conforme a demanda de uso (HILL et al., 2013; SRINIVASAN et al., 2012).

As características citadas refletem a definição apresentada pela NIST (HILL et al., 2013). Além das características destacadas anteriormente, a nuvem também é dividida em modelos de serviços e modelos de entrega que impactam na forma de como é ofertado ao cliente que solicita a aquisição de uma nuvem, como apresentado na Figura 3.5.

Os modelos de serviço apresentam o termo "serviço"nos títulos porque é uma representação

do reuso dos diversos componentes disponibilizados na nuvem (VELTE; VELTE; ELSENPETER, 2010).

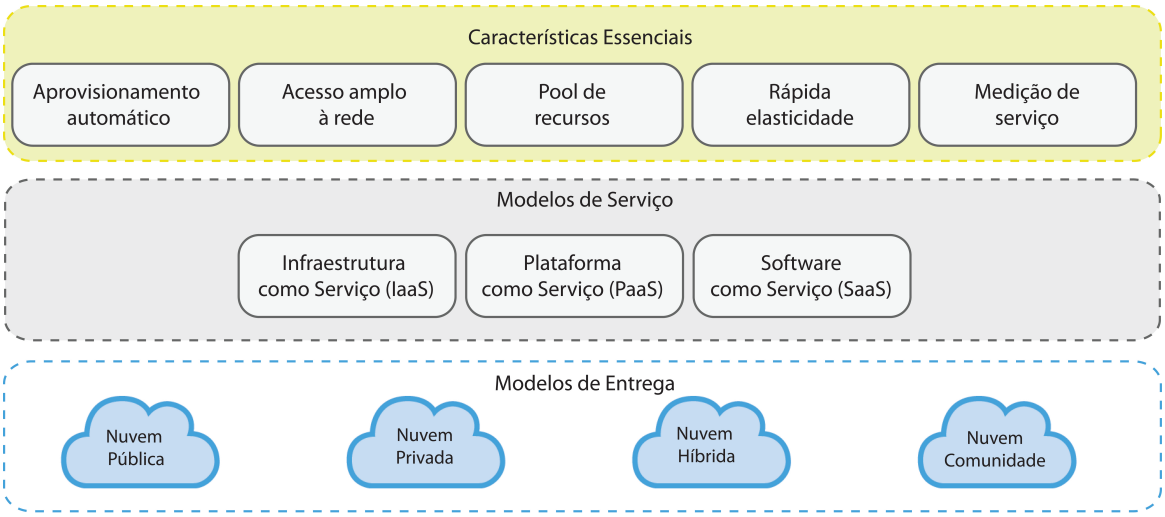

Figura 3.5: Relação de características e modelos. Adaptado de (SRINIVASAN et al., 2012)

O NIST menciona três categorias de modelos de serviços que partem da camada física até a camada de mais alto nível, que são os modelos denominados Infraestrutura como um serviço, Plataforma como um serviço e *Software* como um serviço, que são descritos a seguir:

(a) O **IaaS**, denominado de Infraestrutura como Serviço, é o nível mais baixo dos serviços disponibilizado pela nuvem. Esse nível permite acesso controlado a uma infraestrutura virtual em que o sistema operacional e aplicações podem ser entregues (HILL et al., 2013). Neste modelo de serviço, não há aborrecimentos por parte do cliente em despesas referentes à aquisição e gerenciamento de *hardware*, pois não há controle sobre a infraestrutura física, mas em relação a parâmetros de sistemas operacionais e alguns aspectos de segurança. Os clientes não precisam manter infraestruturas grandes de servidores. Basta escolher os recursos computacionais através do *browser*, já que a nuvem provê recursos computacionais de acordo com a necessidade do cliente pela Internet e sob demanda. (SRINIVASAN et al., 2012).

(b) O **PaaS**, denominado de Plataforma como Serviço, está um nível acima do modelo de serviço IaaS. Esse modelo fornece um ambiente completo para desenvolvimento de *software*, não sendo necessários grandes investimentos em infraestrutura, pois é possível carregar a aplicação na nuvem e pagar conforme o consumo de recursos da plataforma (SRINIVASAN et al., 2012). O controle da infraestrutura está sob responsabilidade do servidor na nuvem e o cliente tem controle sobre o desenvolvimento da aplicação, entrega e configuração dentro

do ambiente de hospedagem (HILL et al., 2013). Google App Engine®, force.com®, Microsoft Windows Azure®, JElastic®, CloudBees® e Cloud Foundry® são exemplos de fornecedores de Plataformas como Serviços.

(c) O **SaaS**, denominado *Software* como Serviço, está um nível acima do PaaS. É nesse modelo que as aplicações são oferecidas como um serviço e os clientes acessam pela Internet. O modelo SaaS não se preocupa com detalhes de infraestrutura e plataforma, concentra-se apenas nos níveis de aplicação que podem ser disponibilizadas em navegadores de Internet ou em aplicações de celulares. O Gmail é um bom exemplo de SaaS, pois uma empresa pode adotar o uso e não precisar se preocupar com manutenção de *hardware*, atualizações de segurança ou mesmo gerenciamento de infraestrutura. A empresa apenas modifica parâmetros pela própria interface *web* de acordo com a necessidade do uso (HILL et al., 2013). O SaaS provê acesso para comercializações de aplicações, pois o *software* é gerenciado de forma centralizada e os clientes acessam aplicações desde que tenham acesso à *web* (VELTE; VELTE; ELSENPETER, 2010).

As características dos modelos de serviço englobam facilidade de adesão por pequenos negócios, alta escalabilidade, multi-inquilinos e independência de dispositivos (VELTE; VELTE; ELSENPETER, 2010).

Os modelos de entrega estão relacionados à forma como a nuvem pode ser disponibilizada para o usuário final, ou seja, no modelo público (*Public Cloud*), ela é gerenciada por uma organização de negócios, acadêmica ou governamental, e liberada para o público em geral; no modelo privado (*Private Cloud*), a nuvem é de uma única organização e de seus consumidores, podendo ser localizada dentro ou fora das instalações do cliente e controlada por terceiros ou pela organização, ou ambos (SRINIVASAN et al., 2012); o modelo comunidade (*Community Cloud*), é para uso de cliente de uma comunidade em particular, a partir de uma organização com visões comuns (SRINIVASAN et al., 2012); e no modelo híbrido (*Hybrid Cloud*), a nuvem resultante é a combinação de mais de um tipo de infraestrutura de nuvem.

Este trabalho adota o modelo de nuvem PaaS, Plataforma como Serviço. Nesse formato de nuvem, o desenvolvedor tem o controle sobre o desenvolvimento, a configuração e a entrega dentro do ambiente de hospedagem. O modelo PaaS permite que o desenvolvedor trabalhe localmente em sua plataforma de desenvolvimento e faça o *deploy* na nuvem conforme as versões vão se tornando estáveis.

3.7 Considerações finais

Apontar quem causa a falha ou relatar se a falha foi provocada por um erro comum ou não previsto é importante quando erros ou invasões acontecem. A auditoria usa trilhas de auditoria (*logs*) para apoiar a identificação de problemas e detectar mau uso, fornecendo evidências e respostas a incidentes e executando análises forenses..

De acordo com Almulla e Yeun (2010), questões relacionadas à preocupação com privacidade aumentam quando os dados são armazenados na nuvem. A nuvem deve preservar a integridade dos dados e a privacidade do usuário através dos múltiplos provedores de serviços. A rede, a hospedagem e as aplicações necessitam apresentar algum nível de auditoria para analisar respostas a incidentes e análises forenses e em cada nível é necessário satisfazer requisitos de segurança para preservar a confidencialidade, a integridade e a disponibilidade de dados. Auditoria é um elemento presente no gerenciamento de identidade de acesso que ajuda a revisar e examinar a autorização e a autenticação de registros para validar com as conformidades de segurança pré-definidas, além de ajudar a identificar brechas de segurança (ALMULLA; YEUN, 2010).

Hamlen et al. (2011) mostra que uma das áreas críticas que precisam de atenção é a de gerenciamento de identidade, pois o usuário tem múltiplas identidades que podem ser usadas através de um domínio federado que precisa manter e gerenciar. O sistema de gerenciamento de identidade precisa ter a aplicação da auditoria, uma das características a ser destacada conforme Hamlen et al. (2011) . Em sistemas distribuídos, a auditoria é um desafio, pois deve considerar quantos dados auditáveis precisam ser coletados e quais técnicas podem ser usadas para analisar os dados. Na nuvem, esse desafio é alto, porque os recursos são alocados dinamicamente e podem ser requisitados por qualquer usuário..

Spencer (2012) classifica sistema de gerenciamento de identidade em cinco partes: autenticação, autorização, gerenciamento de conta, *log* de auditoria e federação. Na parte de auditoria, busca-se rastrear o acesso para prevenir incidentes de segurança.

No mecanismo desenvolvido neste estudo, a auditoria é aplicada para aumentar a proteção de acesso ao dado liberado por aplicações a um determinado público. Para isso, a identidade é solicitada e o acesso ao dado é auditado, oferecendo um cenário geral dos acessos.

O capítulo apresentou uma visão sobre como auditar com garantia de privacidade focando no funcionamento da auditoria de pseudônimo e das adaptações realizadas para atender o trabalho. Por fim, apresentou uma visão geral do que é computação na nuvem, características, modelos de serviços, modelos de entrega e a motivação da escolha da computação na nuvem

como ambiente de desenvolvimento do mecanismo proposto no projeto.

Capítulo 4

TRABALHOS RELACIONADOS

O capítulo apresenta trabalhos envolvidos no mesmo contexto deste projeto, que focam a auditoria no ambiente de computação em nuvem ou trabalhos que mostram a questão do acesso ao dado em um ambiente em nuvem.

4.1 Considerações iniciais

O mecanismo em nuvem de monitoramento a dados sensíveis propõe uma nova abordagem de um software como serviço para auditar o acesso ao dado com garantia de privacidade baseado em uma solução na nuvem. O serviço é uma ferramenta que visa aumentar o controle do acesso ao dado definido como sensível pelo proprietário do dado.

Foram pesquisados trabalhos com foco nos dados sensíveis, prevenção de mau uso, auditoria para evitar o mau uso e o modo como a computação na nuvem lida com os dados sensíveis.

É possível perceber que a abordagem do mecanismo proposto difere de outros trabalhos, corroborando suas contribuições.

No decorrer deste capítulo, é apresentada uma visão geral de alguns artigos e feita uma comparação com o mecanismo aqui proposto. Os artigos pesquisados são: *"A Monitoring and Audit Logging Architecture for Data Location Compliance in Federated Cloud Infrastructures"* Massonet et al. (2011); *"Cloud Verifier: Verifiable Auditing Service for IaaS Clouds"* Schiffman et al. (2013); *"Design and auditing of Cloud computing security"* Gowrigolla, Sivaji e Masillamani (2010); *"Providing privacy preserving in Cloud computing"* Wang et al. (2010); *Novel Data Protection Model in Healthcare Cloud* Chen e Hoang (2011); *Fog Computing: Mitigating Insider Data Theft Attacks in the Cloud* Stolfo, Salem e Keromytis (2012); *Promoting Distributed Accountability in the Cloud* Sundareswaran et al. (2011); *Tracking of Data Leaving*

the Cloud Tan et al. (2012); *Cloud computing: security risk* Sumter (2010); *Accountability for cloud and other future Internet services* Pearson et al. (2012); *Policing as a Service in the Cloud* Zargari e Smith (2013) e *Simultaneously Supporting Privacy and Auditing in Cloud Computing Systems* Grandison, Thorpe e Stenneth (2013).

4.2 Projetos

Recentemente, estudos têm sido feitos focando na proteção de dados na nuvem, como a criação de uma arquitetura de auditoria (MASSONET et al., 2011); auditoria para aumentar a confiança do usuário na nuvem (SCHIFFMAN et al., 2013); arquitetura para garantir a integridade dos dados na nuvem (GOWRIGOLLA; SIVAJI; MASILLAMANI, 2010); algoritmos para anonimizar dados para os serviços de computação na nuvem (WANG et al., 2010); auditoria para reforçar a segurança de aplicações na nuvem (CHEN; HOANG, 2011); detecção de acesso anormal por meio de monitoramento na nuvem (STOLFO; SALEM; KEROMYTIS, 2012); rastrear o uso do dado para evitar o mau uso na nuvem (SUNDARESWARAN et al., 2011; TAN et al., 2012); monitorar o acesso de dado na nuvem (STOLFO; SALEM; KEROMYTIS, 2012; SUMTER, 2010); como os servidores da nuvem usam o dado (PEARSON et al., 2012; ZARGARI; SMITH, 2013) e privacidade e auditoria na nuvem (GRANDISON; THORPE; STENNETH, 2013).

No artigo *A Monitoring and Audit Logging Architecture for Data Location Compliance in Federated Cloud Infrastructures*, de Massonet et al. (2011), a auditoria é utilizada para verificar se o sistema como um todo está cumprindo as exigências do usuário. Assim, uma arquitetura de *logging* capaz de rastrear um conjunto completo de operações é proposta, permitindo ao usuário conhecer como o servidor de infraestrutura gerencia o dado e o local de execução do dado durante o ciclo de vida.

Diferente do trabalho de Massonet et al. (2011), o mecanismo rastreará o usuário quando o dado acessado for considerado sensível. Dessa forma, o proprietário do dado conhecerá melhor sobre qual dado, quando, quem e a partir de onde o acesso é realizado.

Os autores Schiffman et al. (2013), no artigo *Cloud Verifier: Verifiable Auditing Service for IaaS Clouds*, trabalham baseados na falta de transparência na nuvem, a qual conduz o cliente à insegurança em relação aos dados sensíveis do usuário e do processamento nesse ambiente. Um *framework* aos fornecedores de nuvem é proposto pelos autores para prover serviço de monitoramento na nuvem para os clientes validarem o ambiente e executarem como esperado em uma nuvem IaaS.

No projeto de Schiffman et al. (2013), o *framework* é dividido em duas partes: a primeira

parte é chamada *Cloud Verifier*, que permite monitorar a saúde da sua própria instância, e a segunda parte é definida como *Instance Monitor*, que é um serviço para monitorar a instância para detectar violações de acordo com as exigências dos clientes. Os autores acreditam que esse *framework* é uma maneira eficiente para assegurar que a instância está sendo executada de acordo com as exigências dos clientes.

A diferença em comparação com o trabalho de Schiffman et al. (2013) é que a atenção do mecanismo em nuvem de monitoramento a dados sensíveis está sobre os dados da aplicação. O mecanismo não se concentra em compreender a saúde da instância em que o aplicativo está incidindo, ou se o aplicativo recebe algum ataque. Mas, o serviço garante ao proprietário dos dados uma visão melhor de quem está acessando os dados liberados a partir da aplicação.

No artigo *Design and auditing of Cloud computing security*, de Gowrigolla, Sivaji e Masillamani (2010), o uso de auditoria promove a integridade do dado através do esquema de auditoria pública e os elementos que garantem a privacidade do dado mantendo a funcionalidade do sistema estão presentes na arquitetura. Dois problemas são abordados: proteger dados que entram na nuvem e fornecer a integridade do dado protegido. A diferença em comparação com o trabalho de Gowrigolla, Sivaji e Masillamani (2010) é que o mecanismo está preocupado com a privacidade do usuário que acessa o dado para evitar pôr em risco o usuário sem uma análise clara da conduta de acesso. Então, não foi utilizado um esquema de auditoria pública , mas uma adaptação da auditoria pseudônimo.

O artigo *Providing privacy preserving in Cloud computing*, de Wang et al. (2010), propõe um novo algoritmo de anonimato para os serviços de computação em nuvem. O novo algoritmo processa o microdado antes de ser publicado e então envia o dado para o provedor de serviços de nuvem. A diferença em comparação com o trabalho de Wang et al. (2010) é que o mecanismo mantém a privacidade do usuário usando uma adaptação da auditoria de pseudônimo. O pseudônimo nessa técnica é importante, e é usado para manter o sigilo dos dados que revelam a informação sensível do usuário.

O trabalho de Chen e Hoang (2011), denominado *Novel Data Protection Model in Healthcare Cloud*, mostra o *framework* CPRBAC, que aborda os desafios de segurança e privacidade na área de saúde na nuvem. A fim de aumentar a segurança nesse ambiente, é executada na base uma auditoria ativa com a qual é possível monitorar e relatar a operação ilegal. No artigo de Chen e Hoang (2011), é possível identificar uma similaridade de conceito, porque a ideia de usar auditoria para melhorar a segurança do ambiente está presente em ambas. Mas, no mecanismo, a ideia é melhorar o controle de acesso ao dado sensível não focando na operação que acontece no ambiente.

No artigo *Promoting Distributed Accountability in the Cloud*, de Sundareswaran et al. (2011), os autores estão preocupados com a perda de controle quando os dados do usuário estão na nuvem. Então, os autores fizeram uma prestação de contas distribuídas na nuvem usando a capacidade de programação de um JAR permitindo um mecanismo de logging com o dado e uso de políticas ao enviar o dado para a nuvem. O acesso ao dado inicializa um mecanismo de registro de autenticação local automatizado.

A diferença em comparação com o trabalho de Sundareswaran et al. (2011) é que o mecanismo em nuvem de monitoramento a dados sensíveis se preocupa com a perda de controle sobre os dados liberados pela aplicação. É uma abordagem diferente em relação a este e outros trabalhos. Todos estão preocupados com o conteúdo e os dados considerando-se a ponto do prestador de serviços em nuvem, por exemplo. O mecanismo vem e acrescenta uma nova perspectiva de um possível uso indevido do usuário que tem o direito de acesso aos dados sensíveis. Uma ferramenta para melhorar essa perspectiva como um software como serviço pode melhorar a segurança de acesso ao dado em serviços na nuvem.

Os autores Tan et al. (2012), no artigo *Tracking of Data Leaving the Cloud*, fizeram uma proposta fundamentada de que o vazamento de dados é um problema crítico para os usuários finais e provedores de serviços em nuvem, independentemente se o vazamento está dentro ou fora da nuvem. A auditoria na proposta é usada para rastrear os dados que saem da nuvem. Assim, um *framework* para prestação de contas de dados chamado *CloudDT* é criado e captura ações, eventos ou modificações e acessa os dados dentro e fora da nuvem.

A diferença em comparação com o trabalho de Tan et al. (2012) é que o mecanismo quer controlar o acesso do usuário. Para isso, é necessário chamar o serviço dentro da aplicação. As ações, eventos e modificações nos dados não são o ponto central nessa primeira versão, mas o acesso dos dados que precisam ser liberados para algum público e precisam de atenção na liberação.

Em estudos dos autores Stolfo, Salem e Keromytis (2012), do artigo *Fog Computing: Mitigating Insider Data Theft Attacks in the Cloud*, o foco é o ataque que a nuvem pode sofrer. Para isso, propõe-se o monitoramento de acesso a dados para lançar um ataque de desinformação protegendo os dados originais de mau uso quando um comportamento anormal é detectado.

Ainda sobre o controle da nuvem, o artigo *Cloud computing: security risk*, de Sumter (2010), propõe uma maneira de o usuário monitorar as informações sobre o servidor e garantir que as informações vitais não foram adulteradas ou mal utilizadas. Esse tipo de monitoramento diminuirá a dúvida sobre se a informação está segura ou não. Além disso, está inclusa nesse mecanismo de monitoramento uma maneira para que o usuário faça uma consulta usando o

Secure Watch Agent, que investiga e responde a pergunta do usuário.

A diferença, em comparação com os trabalhos de Stolfo, Salem e Keromytis (2012), Sumter (2010), é que o mecanismo busca utilização indevida com base no cenário de detecção disponível no serviço. Os cenários inicialmente apresentam ideia sobre a frequência do acesso, os dados mais acessados e assim por diante.

Entender como o serviço na nuvem pode gerenciar informações pessoais, sensíveis e confidenciais é outra tendência que está presente no estudo *Accountability for cloud and other future Internet services*, dos autores Pearson et al. (2012). O estudo de Pearson et al. (2012) relata a criação do projeto A4Cloud, que permitirá soluções para ajudar os usuários a decidir e acompanhar como os seus dados são utilizados pelo prestador de serviços de nuvem.

O foco do A4Cloud é a responsabilidade de promover um controle eficaz para os dados corporativos e privados processados na nuvem. A prestação de contas do *framework* proposto permitirá agir de maneira preventiva, detectiva e corretiva.

Outros estudos na mesma linha de pensamento sobre como o provedor de serviços de nuvem lida com os dados do usuário são apresentados no artigo *Policing as a Service in the Cloud*, de Zargari e Smith (2013), que fornece policiamento como um serviço. O serviço apresentado em Zargari e Smith (2013) será um passo fundamental para gerar confiança na nuvem. Será possível obter informações básicas dos clientes sobre quantas cópias de dados do usuário estão disponíveis, serão geradas notificações sobre acesso a dados e será também possível permitir a análise de metadados. Nesse artigo ainda é discutida a ideia de proporcionar privacidade e forense como um serviço tais como um sub-serviço.

A diferença em relação aos trabalhos de Pearson et al. (2012), Zargari e Smith (2013) é que o mecanismo está focado sobre os dados utilizados pela aplicação e como os dados são acessados por outras pessoas que de alguma forma têm o direito de acesso, mas podem ter alguma má intenção. Nos trabalhos de Pearson et al. (2012), Zargari e Smith (2013), a ideia está relacionada com o provedor de serviços em nuvem, ao contrário do mecanismo que insere uma nova perspectiva sobre a questão de dados.

O artigo *Simultaneously Supporting Privacy and Auditing in Cloud Computing Systems*, de Grandison, Thorpe e Stenneth (2013), foca em garantir a auditoria e a privacidade, apresentando uma camada entre o cliente e o fornecedor de serviços de nuvem, que garante que o CSP armazene o dado sem obter informações confidenciais sobre o cliente.

Todos os pedidos para o CSP são interceptados por uma camada sobre a nuvem que se aplica às funções de preservação da privacidade antes de enviar para a API nativa na nuvem. A

parte de auditoria do esquema ocorre de forma física independente por meio da qual o usuário pode enviar pedido para o auditor, que irá solicitar registros CSP e irá retornar com parâmetros sensíveis criptografados, analisar e divulgar os resultados.

A diferença em comparação com o trabalho de Grandison, Thorpe e Stenneth (2013) é que o mecanismo está focado em fornecer uma maneira de o proprietário do dado responsabilizar alguém pelo acesso feito aos dados liberados.

4.3 Considerações finais

Em uma visão geral, a contribuição do mecanismo em relação aos trabalhos mencionados é que o mecanismo em nuvem de monitoramento a dados sensíveis irá reforçar a segurança do acesso ao dado através da auditoria, promovendo seu monitoramento. O serviço disponibilizado delega poder ao proprietário do dado para rastrear o acesso ao dado que é considerado sensível segundo o ponto de vista do proprietário.

Em outras palavras, dados sensíveis podem ser liberados e o mecanismo pode revelar a frequência de acesso ao dado do proprietário, por exemplo. Isso é possível porque o mecanismo oferece tal tipo de informação. Com o mecanismo é possível prevenir o mau uso e garantir compreensão sobre o acesso para propor melhoria nas políticas de liberação de dados, considerando um ambiente tal como o exemplo do Portal da Transparência, do governo brasileiro. Ao mesmo tempo, o serviço garante a privacidade preservando a identidade do usuário que realizou o acesso, usando técnicas relacionadas à auditoria de pseudônimo.

Capítulo 5

MECANISMO EM NUVEM DE MONITORAMENTO A DADOS SENSÍVEIS

O capítulo apresenta os requisitos identificados para o desenvolvimento do serviço Log-Cloud. É apresentada uma visão geral e o que se pretende com o serviço. Outros pontos tais como a proposta conceitual, implementação e tecnologias utilizadas para o desenvolvimento do software como serviço para auditar o acesso ao dado com a garantia de privacidade são tópicos presentes neste capítulo, que ainda contempla detalhes sobre como deve ser realizada a invocação do mecanismo dentro de aplicações e qual o padrão definido para identificar os dados sensíveis.

5.1 Considerações iniciais

Este trabalho propõe uma nova abordagem do uso da auditoria em relação a trabalhos anteriores. A ideia é usar a auditoria e redirecioná-la para melhorar o controle sobre o acesso ao dado, gerando um cenário sobre os acessos que vão guiar o proprietário do dado na definição de políticas de liberação ou identificação de mau uso.

O mecanismo em nuvem de monitoramento a dados sensíveis é ofertado ao usuário final como um serviço. Ou seja, a ideia é que se torne um SaaS, em que a funcionalidade principal é incrementar o controle de acesso de um aplicativo que precisa liberar os dados sensíveis a um determinado público.

Para a validação da proposta, é usada a implementação do mecanismo e de um protótipo do Portal da Transparência do Brasil, que, conforme vimos, oferece acesso aos cidadãos a dados sensíveis dos servidores federais, a fim de promover ações de transparência do governo brasileiro.

A ideia deste serviço pode ser estendida para outro tipo de aplicação que necessita de algum tipo de controle, como organizações que precisam lidar com o acesso a dados sensíveis por empregados através da web em sistemas de planejamento de recursos empresariais (ERP). Além do controle de acesso liberado ao empregado, o SaaS oferece ao proprietário do dado um controle sobre essa liberação, por exemplo, remetendo ao estudo do controle paradoxo mencionado por Brandimarte, Acquisti e Loewenstein (2013).

Para realizar o monitoramento do acesso ao dado é necessário compreender o que é importante coletar no ambiente em que a auditoria vai ser aplicada (FISCHER-HÜBNER, 2001). Não só as ações realizadas pelo indivíduo que acessa precisam ser coletadas, mas também quem está acessando os dados visualizados, quando o acesso é feito e de onde ele vem.

A análise dos *logs* criados é o ponto principal de uma ação de auditoria e com os logs pode ser possível extrair frequência de acessos a dados sensíveis e que tipo de dados estão sendo acessados e construir um mapa do acesso considerando as origens de acesso.

Analisando todos esses possíveis requisitos, algumas metas foram pré-definidas para tornar possível disponibilizar um mecanismo que permita auditar o acesso ao dado mantendo a privacidade e fornecendo uma forma flexível de ser embutido em qualquer aplicação.

No decorrer deste capítulo, são apresentados os requisitos identificados para desenvolvimento de um serviço que possibilite a auditoria e a prestação de conta. A visão conceitual do serviço proposto é apresentada, incluindo a padronização para mapeamento dos dados sensíveis, além dos serviços que compõem a nuvem. Apresenta-se a visão da implementação do serviço, que é dividido em um *core* principal disponibilizado na nuvem e em um plugin a ser embutido na aplicação que precisa ter o acesso ao dado sensível auditado.

5.2 Requisitos

O serviço permite um monitoramento do acesso ao dado sensível liberado ao público pelo proprietário do dado e consequentemente uma análise dos logs dos acessos realizados, caracterizando a auditoria dos logs. O log gerado mantém a privacidade do acesso realizado pela pessoa.

No contexto deste mecanismo é considerado administrador do sistema o proprietário do dado, sendo este o único a ter acesso e interagir com o cenário de detecção de intrusão disponibilizado no serviço. O proprietário do dado refere-se ao indivíduo que disponibiliza o acesso ao dado. Com base no estudo de caso pode-se entender que o proprietário do dado seria o governo,

ou no caso de uma empresa privada o administrador geral do sistema.

A pessoa que acessa o dado sensível e consequentemente tem o seu acesso protegido pode ser considerado um cidadão se analisarmos sob a perspectiva do estudo de caso que seria o portal da transparência. Ou então, pode ser um funcionário de uma empresa se o mecanismo for aplicado em outro tipo de sistema. Assim, no contexto desta especificação considera-se a pessoa que acessa o dado sensível o cidadão ou funcionário. Por fim, o último papel trata-se a do desenvolvedor da aplicação que interage com o SaaS através do uso do plugin LogCloud.

A proposta desse serviço fundamenta-se na necessidade de algumas aplicações de realizar o monitoramento ao acesso a determinado dado que pode ser considerado sensível, dependendo do julgamento do proprietário do dado, em vista de questões relevantes abordadas anteriormente neste trabalho (ver Capítulo 2).

Tal como um serviço, o mecanismo que permite auditar o acesso com a garantia de privacidade tem como propósito ser integrado a aplicações que necessitem de um controle sobre o que é liberado pelo proprietário do dado. Assim, o mecanismo como um serviço (SaaS) pode ser chamado por qualquer outra aplicação disponibilizada na nuvem ou não. Para o desenvolvimento do serviço foram listados requisitos que apresentam a essência da funcionalidade a ser alcançada. Em uma visão do todo, o serviço contempla os seguintes requisitos funcionais e não funcionais:

(a) protege a identidade de quem acessa o dado sensível. Para isso, a identidade real do cidadão/funcionário é substituída(o) por um pseudônimo gerado randômicamente a cada novo acesso à aplicação;

(b) garante uma forma menos intrusiva no código da aplicação quando for invocado o seu uso, pois a aplicação envia ao serviço disponibilizado na nuvem os dados sensíveis e o pseudônimo usado para acessar o dado rotula os dados sensíveis com base na política de dados do proprietário do dado;

(c) gera os logs ao acessar os dados definidos como sensíveis. A geração de logs acontece no repasse dos dados ao serviço na nuvem, formados pelo pseudônimo, dados sensíveis e informações sobre o acesso (data de acesso e localização). A identificação real do cidadão/-funcionário que acessou é realizada com base na reversão do pseudônimo gerado;

(d) recebe os logs a serem armazenados. Assim, uma forma de analisar todo o conteúdo coletado é disponibilizada ao proprietário do dado por meio de cenários de detecção. No caso, o serviço disponibiliza uma área administrativa ao proprietário do dado para que ele

possa visualizar aspectos tais como: quantidade de acesso em relação ao contexto a que o dado está relacionado, i.e., o dado sensível é relacionado a dados financeiros e informações pessoais; visão geral em relação à quantidade de dados sensíveis acessados; um rank dos dados sensíveis mais acessados; de onde o acesso ao dado sensível veio quando for possível identificá-lo e uma forma para o proprietário do dado analisar todos os dados acessados com a possibilidade de revelar a identidade de quem acessa;

(e) garante impacto mínimo ao ser invocado por aplicações que auditam o acesso ao dado. Então, no contexto em que é aplicada a invocação do mecanismo de auditoria, não prejudica a interação ao visualizar os dados sensíveis.

O agrupamento das funcionalidades em visões, apresentado a seguir, facilita a compreensão de como o serviço deve se comportar em relação ao **cidadão/funcionário que acessa o dado sensível**, como o serviço deve se comportar em relação ao **proprietário do dado sensível** e como o serviço deve ser invocado pelo **desenvolvedor da aplicação que precisa ter acesso a dados auditado**.

Visão do cidadão/funcionário ao acessar o dado sensível:

(a) O serviço solicita dados pessoais do cidadão/funcionário que acessa as aplicações que auditam o acesso ao dado sensível. A ideia não é retirar controle de acesso ou autenticação com o uso deste serviço, mas sim garantir mais um controle. Então, dados sobre o acesso para possibilitar a sua futura identificação devem ser informados antes de se liberar o acesso à área com dados sensíveis. As aplicações precisam coletar dados elementares tais como nome completo, número de algum documento identificador, email e nome.

Visão do desenvolvedor da aplicação ao invocar o serviço:

(a) O serviço recebe dados sensíveis mapeados conforme a política criada pelo proprietário do dado. O desenvolvedor das aplicações, ao invocar o serviço, precisa mapear os dados sensíveis a serem enviados ao serviço na nuvem. Com o mapeamento desses dados, torna-se possível gerar o log e consequentemente gerar os cenários de detecção de intrusão para entender as frequências de acesso e local de origem do acesso. Um padrão com base em XML e a definição acerca sobre o que são dados sensíveis focada no escopo a ser alcançado por este serviço são detalhados na Seção 5.3.1 deste capítulo.

Visão do proprietário do dado sensível:

(a) O serviço é fornecido para vários proprietários de dados. Então, um código de acesso (*token*) precisa ser atribuído a cada proprietário do dado, de forma única, ao assinar o uso desse serviço por meio de um cadastro com dados básicos de identificação do proprietário do dado tais como nome, email e senha (acesso área administrativa). O *token* a ser informado ao proprietário do dado deve ser associado a cada log de uma determinada aplicação. Sendo assim, o *token* refere-se a uma aplicação de um proprietário de dado e identifica que os registros de logs gerados referem-se a uma aplicação.

(b) O serviço disponibiliza uma área para analisar os logs gerados dentro de cenários de detecção de intrusão. Tais cenários são disponibilizados ao proprietário do dado para consultar com base em visões dos cenários pré-definidas e o comportamento dos acessos realizados.

A união das visões reflete o comportamento implementado no serviço de auditoria e prestação de contas que permite auditar o acesso ao dado com garantia de privacidade, chamado LogCloud.

5.3 Visão conceitual do LogCloud

O LogCloud funciona conforme o cenário em alto nível apresentado na Figura 5.1. Na Figura 5.1 é possível ver a interação do cidadão/funcionário com uma aplicação que, neste exemplo, está disponível em uma nuvem, a qual utiliza o mecanismo que permite auditar o acesso ao dado provendo um monitoramento das pessoas que acessaram dados julgados sensíveis pelo proprietário.

Na Figura 5.1, é possível observar que o cidadão/funcionário deve se identificar para acessar a aplicação, pois a identidade é repassada para o mecanismo, que retorna um pseudônimo. Depois que o acesso é liberado ao cidadão/funcionário, este pode navegar pelos dados liberados pela aplicação.

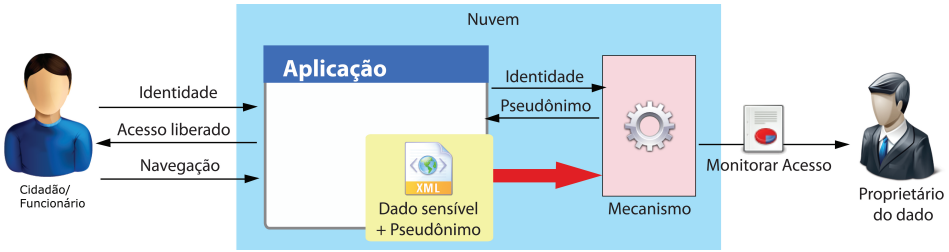

Figura 5.1: Visão geral do funcionamento do serviço com o cidadão/funcionário.

A Figura 5.2 apresenta como o LogCloud comporta-se quando o cidadão/funcionário faz um acesso em uma aplicação que usa o serviço. Na Figura 5.2, é possível visualizar duas

organizações, A e B, sendo que cada uma está usando o LogCloud, mas as aplicações são disponibilizadas para o cliente de formas diferentes.

Na organização A, a aplicação é disponibilizada em uma nuvem e, na organização B, o aplicativo é disponibilizado em um servidor de uma companhia, por exemplo. Então, ambas as formas permitem o uso do serviço e a análise, pelo respectivo proprietário do dado, dos logs através de gráficos e registros disponíveis nos cenários de detecção.

As aplicações que usam o LogCloud devem definir quais dados precisam ter o seu acesso auditado. Ou seja, durante o desenvolvimento das páginas com conteúdo sensível de um determinado aplicativo, o desenvolvedor precisa definir se o acesso a determinado dado precisa ser auditado ou não. Caso o acesso precise ser auditado, devem ser repassados os dados sensíveis juntamente com o pseudônimo, conforme esboçado na 5.1.

Todo acesso a dados sensíveis repassado para o serviço gera log que alimenta os cenários de detecção de intrusão. A intrusão no contexto deste trabalho é algum tipo de comportamento a ser definido pelo proprietário do dado como mau uso, com base nos cenários disponíveis na área administrativa do serviço. A área administrativa é acessada pelo proprietário do dado com o nome do usuário e a senha informados quando foi realizado o cadastro de uso do serviço.

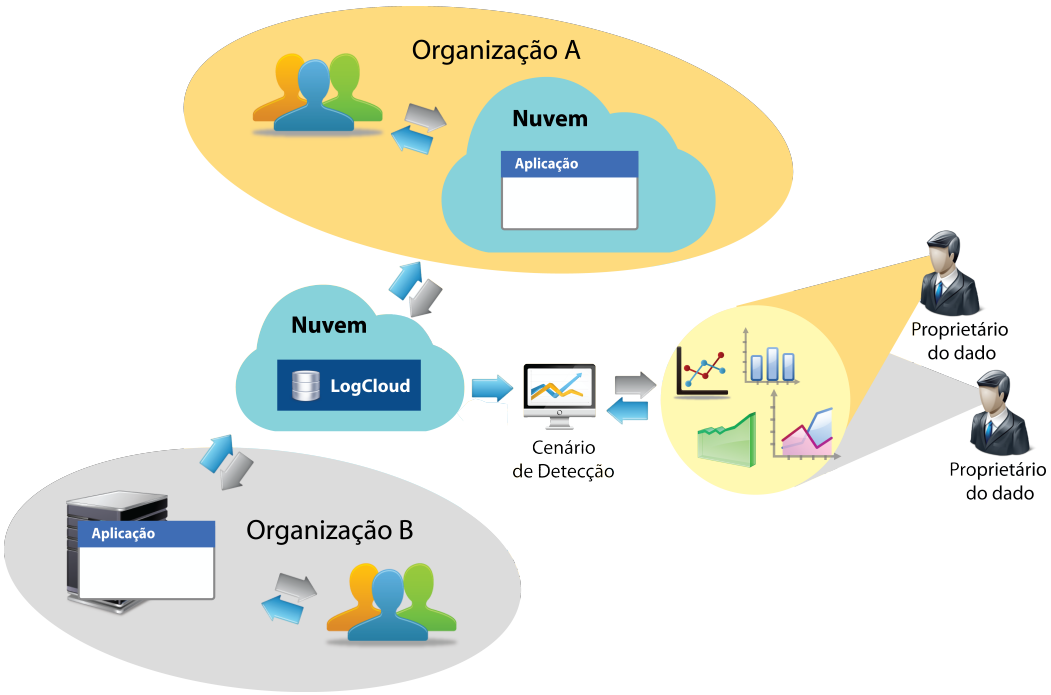

Figura 5.2: Visão geral do funcionamento do serviço com o ambiente.

O serviço é formado por componentes responsáveis por gerar o pseudônimo, gerar logs de auditoria e prover meios de consultar os logs através de análises dos cenários de detecção de intrusão liberados na área administrativa do proprietário do dado.

O princípio é ofertar o mecanismo como um serviço (SaaS) que pode ser invocado por outra aplicação com apenas algumas inserções de código na aplicação que o requisita, garantindo a interoperabilidade entre os sistemas, pois a passagem de dados sensíveis é realizada por meio do envio de dados sob o padrão XML, e o serviço na nuvem recebe tais dados através de um *webservice*, que segue os princípios do estilo arquitetural RESTful (*Representational State Transfer*).

O serviço é um *software* instalado em um ambiente computacional com acesso à rede e é consumido por uma aplicação que passa a ser cliente desse serviço. Logo, a descrição do serviço revela detalhes da interface e implementação incluindo tipos de dados, informação de ligação e localização de rede (OLIVEIRA, 2012).

Para garantir a interoperabilidade que caracteriza a capacidade de um sistema de se comunicar com outro de forma transparente, é adotada a implementação do mecanismo como um *webservice*. As tecnologias mais utilizadas para integração entre aplicações heterogêneas são os *web services* que, devido aos padrões abertos, realiza o intercâmbio de dados de uma forma mais transparente.

Segundo a W3C (W3C, 2012), o *web service* é projetado para interações entre máquinas-para-máquinas sobre uma rede de forma interoperável. Para o desenvolvimento deste mecanismo, é adotado o *web service* RESTful (*Representational State Transfer*), que se refere a um estilo arquitetural para construção de sistemas distribuídos, definido por Roy Thomas Fielding, simplificando o desenvolvimento de serviços *web* (OLIVEIRA, 2012).

Os padrões amplamente usados – tais como HTTP, XML, URI, entre outros – foram empregados para o desenvolvimento do RESTful, evitando o excesso de padronização (OLIVEIRA, 2012). Além disso, a característica da arquitetura RESTful é o recurso: uma página *web*, figura, imagem ou vídeo. A localização de tais recursos dentro de uma interação entre os componentes da arquitetura é feita pelo chamado identificador de recursos. Logo, os recursos podem ser representados em diferentes formatos tais como HTML, XML, JSON, entre outros (OLIVEIRA, 2012).

O *web service* RESTful é considerado simples devido às seguintes características: aproveita os padrões conhecidos da W3C/IETF (HTTP,XML,URI,MIME), a infraestrutura necessária já se tornou generalizada; os clientes HTTP e servidores estão disponíveis para a maioria das linguagens de programação/ sistemas operacionais / plataformas de *hardware* e a porta 80 é geralmente deixada como padrão pelo *firewall*. Assim, uma infraestrutura leve em que serviços são desenvolvidos com mínimas ferramentas tem uma barreira de adoção baixa (PAUTASSO; ZIMMERMANN; LEYMANN, 2008).

Com base na estrutura fundamentada em serviço, foi criado um diagrama de sequência, apresentado na Figura 5.3. Na Figura 5.3, é esboçado um diagrama com a invocação dos serviços existentes no SaaS criado para cumprir a finalidade de anonimizar o cidadão/funcionário com o uso do pseudônimo e de geração de logs.

O diagrama de sequência apresenta a troca de mensagens entre a aplicação e o serviço disponível na nuvem. Analisando o fluxo, pode-se perceber a necessidade da aplicação de solicitar quem está realizando o acesso. Ou seja, o serviço tem que ser usado em conjunto com os outros controles já existentes na aplicação, ou então criar uma área de autenticação antes de entrar na área onde os dados estão mapeados como sensíveis. A forma como vão ser passados os dados do cidadão/funcionário é de responsabilidade do desenvolvedor e não do serviço, que apenas recebe e anonimiza os dados.

Na sequência, a aplicação do desenvolvedor valida o acesso. Por fim, observa-se que, ao clicar no dado mapeado como sensível disponibilizado em alguma página do aplicativo é disparada a requisição aos serviços do LogCloud. Na requisição de serviços ocorre a geração de um pseudônimo, e na sequência, a geração do log com o pseudônimo, alimentando os cenários de detecção de intrusão.

A interação do cidadão/funcionário, como apresentado na Figura 5.3, pode resultar no envio de dados sensíveis ao serviço disponibilizado na nuvem para gerar os logs. A invocação só é realizada porque, na página da aplicação que contém dados sensíveis, eles foram mapeados previamente pelo desenvolvedor.

5.3.1 Mapeamento de dados sensíveis

O mecanismo disponibilizado na forma de serviço visa aumentar a proteção a dados liberados pelo proprietário através da coleta de log dos acessos dos dados rotulados como sensíveis. Assim, cria-se um monitoramento de acesso aos dados compreendendo quem, quando e de onde tais acessos são realizados.

Um dos desafios quando se propõe a construção de um mecanismo de auditoria de acesso ao dado como este está em como mapear o dado sensível pelo proprietário. A primeira observação é o que pode ser definido como dado sensível no contexto do que precisa ser monitorado.

O que foi apresentado até agora aponta uma preocupação em relação ao dado que revela algum aspecto relacionado a informações financeiras, pessoais e estratégicas disponíveis em um sistema de uma organização, como a folha de pagamento de empregados, disponível no Portal de Transparência do Brasil.

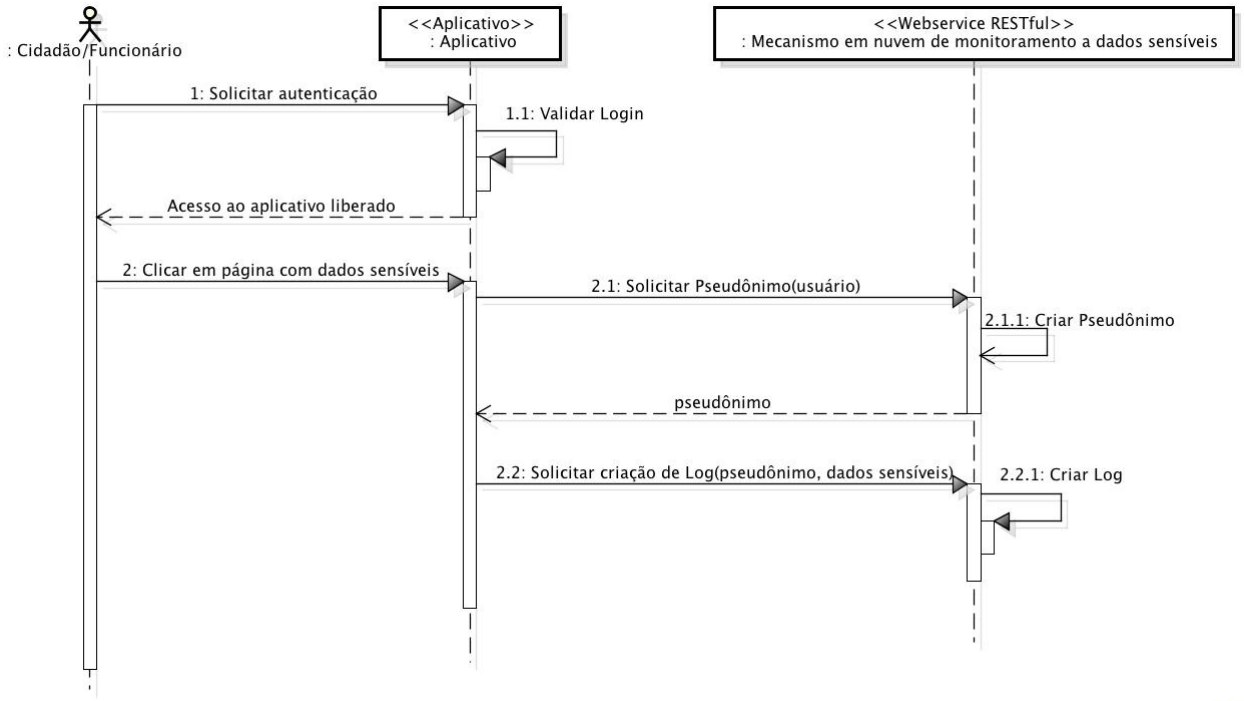

Figura 5.3: Diagrama de sequência com as requisições de serviço.

Com esses cenários apresentados, pode-se observar a relação entre revelar a informação sobre alguém que não pode interferir no sistema para remover o dado ou proibir o acesso. Assim, possuir certo controle para monitorar o acesso ao dado pode dar alguma segurança para indivíduos que tem o dado sensíveis liberados, aumentando o controle sobre sua liberação.

Assim, a existência de proprietários do dado, além da organização ou governo que detém o dado está clara. Isto é, o dado sensível que revela informação sobre uma pessoa está em posse de um outro ente, que pode ser uma organização ou o governo. Por sua vez, esse ente tem o direito de liberar o dado a outras partes sempre que desejar, como acontece com o Portal da Transparência brasileiro.

Então, com base na compreensão desse cenário inicial, é definida uma estrutura a qual tem um proprietário com os dados acessados. A fim de compreender a forma como os dados são mapeados, uma estrutura XML é apresentada no Código 5.1.

Código 5.1: XML do log a ser enviado ao serviço

```
1 <log>
2         <pseudonym>...</pseudonym>
3         <ip>...</ip>
4         <device>...</device>
5         <owner>...</owner>
6         <date>...</date>
7         <hour>...</hour>
8         <context>...</context>
9         <logFields>
```

```
10                  <data>
11                          <field></field>
12                          <value></value>
13                  </data>
14                  ...
15          </logFields>
16          <token>...</token>
17  </log>
```

Neste ponto, é possível compreender que cada requisição a qualquer dado mapeado como sensível disponível no aplicativo que está usando o LogCloud vai desencadear uma ação que envia o XML para a nuvem, apresentado no Código 5.1 .

O XML não é limitado a um conjunto de *tags*, pois ele permite a criação de *tags* personalizadas. As vantagens em seu uso estão relacionadas ao envio de qualquer informação e à facilidade de integração, pois tem um mapeamento mais genérico. Além disso, trata-se de um padrão aberto com o qual a empresa pode definir quais *tags* podem ser criadas e o significado de cada uma (POWELL, 2007).

Os dados sensíveis são mapeados no aplicativo e enviados ao mecanismo através de um padrão XML. O mapeamento de dados no formato em XML visa coletar quem acessa o dado sensível e qual dado é acessado.

Analisando a criação de estrutura XML, é possível perceber que o campo OWNER se refere à pessoa a quem os dados realmente pertencem; o campo IP refere-se ao local de onde o acesso veio; o campo LOGFIELDS refere-se à quantidade de dados coletados pertencentes a esse usuário; o campo DATE refere-se à data de acesso; o campo HOUR refere-se à hora do acesso; o campo CONTEXT refere-se ao tipo de informação que esses dados revelam, i.e., financeiro, pessoal, e assim por diante. Agora, o pseudônimo e o campo *token* referem-se à pseudo-anonimização do usuário e o campo que vincula todos esses dados a um determinado proprietário do dado.

5.3.2 Serviços disponíveis no SaaS

O mecanismo ofertado com um serviço oferece a geração de pseudônimos, a geração de logs e a oferta de uma área administrativa com cenários de detecção a ser visualizada pelo proprietário do dado. Lembrando que na geração de pseudônimos está sendo implementada a nova abordagem da auditoria de pseudônimo.

Inicialmente, o serviço recebe os dados sensíveis acessados de uma aplicação através de um arquivo no formato XML (ver - Seção 5.3.1) a ser repassado para o LogCloud, que gera o log

para alimentar a base dos cenários de detecção. Na Figura 5.4, é apresentada uma visão conceitual dos serviços disponíveis no LogCloud e, dentro de cada serviço, as bases e os vínculos lógicos entre essas bases.

A Figura 5.4 apresenta os três serviços que são ofertados pelo LogCloud. Dois serviços são invocados dentro da aplicação que audita o acesso ao dado, tal como: o serviço de geração de pseudônimo e o serviço de geração de log. Já o terceiro serviço refere-se aos resultados da análise dos logs que geram os cenários de detecção previamente configurados no LogCloud. Seguem mais detalhes sobre os serviços:

(a) O **serviço de geração de pseudônimo** requisita dados do cidadão/funcionário para a geração de um pseudônimo. Em seguida, o serviço retorna um pseudônimo para a aplicação seguindo as regras da nova abordagem da auditoria de pseudônimo proposta neste trabalho (ver - Seção 3.5 do Capítulo 3);

(b) O **serviço de geração de logs** permite que a aplicação na nuvem repasse os dados sensíveis conforme um padrão pré-estabelecido, citado anteriormente, para o mecanismo gerar os logs.

(c) O **serviço de recuperação de logs** acontece a partir do acesso à área administrativa do LogCloud. Nessa área administrativa estão presentes os cenários de detecção de intrusão, com os quais o proprietário do dado pode interagir.

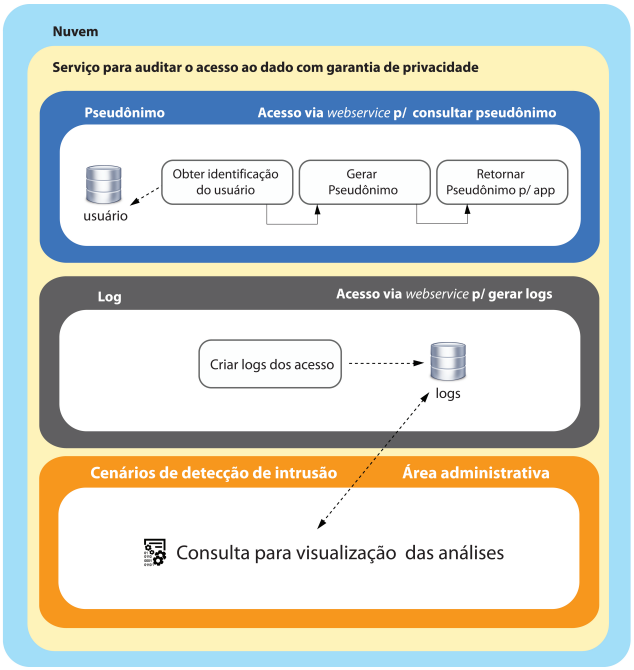

Figura 5.4: Serviços do LogCloud (nível conceitual).

5.4 Implementação

O serviço de auditoria e prestação de contas chamado LogCloud foi implementado considerando todos os requisitos levantados na seção 5.2 e as visões ali comentadas. As principais tecnologias exploradas para o desenvolvimento do protótipo, para a prova de conceito, foram a linguagem de programação Java®; Apache Maven® para gerenciamento das dependências de bibliotecas do projeto; banco de dados não relacional (MongoDB®), trazendo flexibilidade para a criação dos logs para as trilhas de auditoria; o *webservice* estilo arquitetural denominado RESTful (*Representational State Transfer*), baseado na especificação JAX-RS usando o projeto *Jersey*; framework Spring® v3.1, para gerenciamento do MVC; framework Spring Data MongoDB® v1.3, para acesso, geração de consultas e inserção no banco de dados MongoDB, e a biblioteca Quartz-jobs para escalonamento de serviços.

Para o *deploy* do serviço LogCloud, foi escolhido um servidor na nuvem do estilo PaaS da empresa RedHat chamado OpenShift®. O modelo PaaS oferecido por essa empresa garante uma hospedagem de aplicações com a mínima alteração de código.

Por se tratar de um serviço que precisa ser integrado com as aplicações, o projeto LogCloud foi dividido em duas partes principais. A primeira parte é formado pelo *core* principal do serviço a ser disponibilizado na nuvem. Juntamente com o serviço de pseudônimo e geração de logs está disponível ao proprietário do dado, via web, a área administrativa, para realizar consultas aos cenários de detecção de intrusão.

A segunda parte trata-se de um plugin a ser embutido em qualquer aplicativo Java® com o uso do Apache Maven®. Inicialmente, para fins de prototipação da proposta, o plugin foi testado em aplicações web Java® com o Apache Maven®. O plugin tem a funcionalidade de interceptar os dados mapeados definidos como sensíveis pelo desenvolvedor com base nos critérios do proprietário do dado seguindo o padrão apresentado na seção 5.3.1. Para isso, foram usados aspectos juntamente com *annotations* do Java®.

5.4.1 Core

O *core* do LogCloud é composto por serviços acessados pelo plugin embutido na aplicação como uma biblioteca. Os serviços requisitados pela aplicação são dois que tratam a geração de pseudônimos e a geração de logs. O serviço de recuperação é disponível pelo proprietário do dado através do acesso da página administrativa.

Para usar o serviço LogCloud, o proprietário do dado precisa criar uma conta de acesso

liberado na *Home* do *website* LogCloud. A criação da conta representa a assinatura do serviço e para criar a conta o proprietário precisa informar alguns dados pessoais incluindo email e senha, como apresentado na Figura 5.5.

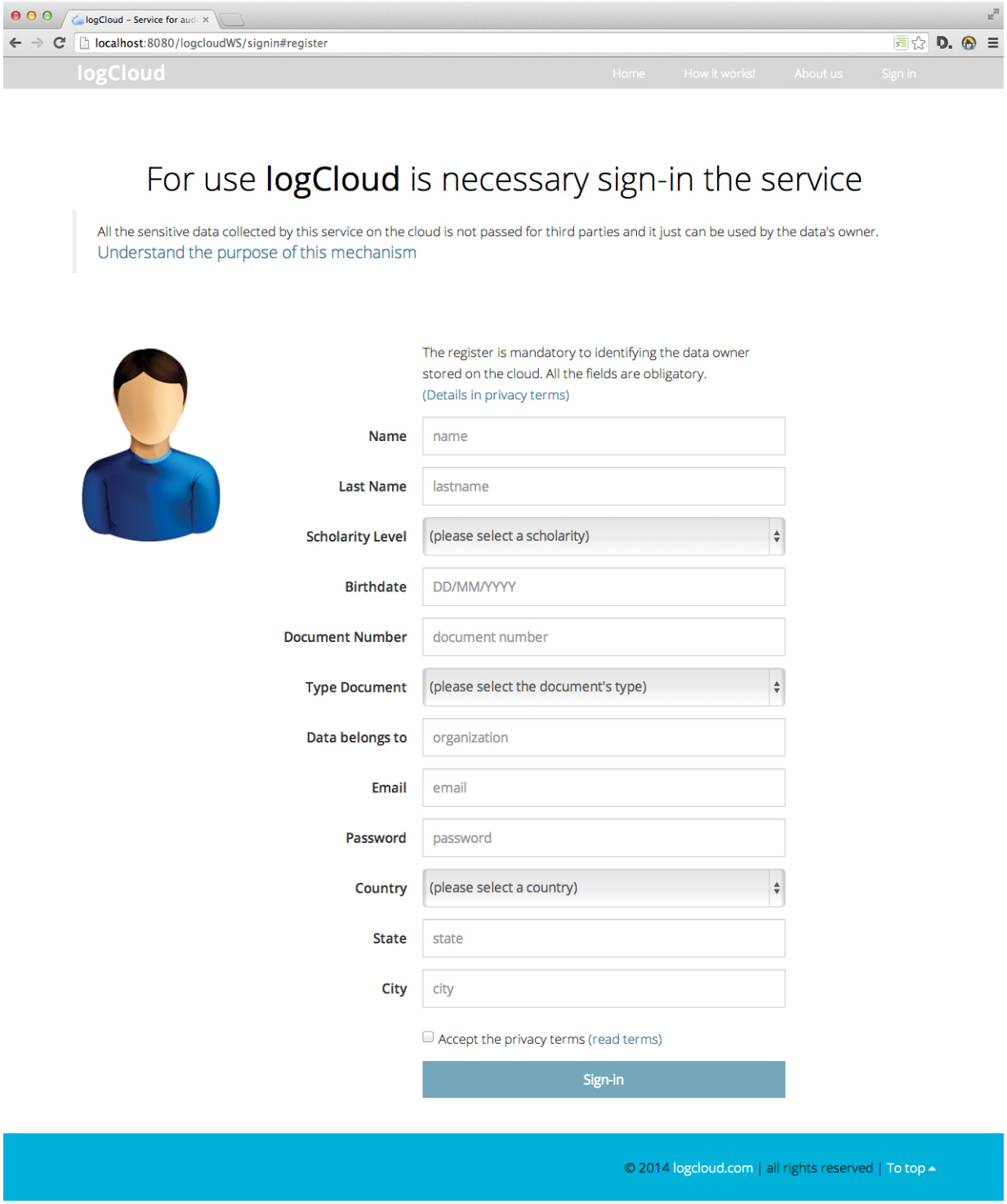

Figura 5.5: Criar conta de acesso aos serviços do LogCloud

O email e senha informados na criação da conta são usados para acessar a área administrativa, como apresentado na Figura 5.6. É na área administrativa, liberada pelo proprietário

do dado após a validação do usuário e senha, que está o valor do *token* para este proprietário e os cenários de detecção de intrusão, desenvolvidos para análise do proprietário do dado apresentado na Figura 5.7.

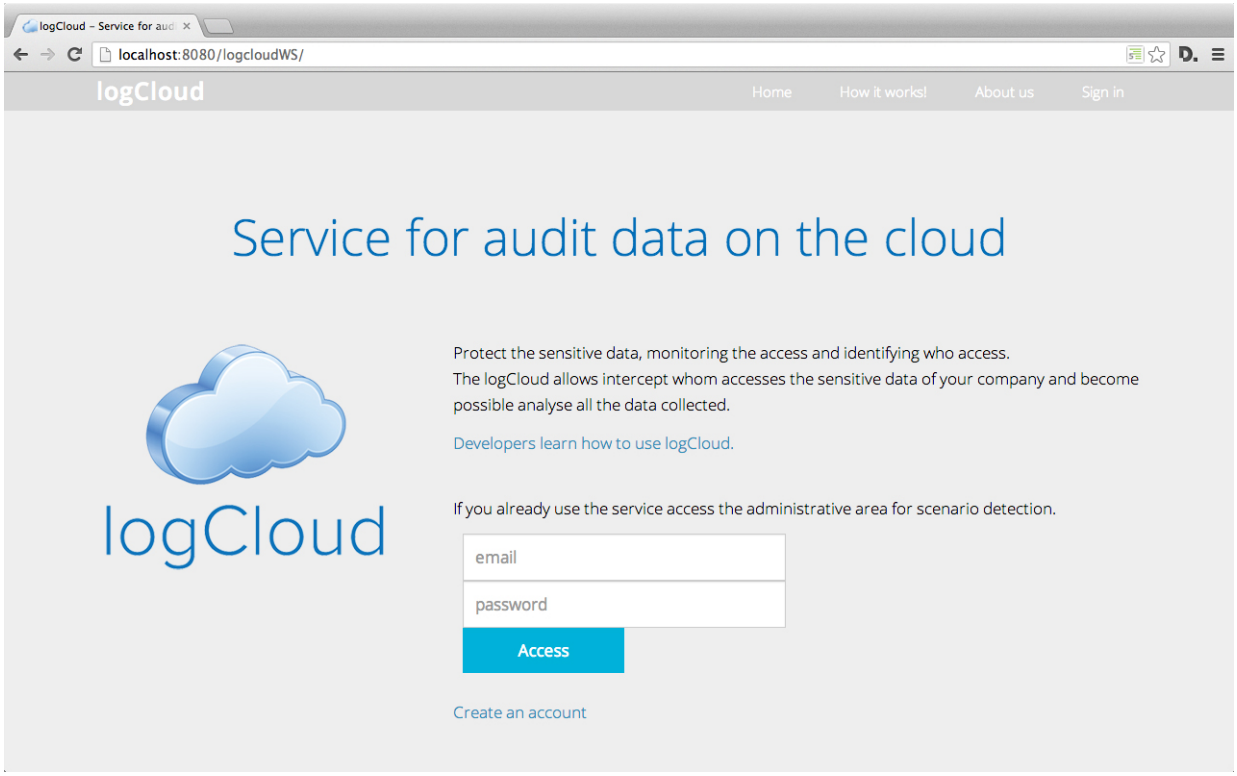

Figura 5.6: Acessando área administrativa

Com a conta de acesso criada, o proprietário do dado ganha uma área administrativa para manter-se atualizado com todos os acessos realizados no dado mapeado como sensível na aplicação, como apresentado na Figura 5.7. Então, a área administra contempla as seguintes visões: geral *(general view)*, acesso com base no contexto do dado *(context access)*, dados mais acessados *(top access)*, local de origem dos acessos *(access by area)* e uma relação de todos os dados acessados com quantidade de visita *(accessed data)*.

Os cenários mencionados apresentam um fundo dos acessos realizados disponíveis no serviço LogCloud. Com base na análise realizada nos logs de acesso, as regras das consultas podem ser equiparadas com visão do que se espera dos sistemas de detecção de intrusão onde são aplicados regras para identificar algum tipo de comportamento com base em uma noção prévia do que se espera.

A área administrativa somente está disponível para o proprietário do dado e somente o proprietário do dado tem o poder de identificar quem realizou o acesso ao dado com base no seu julgamento fundamentado pelos cenários ofertados seguindo os princípios da adaptação de auditoria de pseudônimo proposta neste projeto.

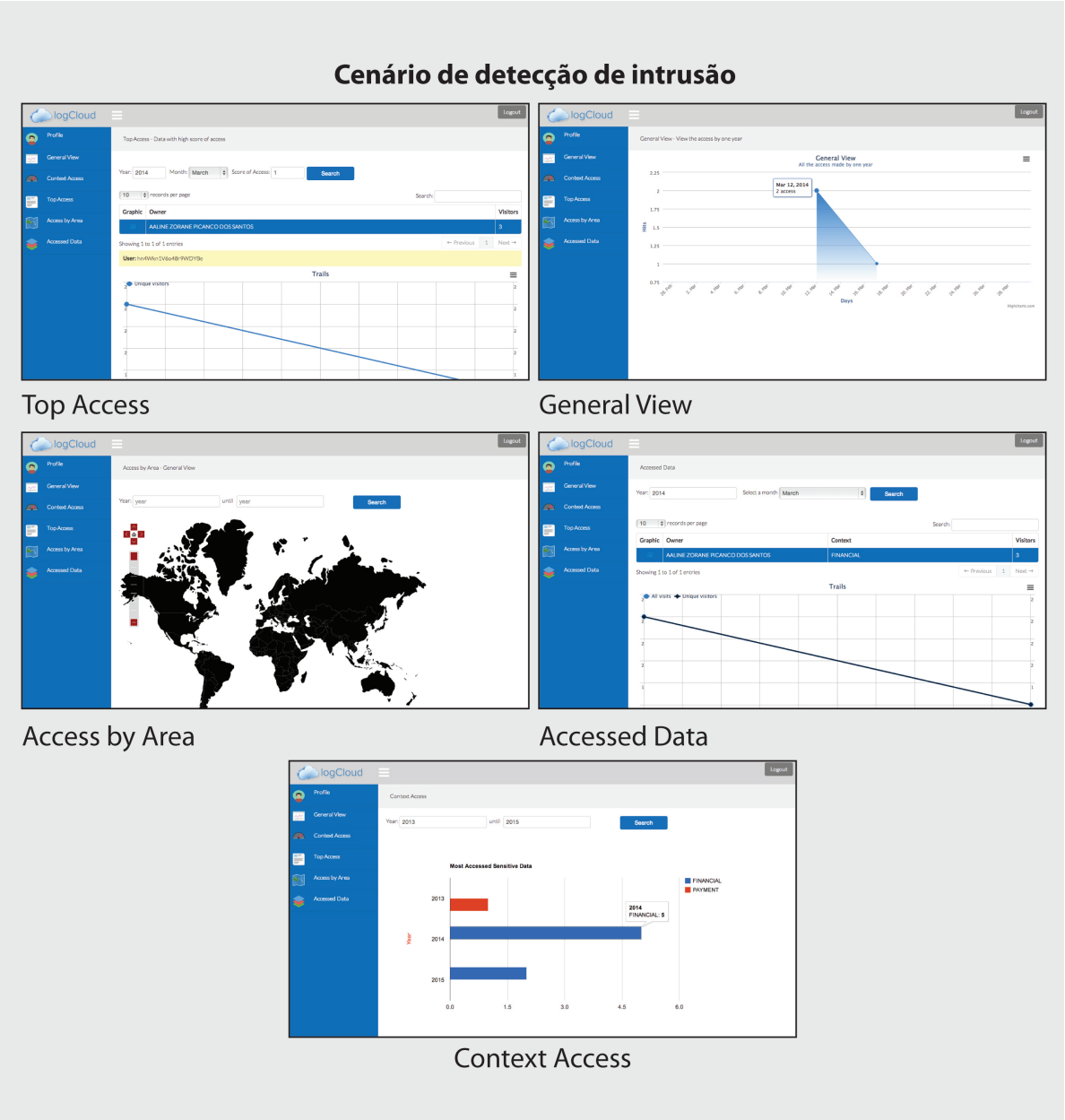

Figura 5.7: Cenários de detecção de intrusão

A conta do proprietário do dado é crucial para uso do serviço, porque com a criação da conta é gerado um *token* a ser incluído na aplicação quando invocar o serviço LogCloud. O *token* tem um papel fundamental, pois relaciona o log enviado à nuvem a um determinado proprietário do dado como apresentado na Figura 5.8.

O proprietário do dado somente tem acesso ao *token* depois que entrar na área administrativa. Após acessar a área administrativa informando o email e a senha é exibido ao proprietário do dado na página denominada perfil (*Profile*) um *token* gerado com base na criptografia SHA. O *token* é formado pela combinação da data da criação da conta, hora da criação da conta e o email do proprietário do dado. Tal *token* deve ser informado no momento do envio do log de

como vai ser abordado na seção 5.4.2.

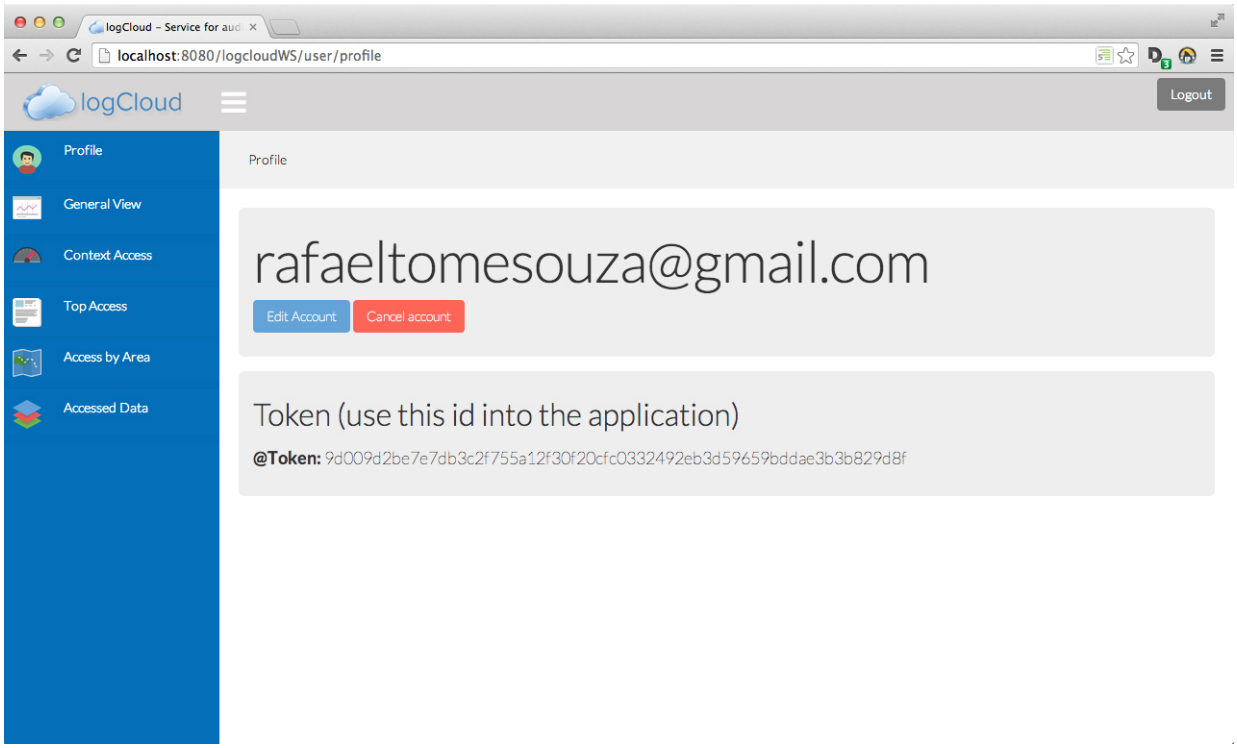

Figura 5.8: Área administrativa

A Figuras 5.5, 5.6, 5.7 e 5.8 resumem a área administrativa liberada ao proprietário do dado. Com base nos cenários de detecção de intrusão apresentada na Figura 5.7, o proprietário do dado pode fazer suas críticas com base nas consultas disponibilizadas nos cenários.

A análise do sistema de detecção de intrusão, no contexto deste serviço de auditoria e prestação de contas, é gerado de uma forma dinâmica considerando os logs que alimentam a base de dados não relacional. Os cenários representam consultas pré-configuradas a esta base onde o proprietário pode interagir com a passagem de alguns parâmetros tais como período de acesso e visualização de frequência.

A estrutura liberada ao proprietário do dado foi construída usando o framework Spring v3.1 e para a base de dados foi escolhido o banco de dados não relacional denominado MongoDB. A Figura 5.10 apresenta os modelos de classe envolvidos na implementação do projeto, em uma visão macro, para compreensão da forma como a parte administrativa liberada pela web foi implementada.

A Figura 5.10 apresenta um total de nove classes *"boundary" Java Server Pages* que usam os templates existente no pacote de mesmo nome. A requisição de acesso às classes são gerenciadas pelo *DispatcherServlet* que conduz a página a seu respectivo *controller*. O *controller* faz a intermediação com as classes DAO que estendem a classe GenericDAO e fazem o acesso ao

banco MongoDB através da biblioteca *framework Spring Data MongoDB* v1.3 .

O DAO acessa o modelo do projeto que são as classes que retornam as analises dos logs a serem apresentadas nos cenários de detecção de intrusão. Para isso, o DAO usa classes pertencentes aos pacotes com.logcloudWS.bigdata.accessByArea, com.logcloudWS.bigdata.contextAccess, com.logcloudWS.bigdata.generalView e com.logcloudWS.bigdata.accessedData. Já as classes Scholarity, UserLog, UserAdmin e Country oferecem respaldo para a criação do cadastro do proprietário do dado e do usuário do log.

Os cenários de detecção de intrusão são compreendidos em cinco análises realizadas na base do LogCloud. Todas as análises são realizadas dentro da base de log que é alimentada durante o envio dos dados pelo proprietário do dado. Cada um dos cenários contempla regras diferenciadas permitindo ao proprietário do dado uma visão diferenciada sobre os acessos aos dados mapeados como sensíveis.

O cenário *General View*, Figura 5.9, apresenta uma visão geral de todos os acessos aos dados sensíveis do proprietário do dado. Para isso é usado nas buscas o *token* obtido automaticamente quando o proprietário acessa a área administrativa.

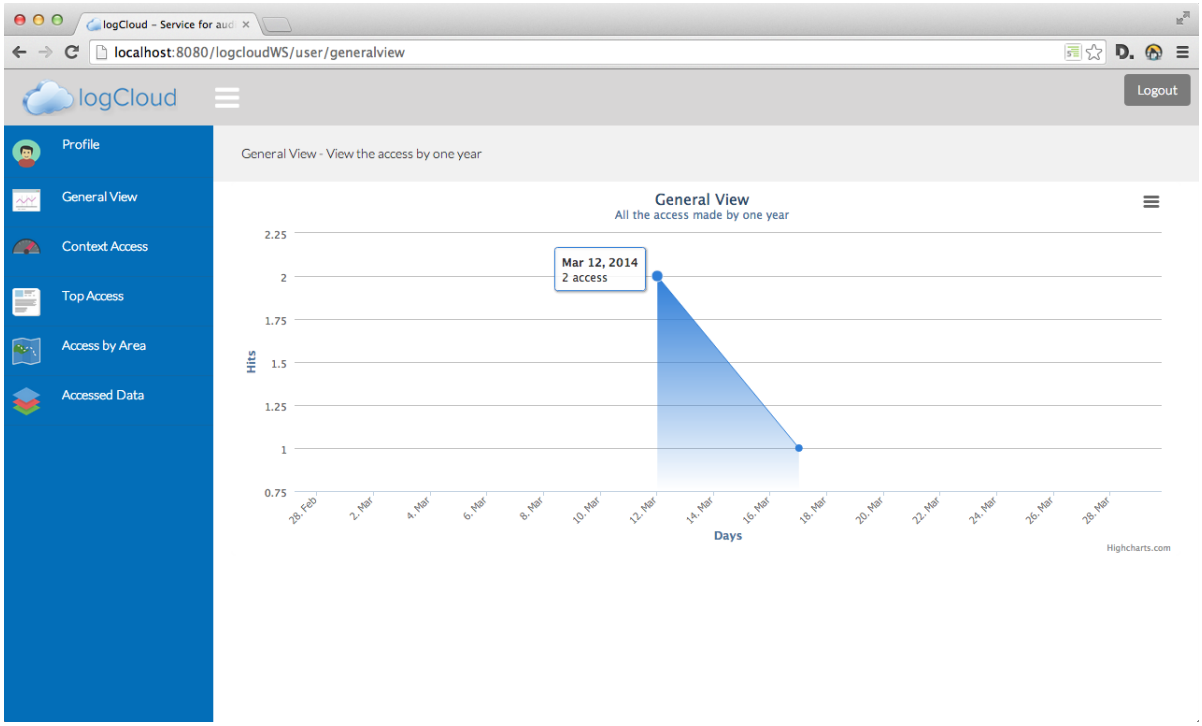

Figura 5.9: General View

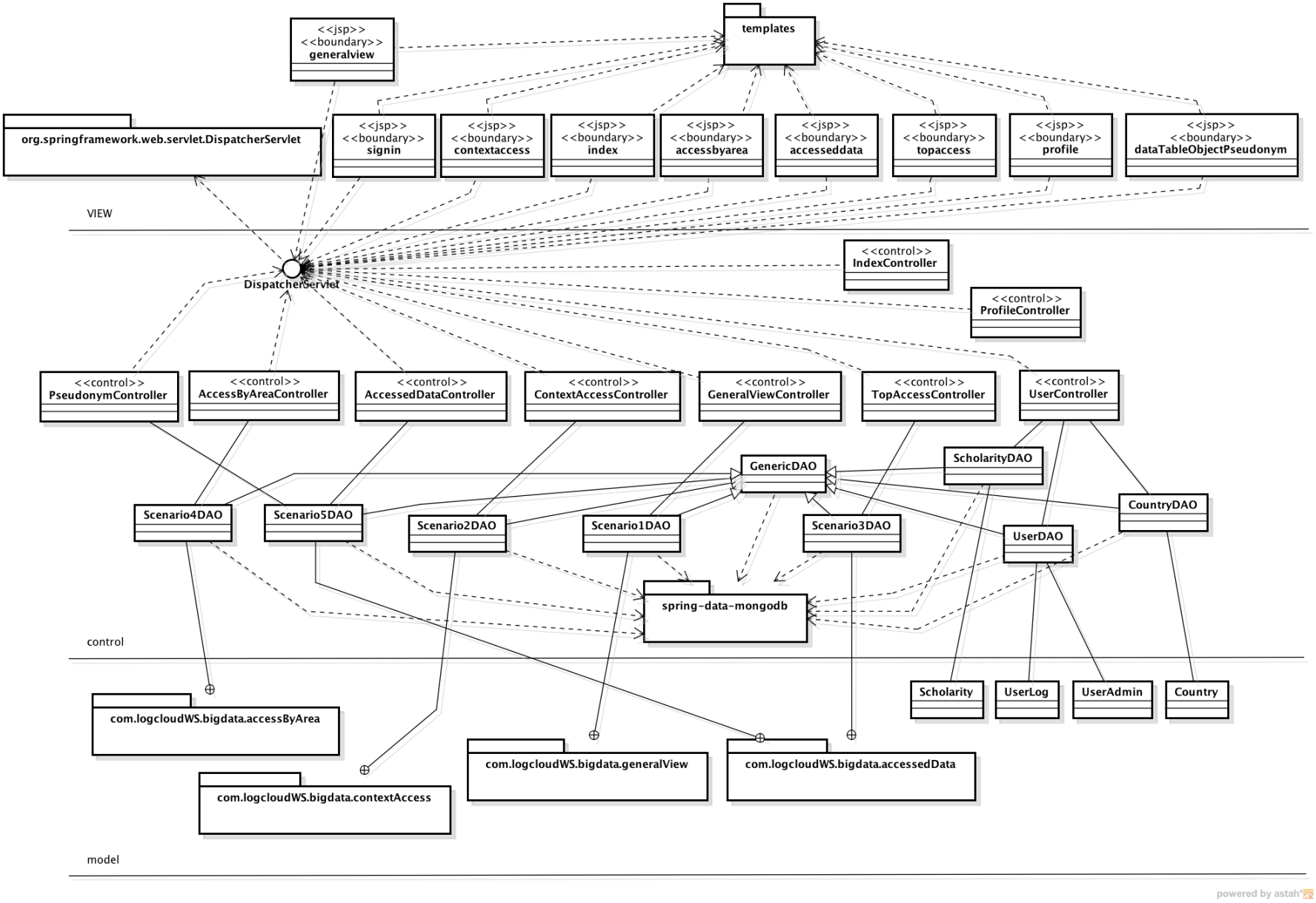

Figura 5.10: Modelo de classe

Este cenário retorna uma quantidade de acessos por dia. Então, o período de busca é o ano atual. E o período pesquisado é o primeiro dia do ano até o último dia do ano. Com esse cenário o proprietário do dado pode ter uma visão da quantidade de acessos que os dados sensíveis andam sofrendo.

O cenário *Context Access*, Figura 5.11, apresenta uma visão geral dos acessos realizados nos dados sensíveis considerando o tipo de contexto informado e o proprietário do dado. No mapeamento do dado sensível, o desenvolvedor é obrigado a especificar um tipo de contexto do conjunto de dados a ser auditado.

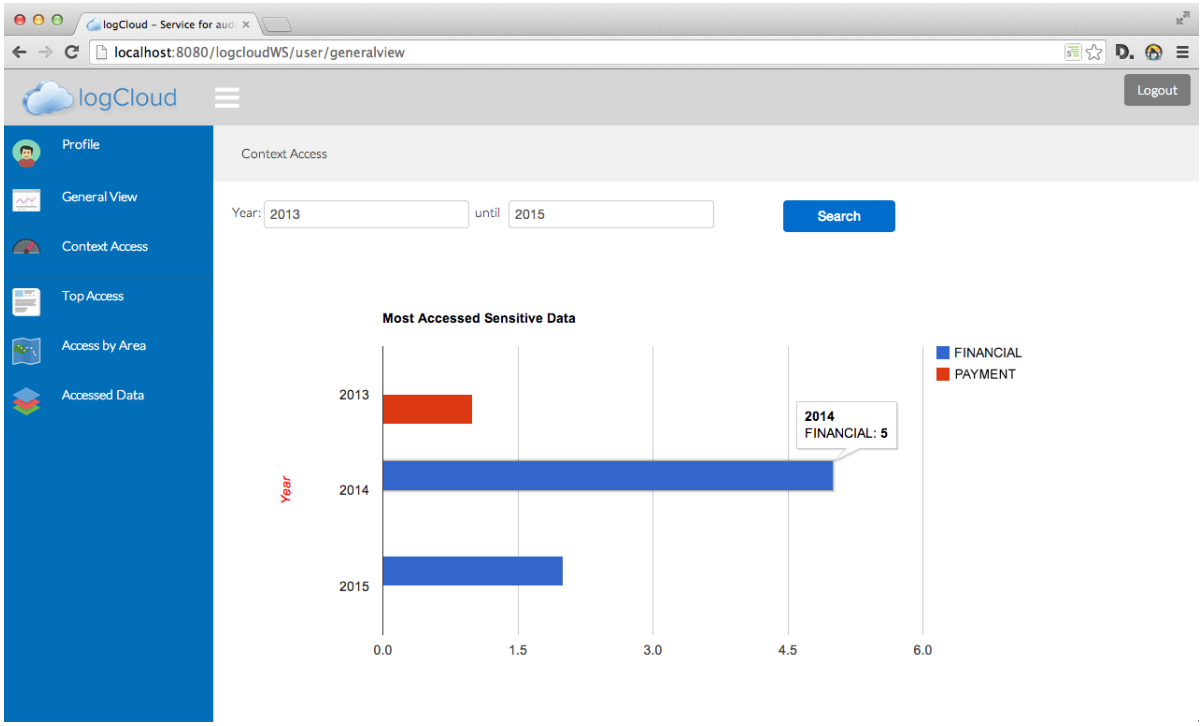

Figura 5.11: Context Access

Com base no tipo de contexto este cenário permite fazer uma inferência na pesquisa do dado com o período em anos. O resultado é a apresentação em uma divisão de anos dos tipos de contextos mais acessados na aplicação.

O cenário *Top Access* , Figura 5.12, apresenta uma visão sobre os dados sensíveis mais acessados com base em um *score*. O *score* seria quantidade de acessos, e o proprietário de dados poderia definir uma quantidade de acessos limite para trazer os resultados que se enquadram na busca. A finalidade deste cenário é trazer qual o dado que um determinado usuário anda mais acessando.

O cenário *Access by Area*, Figura 5.13, apresenta uma visão sobre os dados com base na origem dos acessos. A origem dos acessos é identificada com base na análise da origem do IP

capturado no momento do acesso. Para isso, é usado uma base de dados liberada gratuitamente pelo site da GeoLite que pode ser feito download em: http://dev.maxmind.com.

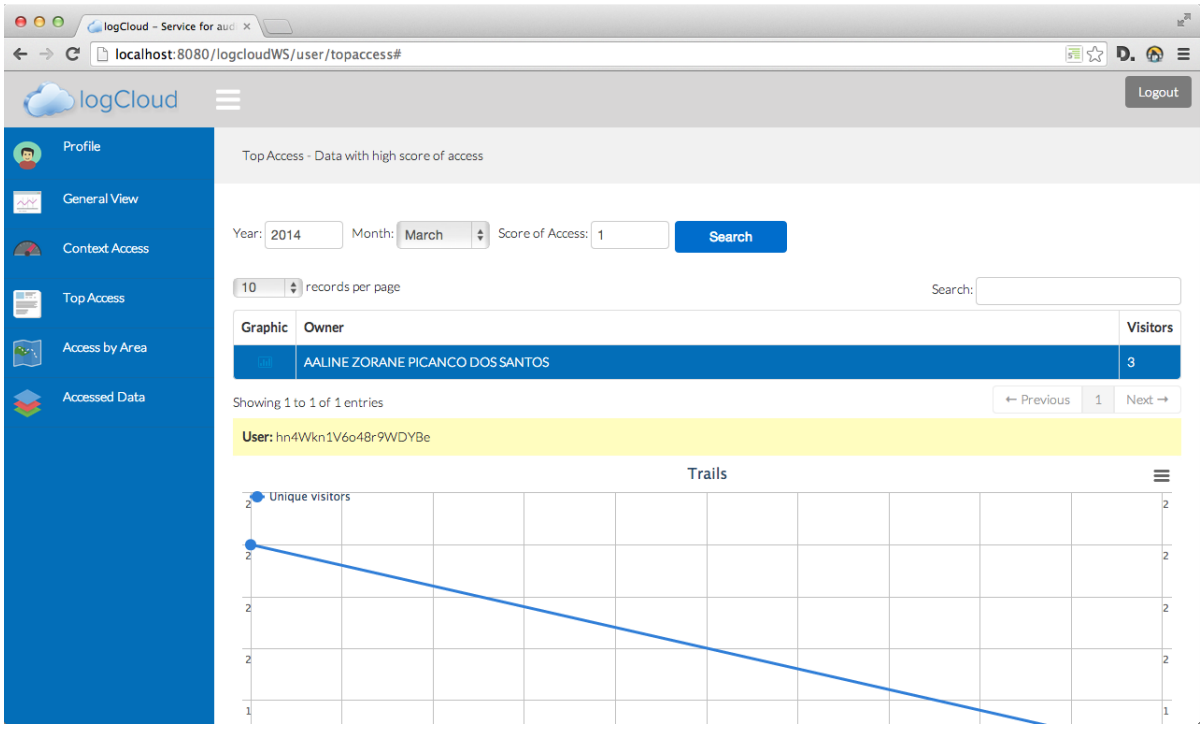

Figura 5.12: Top Access

O banco de dados da GeoLite oferece uma API para identificação de região, cidade e país com base no IP informado. Os IPs que conseguem ser identificados são lançados na página com quantidade de acesso e região.

O cenário *Accessed Data*, Figura 5.14, apresentada uma lista de todos os dados acessados e consequentemente a quantidade de acesso que os dados sofreram durante o período que estão disponíveis. Nesta página o proprietário do dado consegue identificar o usuário real, quebrando pseudônimo do usuário. Além disso, o usuário interage com o cenário informando um ano e um período do mês. Na sequência, o sistema retorna uma lista dos acessos realizados e com a frequência das visitas. Ao clicar nos dados é apresentado os usuários que acessaram juntamente com a proteção da identidade.

A parte administrativa é a área na qual o proprietário do dado tem acesso com base no email e senha usados no momento do cadastro da conta. Porém, o proprietário do dado interage com o serviço LogCloud com o uso do plugin desenvolvido que faz a chamada dos serviços disponíveis juntamente com a parte da área administrativa.

Além da visão de acesso aos cenários de detecção de intrusão, o plugin desenvolvido tem acesso a dois serviços chamados de serviço de geração de pseudônimos e serviço de geração de

logs. Tais serviços completam o *core* principal do LogCloud.

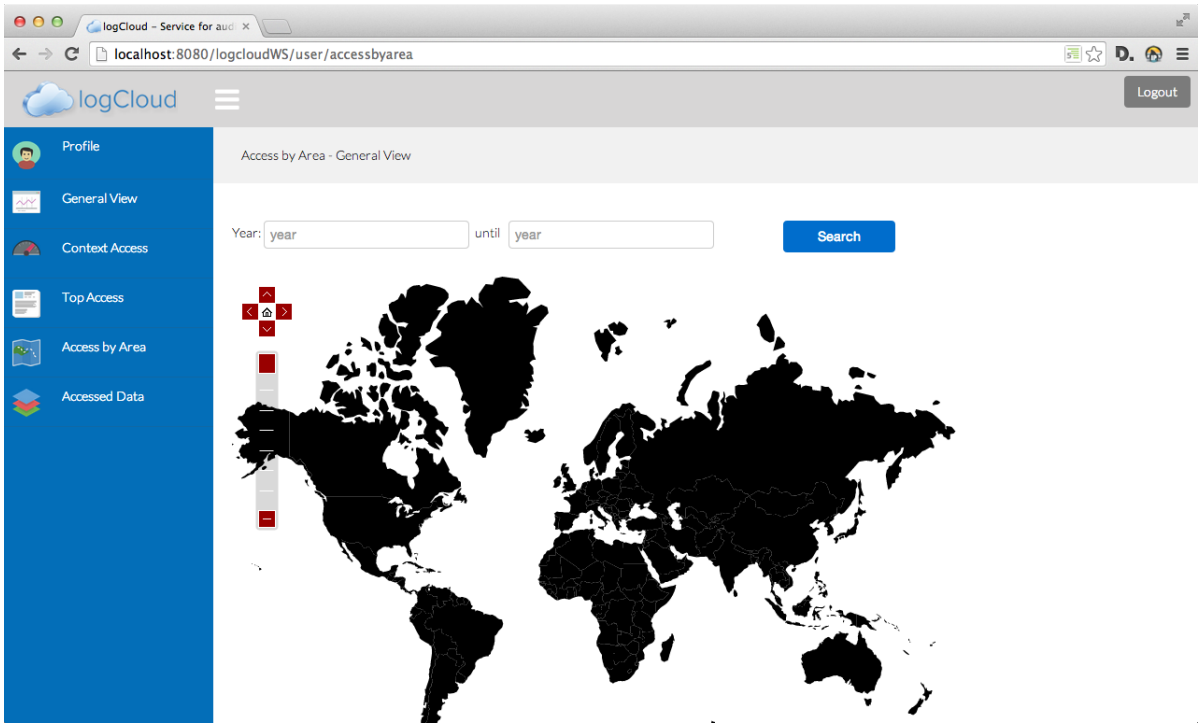

Figura 5.13: Access by Area

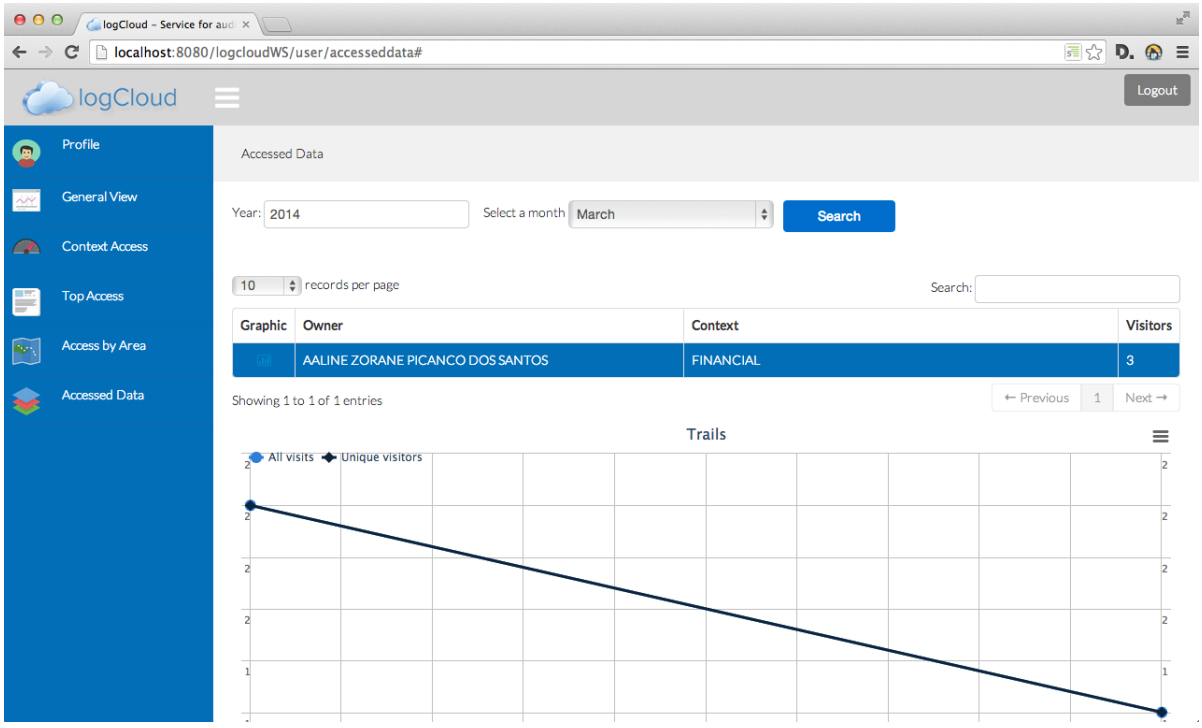

Figura 5.14: Accessed Data

5.4.1.1 Serviço de geração de pseudônimo

O serviço de geração de pseudônimo é onde está implementado a nova abordagem de auditoria de pseudônimo apresentada na seção 3.5 do capítulo 3. A implementação da auditoria de pseudônimo no serviço LogCloud permitiu a proteção da identidade do usuário no momento que o log é gravado na base de logs do serviço. Assim, a identidade só é revelada quando o proprietário do dado julgar necessário.

A **geração de pseudônimo** garante que a cada acesso realizado pelo usuário, o mesmo tenha um pseudônimo novo garantindo a proteção da identidade. Além disso, o pseudônimo do usuário está protegido com a chave simétrica do proprietário do dado e a chave pública do usuário.

Um ponto que tem que ser complementado é que no momento que um proprietário do dado adere ao uso do LogCloud, automaticamente o serviço gera uma chave simétrica ao proprietário usando o algoritmo AES.

Ou seja, quando solicitado a geração do pseudônimo o email que é o identificador adotado no serviço LogCloud para vincular os registros de log com o usuário real é codificado com a chave simétrica do proprietário do dado e chave pública do usuário.

A ação de criptografar duas vezes o email do usuário com as chaves garante que somente a descoberta do usuário seja realizada pelo proprietário do dado. Mesmo se alguém acessar diretamente a base de dados a descoberta da identidade real fica inviável porque necessita identificar os algoritmos de criptografia e descobrir qual a chave simétrica do proprietário do dado.

Como resposta da solicitação a um serviço o sistema envia ao aplicativo um novo pseudônimo, o *modulus* da chave privada e o *exponent* da chave privada baseada no algoritmo de criptografia RSA. A necessidade de registrar o *modulus* e o *exponent* no log aconteceu porque os sistemas de detecção de intrusão criado precisa ter noção de que os logs referem-se a uma pessoa. No entanto, o sistema de detecção de intrusão não precisa saber qual pessoa seria.

A decisão de criptografar o pseudônimo com a chave privada e manter a chave privada no log facilitou a descoberta da identidade real do usuário fragilizando o esquema de auditoria proposto. Então, optou-se por criar uma chave de criptografia simétrica usando o algoritmo AES toda vez que for criado uma conta de proprietário do dado. E essa chave torna-se mestra para proteger a identidade real do usuário, acrescido da chave privada do usuário.

Assim, a proteção da identidade do usuário é garantida por duas pessoas: uma pessoa é o proprietário do dado que encripta o email com a chave simétrica e a outra pessoa é o usuário

que acessa que criptografa com a chave pública. Então, para descobrir a identidade real a chave privada do usuário torna-se necessário e na sequência a chave simétrica do proprietário do dado. A identidade real só pode ser descoberta pelo proprietário do dado, juntamente com os cenários de detecção de intrusão.

Uma outra questão relevante é que alguém pode ter acesso a chave privada do usuário que está no log e com isso fazer uma busca direta na base de dados ferindo todo o esquema proposto. Então, toda a chave do usuário pública e privada é criptografada com a chave simétrica do proprietário do dado, evitando este tipo de problema.

O serviço de geração de pseudônimo é oferecido através de um *webservice* no estilo arquitetural RESTFul. Para isto, o serviço precisa receber um XML contendo os campos apresentados no Código 5.2.

Código 5.2: XML para geração do pseudônimo do usuário

```
1  <userLog>
2          <name>...</name>
3          <lastName>...</lastName>
4          <scholarity>....</scholarity>
5          <birthDate>....</birthDate>
6          <numberDocument>....</numberDocument>
7          <typeDocument>....</typeDocument>
8          <email>....</email>
9          <country>....</country>
10         <state>....</state>
11         <city>....</city>
12 </userLog>
```

Os campos informados no Código 5.2 são repassado ao serviço de geração de pseudônimo quando é invocado pelo plugin que gerência o consumo do serviço. Como resultado da interação, o serviço retorna um XML contendo o pseudônimo como apresentada no Código 5.3.

Código 5.3: XML de retorno do serviço geração de pseudônimos

```
1  <userLog>
2          <pseudonym>...</pseudonym>
3          <modulus>...</modulus>
4          <privateExponent>....</privateExponent>
5  </userLog>
```

A parte principal da implementação do serviço pode ser analisada interpretando o Código 5.4. No Código 5.4 é apresentado o momento em que os parâmetros XML são passados para o método, com base no email e token é identificado se o usuário está inserido na base de usuários do LogCloud. Caso o usuário não esteja é feito o cadastro do usuário a ser usado no momento de identificar a identidade real do usuário do acesso.

No ato de registrar o usuário é gerado chaves pública e privada com base na criptografia RSA e na sequência as chaves são criptografadas com a chave simétrica do proprietário do dado. Após esta operação é realizado a geração do pseudônimo e para isso é obtido a chave simétrica do proprietário do dado a qual criptografa o email e na sequência é criptografado com a chave pública do usuário. Por fim, o serviço retorna um objeto do tipo Pseudonym que contém os campos modulus, privateExponent e pseudonym.

Quando o usuário que solicita o acesso pela segunda vez, o mesmo usuário já está cadastro sendo assim só é gerado um novo pseudônimo. O cadastro não é realizado novamente, simplesmente é recuperado o registro do banco de dados e na sequencia é executado o procedimento de geração de pseudônimo.

Código 5.4: Serviço de geração de pseudônimo

```
1  @Path("/service/pseudonym")
2  public class Alias {
3  ...
4  @PUT
5  @Path("generate")
6  @Produces(MediaType.APPLICATION_XML)
7  public Pseudonym trail(JAXBElement<UserLog> jaxbUser) {
8
9  Pseudonym pseudonym = new Pseudonym();
10 UserLog user = jaxbUser.getValue();
11
12 try {
13
14    CryptografyRSA rsa = new CryptografyRSA();
15    Cipher cipherRSA = rsa.createCipher();
16    UserLog userLog = userDAO.getUser(user.getEmail(),user.getToken());
17    CryptografyAES aes = new CryptografyAES();
18    Cipher cipherAES = aes.createCipher();
19    UserAdmin userAdmin = userDAO.getUserAdmin(user.getToken());
20    SecretKeySpec secretKeyAdmin = new SecretKeySpec(userAdmin.getSymmetricKey(),"↩
          AES");
21
22    if (userLog == null) {
23          //KeyFactory
24          KeyPair generate = rsa.createKeys();
25          KeyFactory fact = KeyFactory.getInstance("RSA");
26
27          //Obtain the PUBLIC KEY
28          RSAPublicKeySpec pub = fact.getKeySpec(generate.getPublic(), ↩
                RSAPublicKeySpec.class);
29          String modulusPub = aes.encrypt(pub.getModulus().toString(), cipherAES, ↩
                secretKeyAdmin);
30          String exponentPub = aes.encrypt(pub.getPublicExponent().toString(), ↩
                cipherAES, secretKeyAdmin);
31          KeyRSA publicKey = new KeyRSA();
```

```
32          publicKey.setModulus(modulusPub);
33          publicKey.setExponent(exponentPub);
34
35          //Obtain the PRIVATE KEY
36          RSAPrivateKeySpec priv = fact.getKeySpec(generate.getPrivate(), ↩
                RSAPrivateKeySpec.class);
37          String modulusPriv = aes.encrypt(priv.getModulus().toString(), cipherAES, ↩
                secretKeyAdmin);
38          String exponentPriv = aes.encrypt(priv.getPrivateExponent().toString(), ↩
                cipherAES, secretKeyAdmin);
39          KeyRSA privateKey = new KeyRSA();
40          privateKey.setModulus(modulusPriv);
41          privateKey.setExponent(exponentPriv);
42
43          user.setPrivateKey(privateKey);
44          user.setPublicKey(publicKey);
45          userDAO.save(user);
46          userLog = user;
47      }
48
49      // Obtain the PUBLIC KEY
50      BigInteger modulusPub = new BigInteger(aes.decrypt(userLog.getPublicKey().↩
            getModulus(), cipherAES, secretKeyAdmin));
51      BigInteger exponentPub = new BigInteger(aes.decrypt(userLog.getPublicKey().↩
            getExponent(), cipherAES, secretKeyAdmin));
52
53      String aliasAdminSecretKey = aes.encrypt(userLog.getEmail(), cipherAES, ↩
            secretKeyAdmin);
54
55      RSAPublicKeySpec pubKeyUser = new RSAPublicKeySpec(modulusPub, exponentPub);
56      KeyFactory factoryPubKeyUser = KeyFactory.getInstance("RSA");
57      PublicKey userPubKey = factoryPubKeyUser.generatePublic(pubKeyUser);
58      String aliasUserPublicKey = rsa.encrypt(aliasAdminSecretKey, cipherRSA, ↩
            userPubKey);
59
60      pseudonym.setPseudonym(aliasUserPublicKey);
61
62      //Obtain the PRIVATE KEY
63      BigInteger modulusPriv = new BigInteger(aes.decrypt(userLog.getPrivateKey().↩
            getModulus(), cipherAES, secretKeyAdmin));
64      BigInteger exponentPriv = new BigInteger(aes.decrypt(userLog.getPrivateKey().↩
            getExponent(), cipherAES, secretKeyAdmin));
65
66      pseudonym.setModulus(modulusPriv);
67      pseudonym.setPrivateExponent(exponentPriv);
68
69      } catch(Exception e) {
70           e.printStackTrace();
71      }
72      return pseudonym;
73  }
74  ...
75  }
```

O serviço de geração de pseudônimo protege a identidade do usuário que é revelada apenas no interesse do proprietário do dado ao analisar os cenários de detecção de intrusão, realizando assim a auditoria de pseudônimo.

5.4.1.2 Serviço de geração de logs

O serviço de geração de logs disponível no LogCloud permite a geração das trilhas de auditoria. Para isso, o serviço precisa receber um XML contendo os campos apresentados no Código 5.5.

Código 5.5: XML para geração dos logs

```
<log>
        <pseudonym>...</pseudonym>
        <ip>...</ip>
        <device>...</device>
        <owner>...</owner>
        <date>...</date>
        <hour>...</hour>
        <context>...</context>
        <modulus>...</modulus>
        <privateExponent>....</privateExponent>
        <token>...</token>
        <logFields>
                <data>
                        <field>...</field>
                        <value>...</value>
                </data>
                ...
        </logFields>
</log>
```

O retorno do serviço de geração de pseudônimo deve ser repassado ao serviço de geração de logs. Tal serviço, recebe como passagem de parâmetro os campos presente no Código 5.3, acrescidos de outros campos como apresentado no Código 5.5.

O serviço de geração de logs ao receber o XML, executa procedimentos para identificação da região do IP acrescentando informações ao registro do log tais como *isoCodeCountry*, *country*, *state*, *isoCodeState*, *city*, *latitude* e *longitude*. Além deste campo, o serviço gera um hash do registro de log para identificar quando outros dados iguais ao que foi acessado, usando criptografia SHA.

O registro gerado pelo serviço na base do LogCloud alimenta a base do sistema de detecção de intrusão que é refletida na consulta dos cenários detecção liberados aos proprietários do dado.

A parte principal da implementação do serviço pode ser analisada interpretando o Código 5.6. No Código 5.6 é apresentado o momento em que os parâmetros XML é recebido pelo serviço. A primeira validação é feita em cima do *token* informado. O *token* deve ser válido para que o serviço aceite e gere o log com os parâmetros XML informado.

Após a validação, o serviço gera um hash para ser um identificador único para os dados passados como parâmetro, na sequencia é feito uma analise com base no IP identificado e por último o log é salvo na base de dados.

Código 5.6: Serviço de geração de logs

```
1  @Context
2  UriInfo uriInfo;
3
4  private LogDAO logDAO = new LogDAO();
5  private UserDAO userDAO = new UserDAO();
6  ...
7
8  @Path("/service/trail")
9  public class Trail {
10
11    @PUT
12    @Consumes(MediaType.APPLICATION_XML)
13    @Path("create")
14    public Response trail(JAXBElement<Log> jaxblog) {
15       Log l = jaxblog.getValue();
16          if (userDAO.getToken(l.getToken())) {
17             return putAndGetResponse(l);
18          } else {
19             return Response.status(Status.BAD_REQUEST).build();
20    }
21 }
22
23 public Response putAndGetResponse(Log l) {
24    Response res;
25    String uniqueFields = null;
26
27    Log log = new Log();
28    log.setPseudonym(l.getPseudonym());
29    log.setIp(l.getIp());
30    log.setDevice(l.getDevice());
31    log.setOwner(l.getOwner());
32    log.setContext(l.getContext());
33    log.setModulus(l.getModulus());
34    log.setPrivateExponent(l.getPrivateExponent());
35    log.setToken(l.getToken());
36
37    List<LogField> logFieldList = new ArrayList<LogField>();
38        for (Iterator<LogField> iterator = l.getLogFields().iterator(); iterator.↩
              hasNext();) {
```

```
39              LogField fieldLog = (LogField) iterator.next();
40              logFieldList.add(fieldLog);
41              uniqueFields +=fieldLog.getField()+fieldLog.getValue();
42          }
43      log.setLogFields(logFieldList);
44
45      HashMessageSHA hm = new HashMessageSHA();
46      String m = hm.hashMessage(log.getOwner()+log.getContext()+uniqueFields);
47      log.setHash(m);
48
49      File database = new File(System.getenv("OPENSHIFT_REPO_DIR")+"/src/main/↩
            resources/GeoLite2-City.mmdb");
50      try {
51        DatabaseReader reader = new DatabaseReader.Builder(database).build();
52        CityResponse response1 = reader.city(InetAddress.getByName(log.getIp()));
53        log.setIsoCodeCountry(response1.getCountry().getIsoCode());
54        log.setCountry(response1.getCountry().getName());
55        log.setIsoCodeState(response1.getMostSpecificSubdivision().getIsoCode());
56        log.setState(response1.getMostSpecificSubdivision().getName());
57        log.setCity(response1.getCity().getName());
58        log.setLatitude(response1.getLocation().getLatitude().toString());
59        log.setLongitude(response1.getLocation().getLongitude().toString());
60      } catch (IOException e) {
61        e.printStackTrace();
62      } catch (GeoIp2Exception e) {
63        e.printStackTrace();
64      }
65
66      logDAO.save(log);
67      res = Response.created(uriInfo.getAbsolutePath()).build();
68      return res;
69  }
70  ...
71  }
```

A composição dos serviços de geração de pseudônimo e geração logs mais a parte de consulta aos cenários de detecção de intrusão disponível para o proprietário do dado forma o que foi denominado de serviço de auditoria e prestação de contas, chamado de LogCloud.

O consumo dos serviços disponíveis do LogCloud não é feito com chamadas direta pelo desenvolvedor, para isso é necessário o desenvolvedor usar um plugin. O plugin tem a finalidade de facilitar o uso dos serviços e geração do XML para se comunicar com os serviços com base no que foi discutido na seção 5.3.1 do Capítulo 5.

5.4.2 Plugin

A outra parte do LogCloud, o plugin LogCloud, tem o poder de interceptar os dados marcados com anotações especiais. O plugin a ser embutido intercepta o acesso aos dados sensíveis

e envia ao serviço. Assim, é obrigatório que o proprietário do dado marque o dado de acordo com o padrão estabelecido.

Em outras palavras, para cada atributo marcado como proprietário, precisa ser definido os dados pertencente aquele proprietário do dado. Então, as pesquisas do banco de dados precisam obedecer ao padrão - um proprietário pode ter muitos dados que precisam ser monitorados (Veja a seção 5.3.1 do Capítulo 5).

Para tornar possível capturar os dados acessados foram disponibilizados no plugin os *annotations*. Existem *annotations* que definem quem é o proprietário do dado e o dado a ser mapeado.

O *annotation* @Owner identifica a pessoa da empresa a quem pertencem os dados. Pois, os dados pertencem a uma empresa, mas a empresa tem dados que pertence a pessoa real, como o valor do salário. Então, o *annotation* deve marcar a pessoa que o dado é vinculado. Já o *annotation* @Value define o valor para as trilhas de auditoria, de modo que seja possível usá-lo mais de uma vez e, assim, mapear os dados sensíveis.

Outro ponto é a necessidade de informar ao plugin o usuário que está acessando os dados sensíveis para criar o pseudônimo. Para isso, é necessário utilizar o *annotation* @User, e alimentá-lo com os dados de login do aplicativo. No entanto, para a utilização do @User um tipo de objeto próprio do plugin LogCloud precisa ser usado, o objeto LogUser. Então, é necessário alimentar um objeto do tipo LogUser e marcá-lo com o *annotation* @User.

Para finalizar os *annotations* é necessário informar alguma referência ao proprietário dos dados. Portanto, passa a ser necessário alimentar um atributo com o *token* criado e marcá-lo usando o @Token. Para obter informações sobre o ip e o dispositivo, é necessário criar e alimentar os atributos marcados com o *annotation* @Ip e o *annotation* @Device. Uma observação é que este atributo deve ser alimentado pelo desenvolvedor quando for declarado. Assim, a LogCloud cria um maneira de enviar este dado junto com as trilhas de auditoria.

Para chamar corretamente o plugin LogCloud, recomenda-se usar o padrão DAO em seu aplicativo Java. Desta forma, o *bean* que retorna o proprietário e os dados que estão sendo monitorados usando o *annotation* @Owner e @Value. Assim, todas os outros *annotations* tais como @Token, @User, @Ip e @Device podem ser concentrados no DAO.

Depois de realizar todas as marcações necessárias, o desenvolvedor precisa marcar o método que retorna a coleção a ter o seu acesso auditado usando o annotation @Audited. Neste *annotation*, deve ser definido o contexto que o dado está relacionado como por exemplo, o contexto de um dado relacionado ao cenário financeiro, i.e., deve ser definido @Audited(context=

Type.FINANCIAL).

Em geral, os *annotations* precisam ser anexado ao bean usado para retornar os dados a serem visualizados pelo usuário. Por exemplo, se há um método em uma classe Java que retorna uma lista do tipo *"PublicFederalServerRemuneration"*, assim a classe *"PublicFederalServerRemuneration"* precisa receber as anotações especiais indicando que os dados sensíveis precisam ser auditados quando visualizados.

Além disso, o método de *"getServerDetailRemuneration"* que retorna a lista para a camada de visão precisa usar uma anotação especial chamado @Audited na classe implementada *"PublicFederalServerDAO"*. Os *annotations* são interceptados usando *annotation* Java, reflexão e programação orientada a aspectos (AOP), como é mostrado na Figura 5.15. O uso de *annotations* trouxe flexibilidade para determinar o que é sensível, manter o código com a menor alteração possível e mais intuitivo ao desenvolvedor.

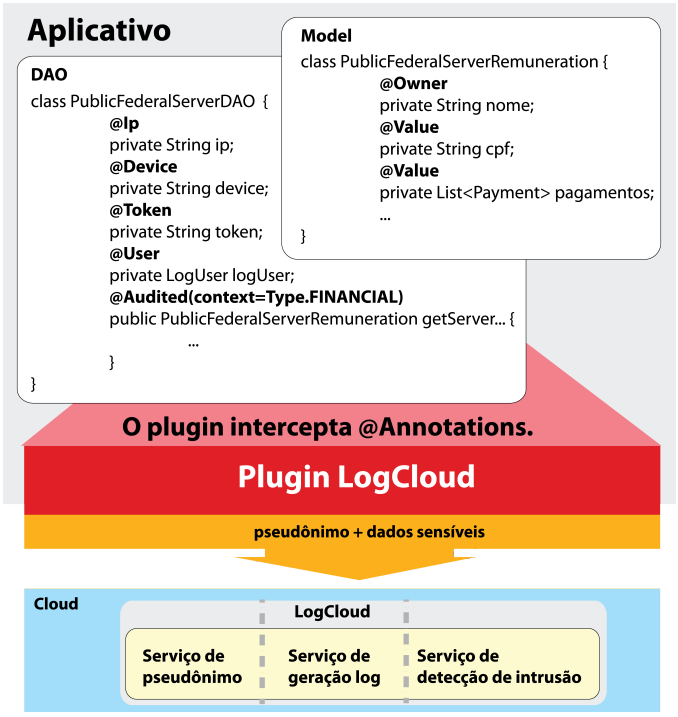

Figura 5.15: O uso de *annotations* específicos invoca a chamada do serviço LogCloud

O projeto do plugin do LogCloud foi concebido considerando o Spring AOP com a integração dos annotations do AspectJ. A parte reponsável por interceptar os annotations pode ser vista no Código 5.7.

Na primeira parte do Código 5.7 é apresentado um trecho em que intercepta o retorno de métodos públicos marcados com o annotation @Audited. Para isso, foi usado o annotation @AfterReturning executado depois que o método retornou um resultado e pode ser usado para interceptar o resultado retornado também.

Na sequência, entre as linhas 19 à 31 é obtido o contexto que os dados estão relacionados. O plugin oferece alguns contextos padrões tais como: FINANCIAL, PAYMENT, PERSONALUSERDATA e LOGGINGDATA.

Código 5.7: Intercepção de annotations para invocar os serviços do LogCloud

```
1  @Aspect
2  public class Monitoring {
3          private LogUser user = new LogUser();
4          private static String token;
5          private String context;
6          private String ip;
7          private String device;
8          private TaskTime task;
9          private MemoryDB memory;
10         public Monitoring() {
11                 task = TaskTime.getInstance();
12                 memory = MemoryDB.getInstance();
13         }
14         ...
15         @AfterReturning( pointcut="execution(public * *(..)) && @annotation(Audited)", returning="result")
16         public void afterReturning(JoinPoint jp , Object result) throws Throwable {
17                 //Obtain the @Audit context
18                 Class<? extends Object> objectTargetClass = jp.getTarget().getClass();
19                 Method[] methods = objectTargetClass.getMethods();
20                 for (Method method : methods) {
21                         Annotation annotation = method.getAnnotation(Audited.class);
22                         if (annotation instanceof Audited) {
23                                 Audited myannotation = (Audited) annotation;
24                                 if (myannotation.context() == null) {
25                                         throw new RuntimeException("Shoud be informed the context of these data [ Mandatory ].");
26                                 } else {
27                                         this.setContext(myannotation.context().name());
28                                 }
29                                 break;
30                         }
31                 }
32                 identifyUser(jp ,result);
33                 identifyFeature(jp , result);
34                 Log log = new Log();
35                 if ( result instanceof java.util.ArrayList) {
36                         @SuppressWarnings("unchecked")
37                         List<? extends Object> list = (ArrayList<? extends Object>) result;
38                         for (Object object : list) {
39                                 log = setRegister(object);
```

```
40                                  log.setIp(this.getIp());
41                                  log.setDevice(this.getDevice());
42                                  log.setContext(context);
43                                  log.setToken(token);
44                                  Calendar data = Calendar.getInstance();
45                                  log.setDate(data.getTime());
46                                  log.setHour(data.getTimeInMillis());
47                                  user.setToken(token);
48                                  TrackLogging o = new TrackLogging();
49                                  o.setLog(log);
50                                  o.setUser(getUser());
51                                  memory.setBuffer(o);
52                          }
53                  } else {
54                          log = setRegister(result);
55                          log.setIp(this.getIp());
56                          log.setDevice(this.getDevice());
57                          log.setContext(context);
58                          log.setToken(token);
59                          Calendar data = Calendar.getInstance();
60                          log.setDate(data.getTime());
61                          log.setHour(data.getTimeInMillis());
62                          user.setToken(token);
63                          TrackLogging o = new TrackLogging();
64                          o.setLog(log);
65                          o.setUser(getUser());
66                          memory.setBuffer(o);
67                  }
68          }
69  }
```

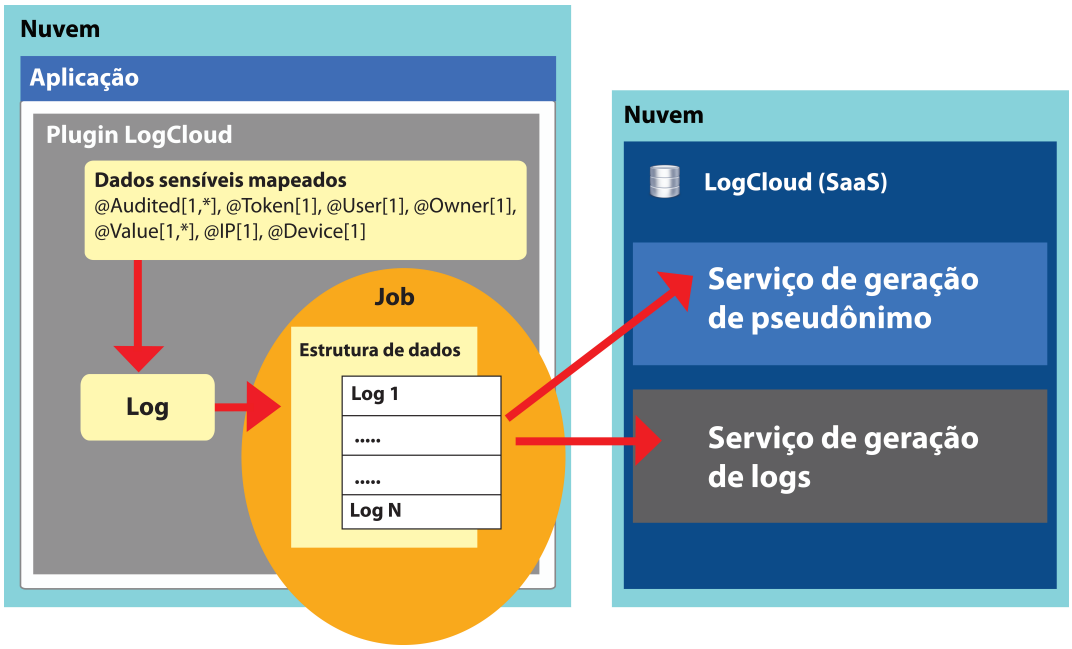

Figura 5.16: Interação do plugin LogCloud com os serviços

Entre as linhas 32 à 69 é apresentado a regra realizada para obter os campos que compõe o

log e os dados coletados são lançados em uma estrutura de dados.

O dado mapeado como sensível é captado através do uso de *annotations* disponibilizados pelo plugin. Após o log ser adicionado em uma estrutura de dado um *job* é ativado e de tempos em tempos extrai o log da estrutura de dados e envia à nuvem. O *job* executa a chamada dos serviços de pseudônimo e na sequência a de geração de logs , como apresentado na Figura 5.16.

5.5 Considerações finais

O serviço de auditoria de acesso ao dado com garantia de privacidade para aplicações visa aumentar a proteção do dado, considerando a necessidade de um controle mais refinado sobre o dado liberado e a necessidade de ter um monitoramento próximo do acesso a dados por um determinado público.

O público pode ser de uma empresa privada ou os cidadãos, como é caso do Portal da Transparência do governo federal brasileiro. O mecanismo disponibilizado na nuvem pode ser utilizado por aplicações que apresentam necessidade, para fins experimentais precisam de aplicações escrita em Java com Maven.

Com base na análise dos cenários de detecção de intrusão, o proprietário tem mais informações sobre os acessos a determinados dados. Assim, tal serviço visa melhorar a política de liberação desses dados pela organização que a usa, pois os meios e recursos para analisar os acessos e origens ficaram mais claros e flexíveis.

O serviço visa ainda auxiliar na percepção do mau uso com base na análise dos cenários de detecção oferecida pela ferramenta. Tal investimento, hoje em dia, tem tanta importância quanto o feito em recursos para evitar a contaminação de máquinas por malwares, vírus e outros códigos maliciosos.

Capítulo 6

ESTUDO DE CASO E AVALIAÇÃO DE DESEMPENHO

O capítulo detalha o estudo de caso da aplicação do serviço de monitoramento em nuvem a dados sensíveis que permite a auditoria e prestação de contas. Inicialmente, são apresentados detalhes da integração do protótipo criado com base no Portal da Transparência do governo brasileiro com o plugin do LogCloud. Por fim, foram realizados testes de desempenho para mensurar o impacto que um serviço como esse exerceria em aplicações que tenham a necessidade de ter o acesso ao dado auditado.

6.1 Considerações iniciais

Muitos estudos têm sido feitos acerca da computação em nuvem com dados sensíveis. Alguns trabalhos, com a mesma linha do LogCloud, mostram muitas maneiras de lidar com dados, prestação de contas e auditoria.

Alguns estudos mostram o uso de auditoria para melhorar a confiabilidade do usuário na nuvem (SCHIFFMAN et al., 2013; HOULIHAN; DU, 2012); arquitetura para garantir a integridade dos dados (GOWRIGOLLA; SIVAJI; MASILLAMANI, 2010); algoritmos para gerar dados anônimos para os serviços de cloud computing (WANG et al., 2010); rastreamento do dado para evitar o mau uso (SUNDARESWARAN et al., 2011; TAN et al., 2012); monitoramento do acesso de dados (STOLFO; SALEM; KEROMYTIS, 2012; SUMTER, 2010); o modo como o provedor de serviços em nuvem utiliza o dado (PEARSON et al., 2012; ZARGARI; SMITH, 2013) e questões referentes à privacidade e à auditoria na nuvem (GRANDISON; THORPE; STENNETH, 2013).

Ressalte-se que alguns dos trabalhos citados apontam a necessidade de validar a avaliação de desempenho para garantir o mínimo impacto no sistema e a menor sobrecarga para permitir

a sua adoção.

Um dos pontos destacados pelos autores Sundareswaran et al. (2011) no projeto que os mesmos desenvolveram que permite promoção de prestação de contas distribuída é que não pode introduzir um forte impacto na comunicação ou computação. Se isso acontecesse, esse tipo de problema poderia impedir a viabilidade do projeto.

Assim, os autores analisaram o tempo de criação do log, tempo de autenticação, desempenho do tempo de registro, o tempo da funcionalidade de fundir logs e a sobrecarga de armazenamento do arquivo criado. Todas as métricas definidas pelos autores ajudaram analisar a viabilidade do projeto proposto por eles.

No trabalho de Sundareswaran et al. (2011) é destacado a preocupação com a perda de controle do usuário quando envia o dado para a nuvem. Foi então proposta uma forma de enviar, juntamente com os dados, políticas de controle de acesso e políticas de registro, embutidas em um arquivo JAR. O projeto de Sundareswaran et al. (2011) contou com recursos de auditoria e avaliação em um ambiente real.

Em outro trabalho de Schiffman et al. (2013), a sobrecarga do desempenho foi avaliada e uma prova de conceito do *framework* foi criada para a plataforma na nuvem OpenStack. A essência é definir o impacto do *framework* proposto criado, chamado *Cloud Verifier*, o qual provê monitoramento de serviços na nuvem para os clientes para validar o ambiente e executar como esperado em uma nuvem IaaS.

O *framework* é composto pelo *Cloud Verifier*, que habilita o monitoramento da saúde da própria instância e um *Instance Monitor*, o qual é um serviço para monitorar a instância para detecção de violações de acordo com os requisitos do cliente.

No trabalho de Houlihan e Du (2012), os autores avaliam a sobrecarga do esquema de auditoria de segurança proposto para sistemas de computação na nuvem. Assim, o tempo de alguns processos executados pelo esquema de auditoria foram coletados dando uma noção da sobrecarga do tempo do sistema. O esquema proposto registra cada execução do sistema antes de ser executado e armazena os logs para fornecer uma evidência real do sistema.

Pode-se perceber que nos trabalhos mencionados as avaliações de desempenho foram aplicadas para identificar o impacto da auditoria no sistema. Assim, considerando a forma com que o LogCloud trabalha, passa a ser necessário identificar a sobrecarga adicional no tempo de resposta usando o mecanismo de auditoria em uma aplicação real e a condição mínima na qual o núcleo do LogCloud pode ser implantando para as aplicações invocarem como serviço.

A aplicação na web chama o serviço usando um plugin disponível pelo LogCloud. Esse

plugin precisa ser incorporado na aplicação e pode prejudicar o desempenho.

Os problemas de desempenho podem ser resultado dos passos necessários para criar o log: identificar os dados sensíveis, acesso ao *webservice* para o pseudônimo baseado na auditoria de pseudônimo e o acesso ao *webservice* para criar os arquivos de logs.

Então, foi coletado o tempo de resposta do acesso à página contendo dado sensível. Para isso, foi definida uma sequência para o teste do tipo *"Baseline test"*; e o *"Load Test"* foi usado analisar o comportamento mínimo do ambiente escolhido para *deploy* do aplicativo com o plugin LogCloud. Na situação do LogCloud, mecanismo de monitoramento, os testes apresentaram uma boa ideia da sobrecarga do tempo de resposta do sistema.

6.2 Estudo de caso

Os problemas que envolvem a possibilidade de auditar o acesso ao dado em aplicações, como levantado no capítulo inicial deste trabalho, apresentam possíveis ideias de aplicações com a possibilidade de utilização do serviço LogCloud.

Entre as aplicações identificadas para testar o mecanismo proposto, o estudo de caso resolveu focar na validação da ideia do serviço através da simulação do teste de desempenho em um protótipo do Portal de Transparência do governo federal.

No Portal de Transparência existe uma área onde o salário dos servidores federais são liberados sem nenhum tipo de controle de acesso. Como o serviço garante que o acesso ao dado marcado como sensível seja auditado e, através dos logs, permite um monitoramento dos acessos, o Portal enquadrou-se no objetivo proposto pelo LogCloud.

O governo brasileiro nos últimos anos com as leis de transparência impõe a necessidade dos órgão públicos a trabalhar de forma transparente com os seus processos e aplicações de dinheiro. Consequentemente, a divulgação de salários são realizadas com a ideia de deixar claro para a população de que a aplicação dos tributos da população são bem aplicados.

Contudo, o controle que o governo idealiza ao deixar tudo acessível para a população resulta em uma perda da privacidade, quando esses dados passam a ser divulgados de forma indiscriminada. O acesso a dados sensíveis de servidores públicos não é identificado e muito menos monitorado para prevenir o mau uso. Por conseguinte, esses dados sensíveis – que revelam a situação financeira, local de trabalho e jornada – passam a ser um grande conjunto de dados que facilitam ações de cibercrimes.

A finalidade da auditoria do acesso ao dado é gerar uma visão sobre a frequência de acessos,

a origem dos acessos e quem acessa. Além disso, ofertar ferramentas para que o proprietário do dado – com base nos cenários de detecção de intrusão – levante possíveis questionamentos sobre os acessos e, ao mesmo tempo, conheça a identidade do usuário somente quando estritamente necessário.

O serviço foi inserido em um protótipo que simula o Portal da Transparência do governo federal. Atualmente, o site não solicita nenhum tipo de autenticação para identificar o usuário. Então, foi adicionado antes de acessar a área denominada sensível uma página de autenticação. A Figura 6.1 apresenta a página atual oficial do Portal e pode-se observar que não é solicitado nenhum tipo de autenticação ao clicar no botão consultar, como apresentado pela seta com a numeração igual a um.

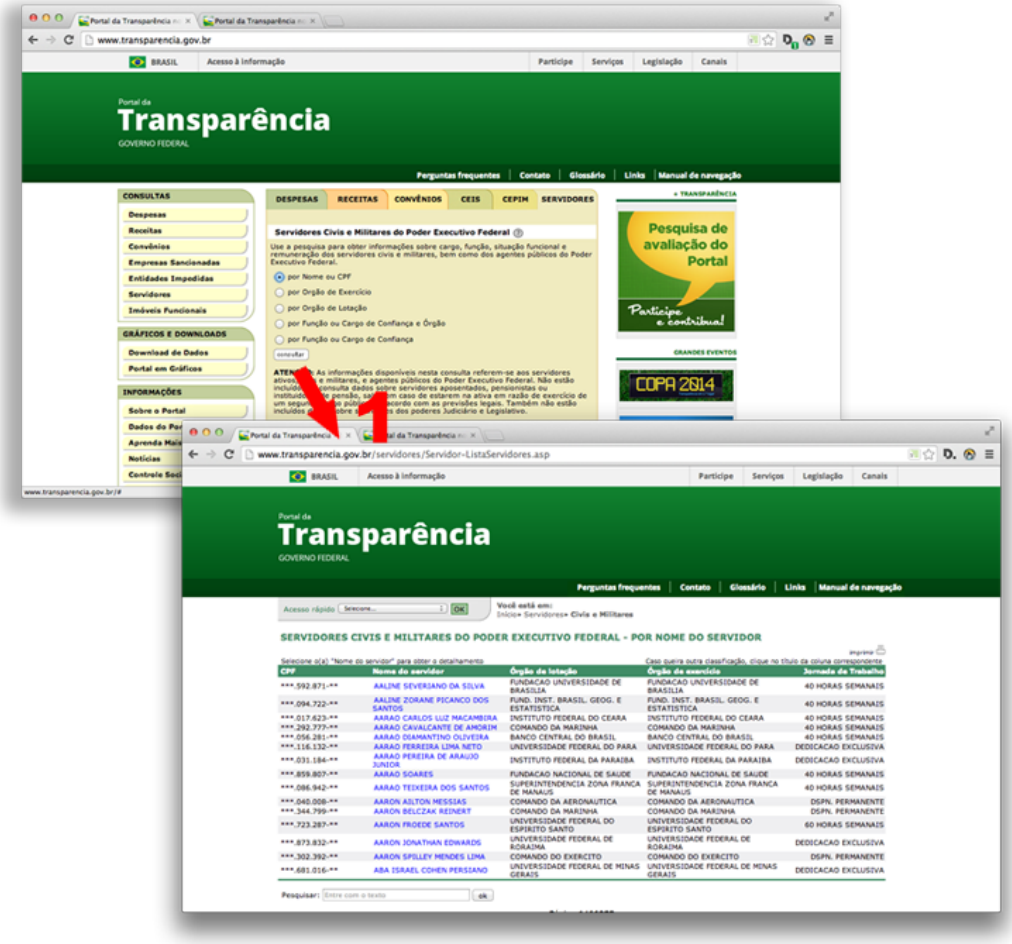

Figura 6.1: Portal da transparência do Governo Federal do Brasil

A alteração realizada no protótipo para contemplar o estudo de caso iniciou-se com a inserção de uma página de cadastro do usuário. Após o cadastro, o usuário deve realizar a autenticação com o email e senha informados. Na sequência, a página valida a autenticação e redireciona a página com dados sensíveis. Ou seja, a mesma página apresentada na Figura 6.1 ao clicar no

botão consultar.

A página de autenticação é a única variação em comparação com a versão original, como apresentado na Figura 6.2 indicadas pelas setas. Ao clicar no botão consultar a seta com numeração um indica a página habilitada solicitando um usuário e senha. E a seta com numeração dois indica a página de cadastro do usuário criada para esta simulação. A modificação foi realizada porque o mecanismo precisa que seja informado o usuário que está acessando, então, optou-se pela inclusão da área de autenticação. No entanto, pode ser explorado outra forma de obter o usuário de uma maneira mais transparente desde que siga princípios apresentados na seção 5.3.1 do Capítulo 5 em relação aos dados necessários para geração do pseudônimo e identificação do usuário.

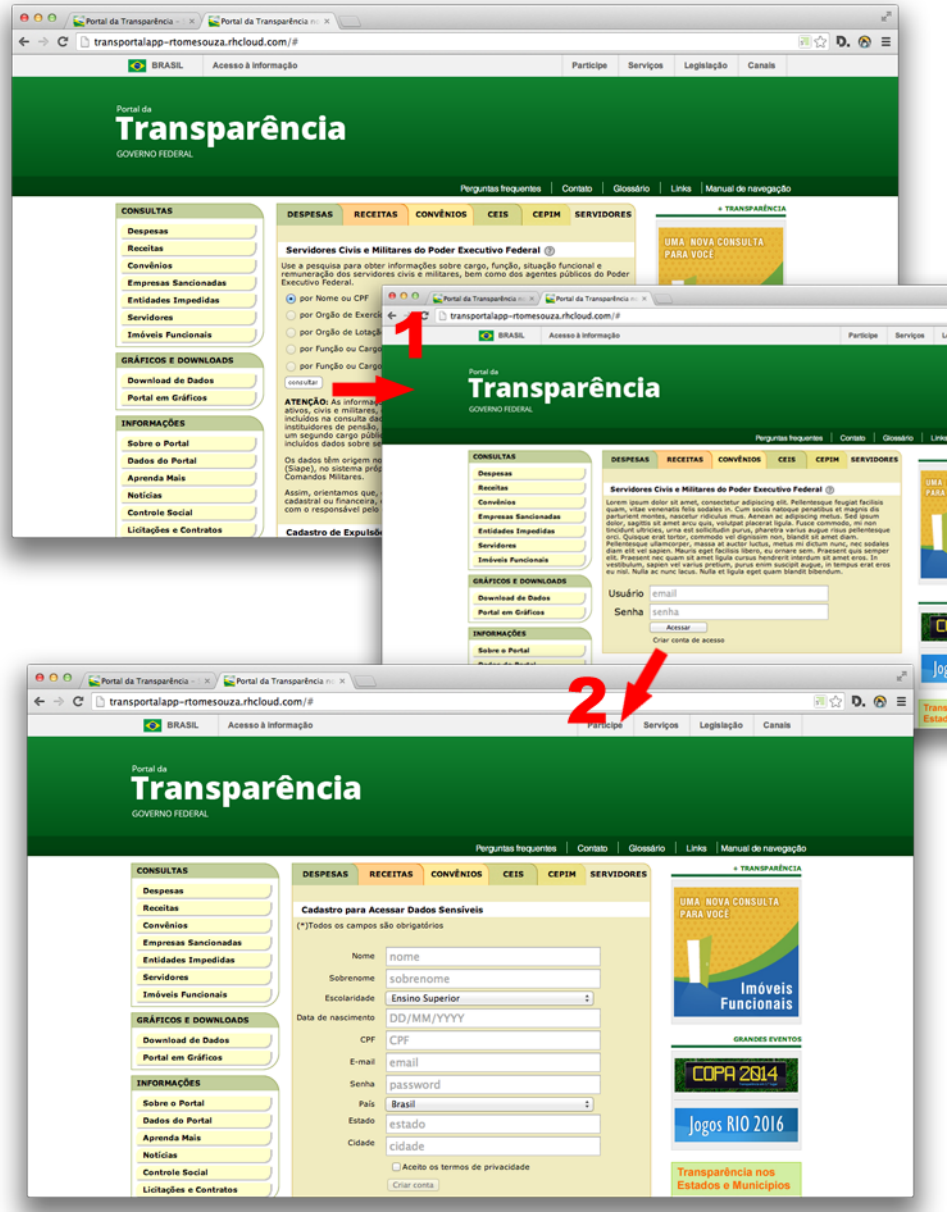

Figura 6.2: Protótipo do portal da transparência do Governo Federal do Brasil

Após a inserção dessa nova página, foi necessário incluir a biblioteca referente ao plugin LogCloud para ser possível mapear os dados sensíveis apresentados na página de remuneração, apresentada na Figura 6.3. No entanto, a página apresentada na Figura 6.3 só é alcançada pelo usuário após a seleção de um servidor público federal específico, como apresentado na Figura 6.3. A seta com a numeração um indica que ao clicar no botão denominado remuneração é acessado uma outra página indicado pela seta com numeração dois, sendo esta página a que está tendo os dados monitorados pelo plugin LogCloud.

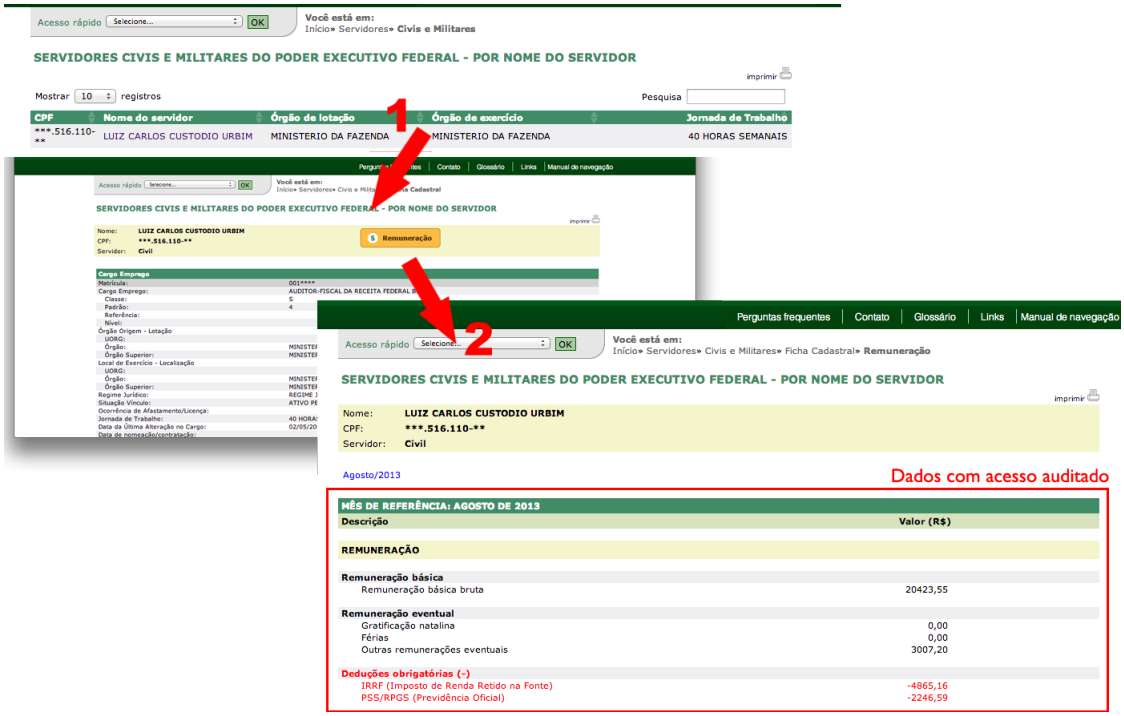

Figura 6.3: Protótipo do portal da transparência do Governo Federal do Brasil

O acesso ao dado é auditado com base na utilização do plugin do LogCloud, que está inserido no protótipo como sendo uma biblioteca. No código 6.1, é listada a classe que modela o domínio dos dados que retornam dados de salários dos servidores federais. Nesse código é possível visualizar o uso dos *annotations* @Owner e @Value, sendo esses *annotations* responsáveis por definir o proprietário do dado e os seus respectivos dados acessados.

Código 6.1: Remuneração do Servidor Federal

```
1 public class PublicFederalServerRemuneration {
2         private String idServidorPortal;
3         @Owner
4         private String nome;
5         @Value
6         private String cpf;
7         @Value
8         private List<Payment> pagamentos;
9         ....
```

```
10 }
```

O Código 6.2 refere-se ao momento em que os dados definidos como sensíveis devem ter o seu acesso auditado. Sendo assim, é usado o *annotation* denominado @Token, @Ip, @Device, @User e o @Audited com o valor do campo de contexto definido como FINANCIAL. Após a invocação do método, automaticamente o retorno referente à coleção denominada *"PublicFederalServerRemuneration"* é interceptada pelo LogCloud. Junto com essa coleção de dados são repassados dados do usuário.

Código 6.2: Auditar o acesso ao dado marcado como sensível pelo proprietário do dado

```
1  public class PublicFederalServerDAO extends GenericDAO<PublicFederalServerDAO> {
2          @Token
3          private String token = "98↩
               e3cfd85dcd284c1e2e1edce38f5b9c4e9dce3d4bf8905660189edd2dc08da5";
4          @User
5          private LogUser logUser;
6          @Ip
7          private String ip;
8          @Device
9          private String device;
10         private CitizenUserSystemDAO citiUserSysDAO = new CitizenUserSystemDAO();
11         ...
12         @Audited(context=Type.FINANCIAL)
13         public PublicFederalServerRemuneration getServerDetailRemuneration(String ↩
               idServerPortal, String month, String year, String hostname, String device, ↩
               String idAcesso) {
14                 Query query = new Query();
15                 query.addCriteria(Criteria.where("idServidorPortal").is(idServerPortal)↩
                       );
16                 query.addCriteria(Criteria.where("pagamentos.ano").is(year));
17                 query.addCriteria(Criteria.where("pagamentos.mes").is(month));
18                 PublicFederalServerRemuneration server = getConnection().findOne(query,↩
                        PublicFederalServerRemuneration.class, "publicFederalServer");
19                 this.setLogUser(citiUserSysDAO.getCitizenDetails(idAcesso));
20                 this.setDevice(device);
21                 this.setIp(hostname);
22                 return server;
23         }
24 }
```

Por fim, o método existente na classe *PublicFederalServerDAO* é invocado pelo *controller* denominado *PublicFederalServerController*, apresentado no Código 6.3. Quando a página web que contém os dados que precisam ser auditados, o plugin do LogCloud é invocado e inicia o processo de geração de log.

Código 6.3: Invocando a visualização da página com dado sensível

```
1  @Controller
2  @RequestMapping("/restrict")
```

```
3  public class PublicFederalServerController {
4          private PublicFederalServerDAO pfdDao = new PublicFederalServerDAO();
5          private String getIp() {
6                  String ipAddress = ((ServletRequestAttributes)RequestContextHolder.↩
                       currentRequestAttributes()).getRequest().getHeader("X-FORWARDED-FOR↩
                       ");
7              if (ipAddress == null) {
8                  ipAddress = ((ServletRequestAttributes)RequestContextHolder.↩
                       currentRequestAttributes()).getRequest().getRemoteAddr();
9              }
10             return ipAddress;
11         }
12        ....
13         @RequestMapping(value="/serverDetailRemuneration/{idServerPortal}/month/{month↩
               }/year/{year}")
14         @ResponseBody
15         public PublicFederalServerRemuneration getServerDetailRemuneration(↩
               @PathVariable("idServerPortal") String id, @PathVariable("month") String ↩
               month, @PathVariable("year") String year, @RequestHeader("User-Agent") ↩
               String userAgent, Principal principal) {
16                 return pfdDao.getServerDetailRemuneration(id, month, year, getIp(), ↩
                       userAgent, principal.getName());
17         }
18 }
```

Em uma visão geral, a necessidade da inserção do plugin LogCloud no projeto para disponibilização do serviço fez com que a integração ficasse mais transparente e intuitiva para o desenvolvedor. A Figura 6.4 apresenta uma visão geral da inserção do plugin dentro do protótipo do Portal da Transparência, mostrando as principais classes que se relacionam com as dependências do plugin.

6.3 Avaliação de desempenho

A ideia da auditoria como um serviço é fornecer uma forma para o proprietário do dado analisar os logs gerados pelo acesso do cidadão/funcionário através de cenários pré-determinados. O SaaS fornece esse ambiente ao proprietário do dado e por se tratar de um serviço, a organização a qual usar o LogCloud precisa criar uma conta de acesso.

O LogCloud é um SaaS que pode receber dados de vários tipos de aplicação e foi construído em duas partes: a primeira parte está implantada em uma nuvem usando os recursos de escalabilidade e armazenamento fornecidos por este tipo de ambiente e a segunda parte é um plugin a ser incorporado no aplicativo que viabiliza a comunicação com o serviço.

O objetivo é analisar o impacto da adoção de um serviço de monitoramento de dados sensíveis dentro de uma aplicação. Com o propósito de identificar a viabilidade do uso do serviço

LogCloud intermediado pelo plugin. Com respeito à eficiência em termos de tempo de resposta do acesso às páginas com dados monitorados, teste de carga para averiguar o suporte do servidor quando o plugin é adicionado a aplicação e o consumo de recursos tais como CPU e memória. Do ponto de vista e no contexto de analista de software.

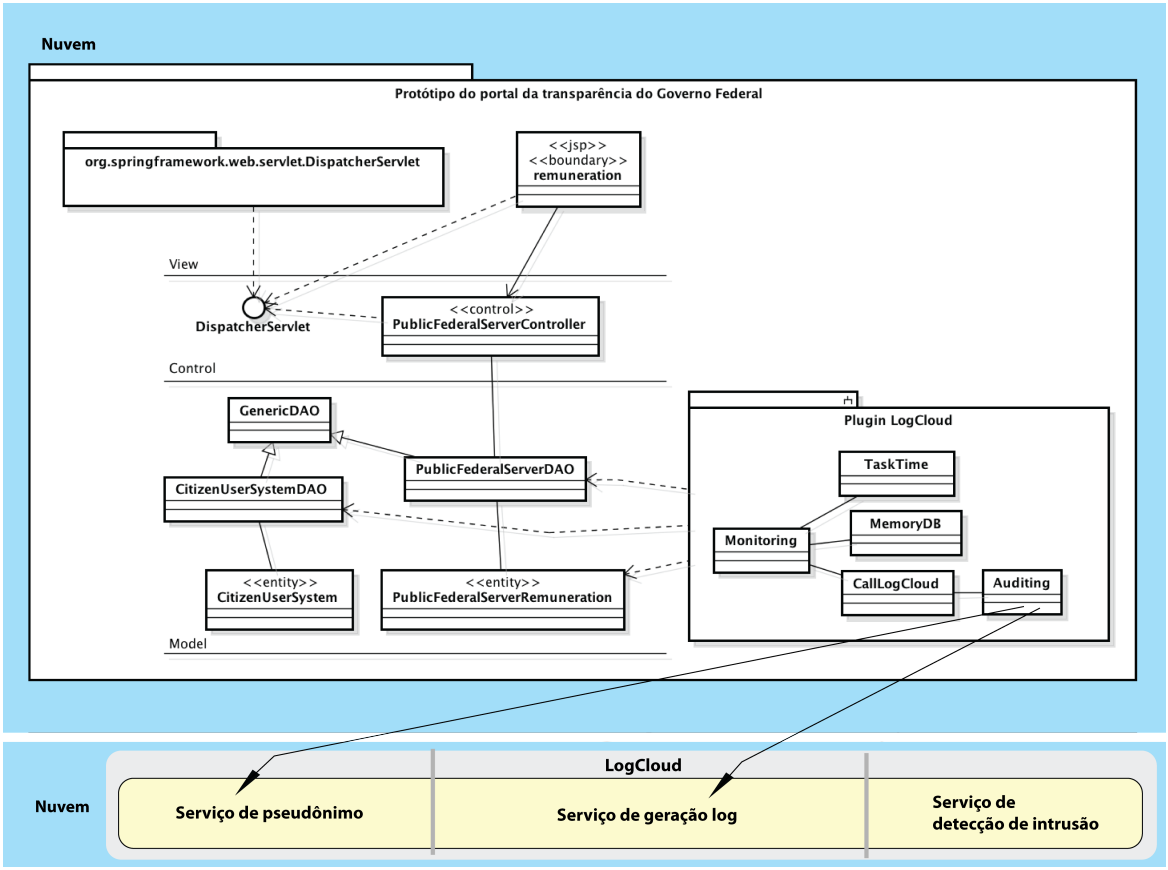

Figura 6.4: Visão geral do plugin inserido no protótipo

O fluxo de interação é apresentado na Figura 6.5. Cada acesso do usuário ao dado sensível faz o plugin LogCloud acionar uma ação para acessar o SaaS na nuvem para registrar quem acessou o dado sensível. Então, o proprietário do dado pode analisar o fluxo de acesso através da área administrativa disponível na web usando a conta criada.

Os testes simulados para avaliar o LogCloud foram feitos no protótipo baseado no Portal da Transparência do Brasil. Todos os dados do Portal estão disponíveis para download. Assim, foi possível recriar a aplicação que mostra os dados sensíveis para os cidadãos usando os mesmos dados disponíveis no Portal original.

Para este estudo de caso, foi criada uma conta na nuvem OpenShift com um total de 3GB de armazenamento e 1.5 GB de memória, sendo toda a capacidade de armazenamento dividida em *gears*. Os *gears* são *containers* com recursos limitados que rodam os softwares definidos pelo desenvolvedor ou cartuchos. Além disso, é possível definir a escalabilidade do software

para permitir que a nuvem lide com o tráfego quando necessário.

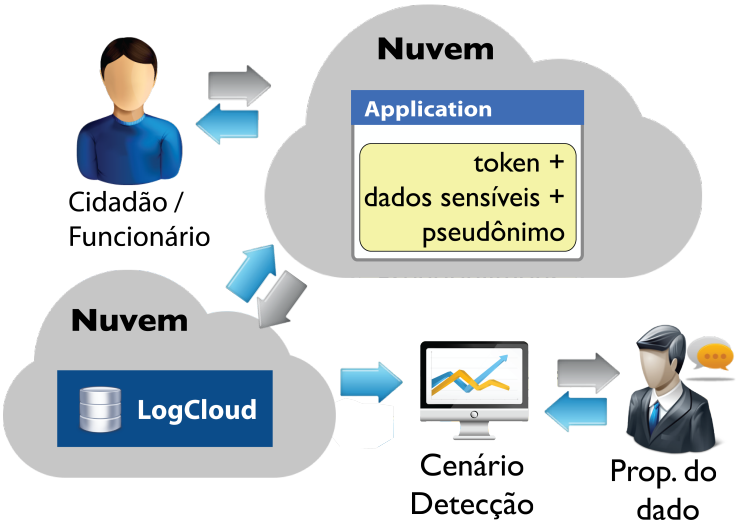

Figura 6.5: Arquitetura do serviço - LogCloud

A Figura 6.6 mostra o modelo da nuvem PaaS criado para implantar o SaaS - LogCloud. A mesma arquitetura apresentada na Figura 6.6 foi adotada para implantar o protótipo do Portal da Transparência usado para avaliar o LogCloud.

Quando o LogCloud e o Portal da Transparência foram implantados na nuvem, o SaaS e a aplicação do portal ocuparam um *gear* e um outro *gear* foi ocupado pelo banco de dados MongoDB. O recurso de dimensionamento de software foi ativado na nuvem para lidar com o tráfego web.

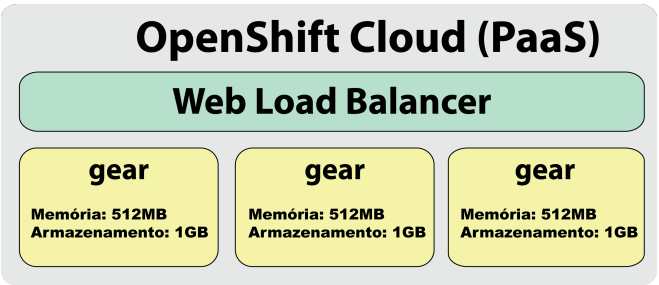

Figura 6.6: Nuvem OpenShift pela Red Hat. Disponível em: www.openshift.com

A Figura 6.7 mostra a interação das duas nuvens criadas para o teste de desempenho e da tecnologia envolvida. Para um teste comparativo e para definir o impacto de um serviço de monitoramento, foi criada uma outra nuvem com a mesma configuração dos outros. Nessa nova nuvem, uma versão do Portal da Transparência foi implantada sem o serviço de monitoramento fornecido pelo LogCloud.

O teste de desempenho foi feito usando o Apache JMeter^{TM} que é uma aplicação projetada para medir o desempenho e testes de carga funcional (JMETER, 2013). Com base na arquitetura

apresentada na Figura 6.7 a viabilidade do uso do serviço de monitoramento foi avaliado através da simulação de acessos ao protótipo do Portal da Transparência que usava o serviço.

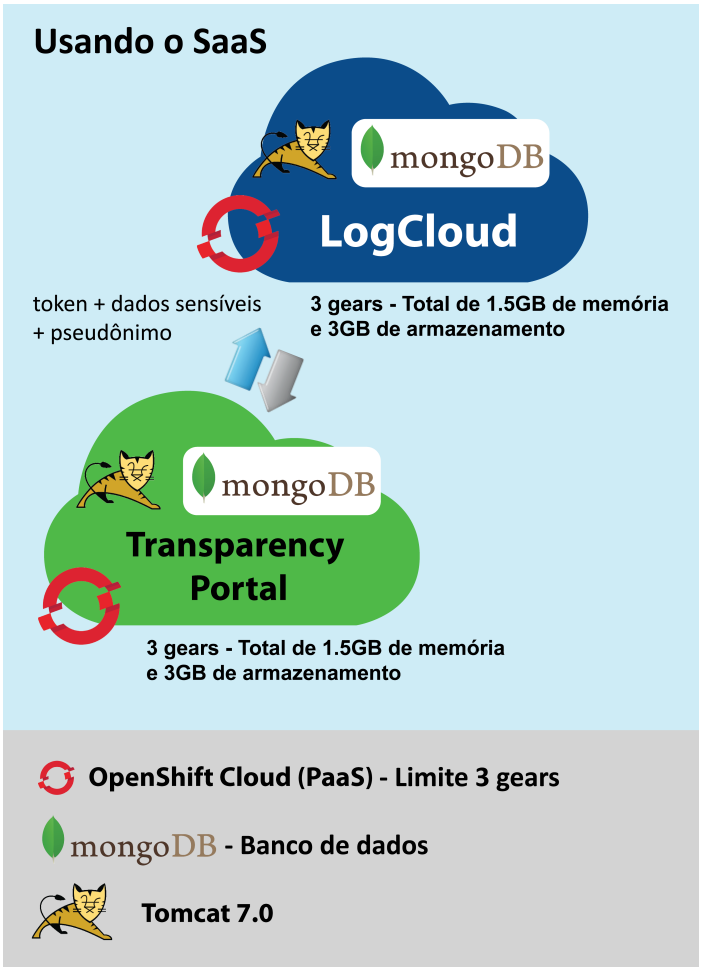

Figura 6.7: Visão geral da arquitetura de implantação.

Outra simulação de acessos foi realizada no protótipo do Portal de Transparência sem o serviço do LogCloud. Sendo possível fazer um teste comparativo e definir o impacto de desempenho com o serviço de monitoramento. O impacto do serviço em uma aplicação foi identificado através de um teste denominado *Baseline*.

O teste *Baseline* usado para medir o tempo de resposta da transação é executado muitas vezes ou por um período de tempo. E o resultado pode definir a degradação de desempenho ocorrida devido ao aumento de requisições de usuários ou *throughput*, por exemplo (MOLYNEAUX, 2009).

A simulação de acesso sequencias para avaliar o serviço de monitoramento em uma aplicação real foi realizada por um computador fora da nuvem onde o LogCloud e o protótipo do Portal da Transparência foram implantados. Assim, o resultado tende a se aproximar da situação de um usuário acessando uma página que está com os dados sensíveis sendo monitorados.

Para simular o acesso, o software JMeter™ (JMETER, 2013) foi usado e, para isso, foi necessário definir a rotina do teste. Para avaliar o impacto do uso do LogCloud no sistema, um usuário deveria requerer 600 vezes a página com dados sensíveis. Tais requisições deveriam ser feitas no protótipo do Portal da Transparência com e sem o serviço LogCloud.

As requisições foram disparadas a partir de um computador com 4GB de RAM com um processador Intel CORE 2 Duo. O software JMeter™ simulou um usuário para ambas as aplicações e os testes foram aplicados em tempos diferentes.

A diferença entre os tempos de respostas foi obtida através do *baseline test*, aplicado aos protótipos do Portal da Transparência com simulações de acessos sequenciais na página com dado sensível. Os testes foram executados em uma rede em que a taxa do ping no período de teste era 17ms, taxa de download com 54.31Mbps e a taxa de upload 113.26Mbps.

O primeiro cenário de testes analisou a sobrecarga em termos de tempo de resposta, considerando o plugin acessando o serviço na nuvem levando em consideração o mapeamento dos dados sensíveis através de disparo sequencias de requisições a uma página em que o acesso ao dado estava sendo monitorado. O primeiro cenário tem como Hipótese nula (H0) é que a adição do plugin não afeta a visualização das páginas; a Hipótese alternativa (H1) é que a adição do plugin gera um acréscimo maior de 10s no acesso às páginas em que os dados são monitorados e a Hipótese alternativa (H2) é que a adição do plugin gera um acréscimo menor que 10s no acesso às páginas em que os dados são monitorados.

O resultado do impacto é apresentado na tabela 6.1. Nesta tabela é possível perceber que o tempo médio decorrido desse conjunto de resultados quando não usa o LogCloud é 11 ms menor do que quando se usa o serviço de monitoramento. Então, é possível concluir um impacto mínimo com a adoção do LogCloud considerando válido a Hipótese (H2) e descartando a Hipótese (H0) e (H1).

Tabela 6.1: Teste Baseline

Application	**Times**	**Average (ms)**	**Std. Dev.**
Portal da transparência	600	200	34,82
Portal da transparência usando SaaS	600	211	47,19

O bom desempenho do serviço de monitoramento é devido ao plugin que se comunica com o SaaS. Todas as solicitações feitas para o serviço são colocadas em uma estrutura de dados para se comunicar posteriormente com o SaaS, liberando o sistema para o curso normal. Obviamente, os serviços que permitem ao LogCloud funcionar corretamente também foram ajustados para garantir um melhor desempenho, já que um impacto mínimo e não degradação do

desempenho do sistema são aspectos esperados quando se pretende aplicar a auditoria (FISCHER-HÜBNER, 2001).

Outro teste de desempenho foi o teste de carga. A ideia é forçar a aplicação que usa o plugin LogCloud próximo da realidade (MOLYNEAUX, 2009). Fazendo esse tipo de teste é possível compreender os limites da concorrência.

O segundo cenário de testes analisou a sobrecarga do servidor em que a aplicação que usa o plugin que acessa o serviço na nuvem está instalada. O teste realizou uma série de disparo sequencias de requisições a uma página em que o acesso ao dado estava sendo monitorado. O segundo cenário tem como Hipótese nula (H0) é que a adição do plugin não deixa o servidor inoperante com os acessos sequencias que o plugin executa e que não tem um aumento simultâneo de threads (usuário) com o tempo de resposta do servidor; a Hipótese alternativa (H1) é que a adição do plugin deixa o servidor inoperante com os acessos sequencias que o plugin executa e que há aumento simultâneo de threads (usuário) com o tempo de resposta do servidor e a Hipótese alternativa (H2) é que a adição do plugin não deixa o servidor inoperante com os acessos sequencias que o plugin executa e que há um aumento simultâneo de threads (usuário) com o tempo de resposta do servidor.

A Figura 6.8 mostra uma evolução de 250 usuários requisitando simultaneamente duas páginas com dados sensíveis por 40 vezes, totalizando 20.000 requisições para o servidor LogCloud, disponível na nuvem. A Figura 6.8 mostra dependências do aumento simultâneo de *threads* (usuário) com o tempo de resposta do servidor. Logo, a hipótese H0 e H1 podem ser descartadas e prevalece a hipótese H2.

Na simulação mostrada na Figura 6.8 é necessário considerar a memória disponível no servidor da nuvem, ou seja, só estão disponibilizados 512 MB. E um ponto interessante é que o PaaS da OpenShift recomenda o uso de 1GB para bancos MongoDB. No entanto, todos os testes foram executados em *gears* com apenas 512MB. A escolha por essa configuração de *gear* é para simular a situação extrema em que o plugin do LogCloud, juntamente com um aplicativo, podem ser implantado. Outro ponto é que a nuvem tem apenas três *gears* e dois estão ocupados restringindo o recurso escalável fornecido pela nuvem OpenShift.

Outro aspecto analisado foi o impacto sobre os recursos do servidor, tais como a memória e o CPU. Os experimentos apresentados nas Figuras 6.9, 6.10, 6.11 e 6.12 foram realizados usando a ferramenta *BlazerMeter*^{TM}, disponível em: *http://blazemeter.com/about-blazemeter*. A ferramenta *BlazerMeter*^{TM} disponibiliza a criação de testes de carga oferecendo diferentes maneiras de avaliar o desempenho.

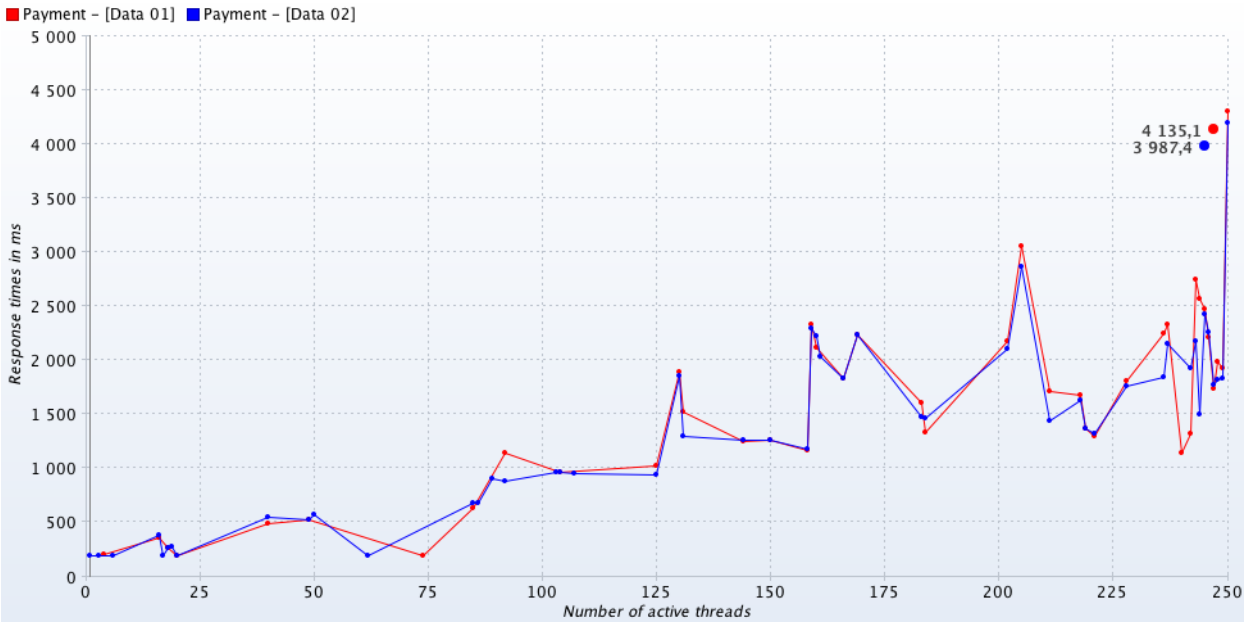

Figura 6.8: Teste de Carga.

No *load test* denominado *Gradual load*, apresentado nas Figuras 6.9, 6.10, 6.11 e 6.12, foi configurado um teste para carregar gradualmente 50 usuários acessando uma página com dados sensíveis por uma hora. Para esse teste, duas versões de um protótipo do Portal da Transparência brasileiro foram geradas. Em uma versão o plugin LogCloud foi incorporado e em outra versão o plugin foi removido.

Ambos os aplicativos foram implantados na nuvem OpenShift. O modelo PaaS da OpenShift oferece nuvens, como mostrado na Figura 6.7. Logo, a única diferença do cenário montado para esse teste foi a desativação da opção de tornar o aplicativo escalável.

O terceiro cenário de testes analisou o tempo de resposta coletado pela percepção do desempenho do sistema para verificar o consumo do CPU e memória em um servidor onde o aplicativo que usa o plugin LogCloud está instalado. O terceiro cenário tem como Hipótese nula (H0) é que a adição do plugin não representa adição de nenhum consumo extra ao servidor tanto em CPU quanto em memória; a Hipótese alternativa (H1) é que a adição do plugin representa adição de um consumo de CPU e memória maior que 50% e a Hipótese alternativa (H2) é que a adição do plugin representa adição de um consumo menor que 50%.

Considerando as hipóteses levantadas para o teste pode aceitar a hipótese H2 em detrimento as hipóteses H1 e H0. Pois, analisando a Figura 6.9 e a Figura 6.10, é possível perceber uma discreta variação em termos percentuais do tempo de resposta recolhido a partir do desempenho do sistema percebido. Assim, se nos testes fossem percebida carga de 70%, sabe-se que o servidor não suportaria. Assim, o tempo de resposta coletado pela percepção do desempenho

do sistema permite obter a mesma conclusão, segundo BlazeMeter (2014).

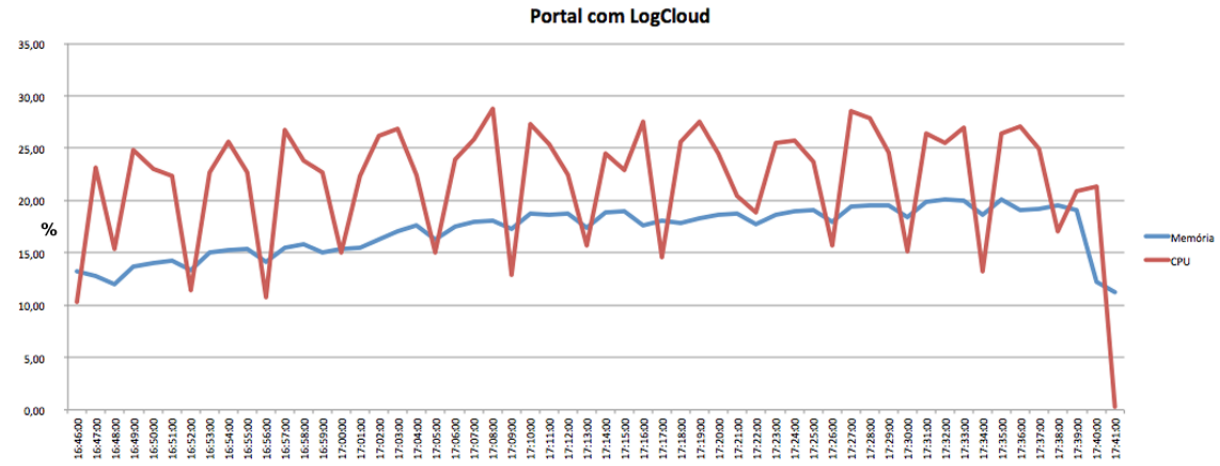

Figura 6.9: Recursos do servidor na Nuvem

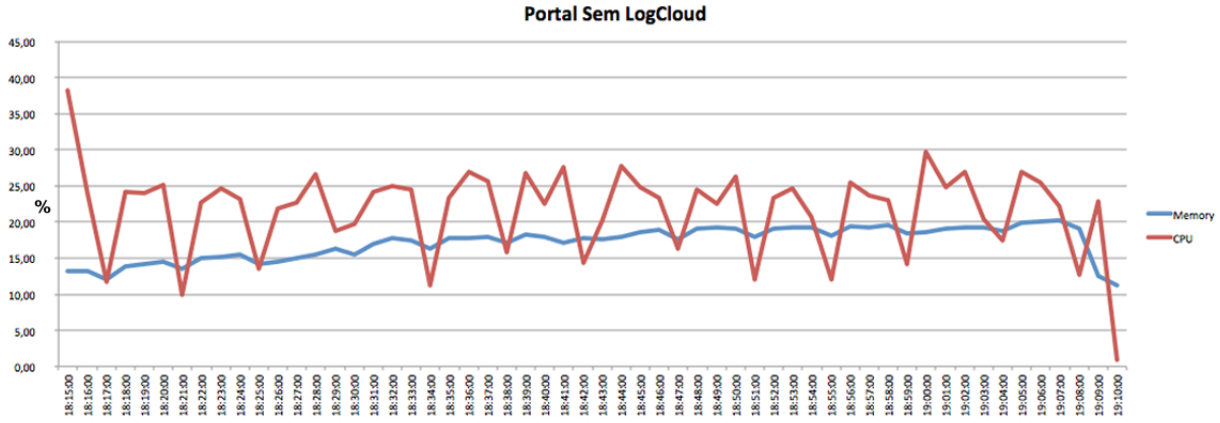

Figura 6.10: Recursos do servidor na Nuvem

Os testes foram executados durante uma hora nos protótipos do Portal da Transparência. Na Figura 6.10, houve um pico de uso do CPU logo no início, provavelmente em decorrência da inicialização do servidor da nuvem. Fora essa variação, a comparação da versão do protótipo que usa o serviço de monitoramento com a versão sem o serviço aponta uma discreta variação, como observado nas Figuras 6.11 e 6.12.

A abordagem adotada por esse mecanismo para monitorar os dados com a geração de logs na nuvem, decorrente do acesso à pagina com dado sensível, criou um *delay* entre a ação de acesso e o registro do log gerado pelo acesso ao dado sensível na nuvem. Esse *delay* proporcionou benefícios que garantem um impacto mínimo na aplicação. No entanto, pode gerar um desconforto ao proprietário do dado, que sempre vai ter alguns segundos de atraso para anali-

sar os cenários de detecção disponibilizados no serviço que é dependente dos logs de acesso armazenados na nuvem.

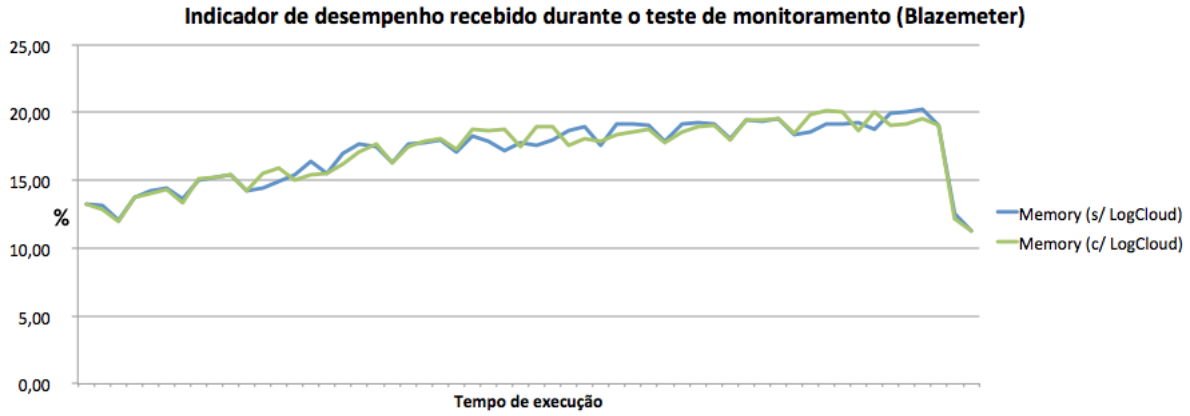

Figura 6.11: Comparativo (Memória)

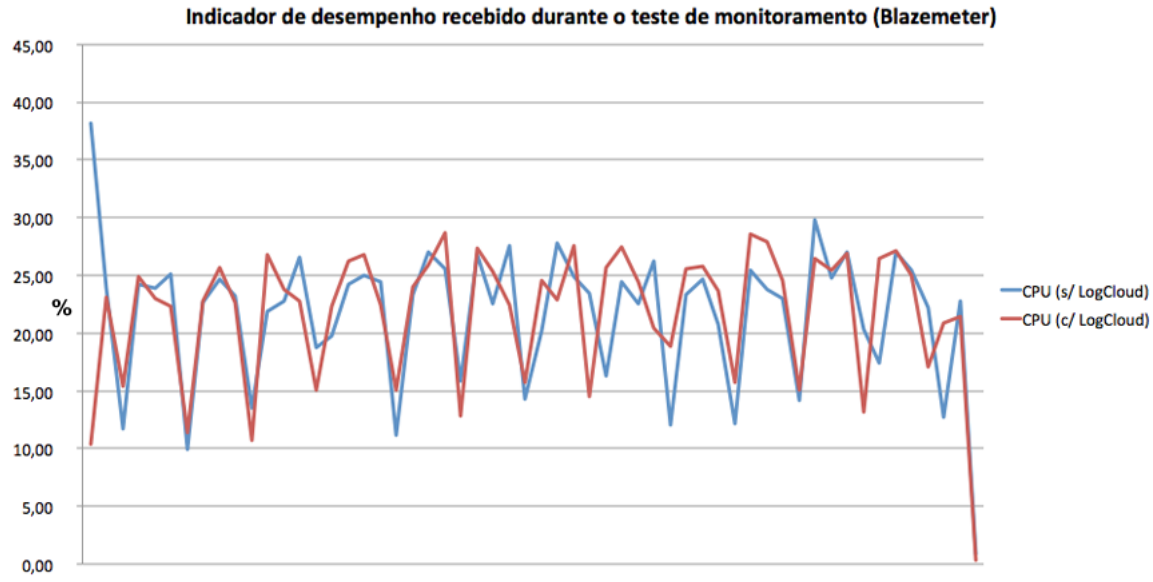

Figura 6.12: Comparativo (CPU)

A avaliação de desempenho ocorreu dentro de um ambiente controlado limitando o escopo desses resultados para este cenário e que a generalização de tais resultados requer novos estudos em um ambiente de produção, caracterizando as ameaças à validade externa. A validade da conclusão está relacionada as medidas coletadas envolvendo as ferramentas utilizadas e os cuidados realizados para a tomadas das análises apresentadas. A validade interna é assegurado de que os resultados foram obtidos em decorrência da execução da sequencia de testes gerados pelas ferramentas JMeter e Blazer. Por fim, a validade à construção corresponde a adequação entre a teoria e a observação, ou seja, o testes selecionados foram adequados para poder identificar o impacto do serviço em uma aplicação.

6.4 Considerações finais

O uso do serviço chamado LogCloud foi avaliada por meio de avaliação de desempenho. O serviço se propõe monitorar o acesso ao dado mapeado como sensível e permite ao proprietário do dado auditar os acessos realizados quebrando o sigilo de quem acessou a partir das análises realizada pelo proprietário do dado com o intermédio dos cenários de detecção de intrusão. A avaliação foi realizada por meio de um protótipo que simula o Portal da Transparência do governo brasileiro.

Para definir a avaliação do impacto da utilização do serviço por intermédio do plugin foi necessário definir o que seria a sobrecarga e quais as métricas a serem coletadas durante os testes. O impasse foi resolvido com a definição da coleta do tempo de resposta do acesso de página mapeada com dados sensíveis através do baseline test, a sobrecarga do servidor com a análise de uma grande quantidade de solicitações e o consumo dos recursos tais como CPU e memória. Entender a sobrecarga do tempo de resposta é o ponto principal porque, adicionando um mecanismo de auditoria, pode-se diminuir o desempenho quando se cria o log que, nesse mecanismo de auditoria, é criado quando dados sensíveis são acessados.

Um desafio encontrado ao desenvolver um serviço como esse foi o plugin usado para comunicar com o serviço. A barreira principal identificada foi como desenvolver a criação de log sem prejudicar a visualização rápida de dados durante o acesso ao dado sensível. Para resolver tal problema de desempenho, foi necessário adicionar os logs em uma estrutura de dados na memória e escalonar um *job* para acessar a estrutura que de tempos em tempos acessa o *webservice* sem impactar a visualização dos dados.

A conclusão da avaliação de desempenho apresentou que há viabilidade do uso do serviço de monitoramento na perspectiva de uma aplicação que necessite ter o acesso aos dados sensíveis monitorados. Tal serviço que implica na adição de um plugin na aplicação demonstrou ser viável e apresenta um baixo impacto tanto na questão de interação do usuário quando visualiza a página em que o acesso é auditado, quanto na questão do servidor que dentro dos limites impostos se apresentou operante e que o consumo de CPU e memória apresentou apenas uma pequena variação não ultrapassando 50% de capacidade do servidor.

Capítulo 7

CONCLUSÃO

O avanço tecnológico permite usar os meios digitais para divulgação de dados, impactando de forma positiva e negativa a vida das pessoas. O aspecto positivo é que o acesso aos dados se tornou mais fácil, mais rápido e menos burocrático. Entretanto, o acesso rápido e o compartilhamento tornam-se cada vez mais instantâneos, influenciando de forma negativa a vida das pessoas ao se analisar os dados sob o aspecto da privacidade.

A necessidade de controlar o acesso ao dado pertencente ao usuário, a computação em nuvem com as incertezas sobre como este dado está sendo manuseado pelos provedores, a preocupação das empresas em controlar a forma como os dados são usados por seus funcionários em serviços na nuvem e a necessidade de liberar dados revelando a situação financeira de seus servidores foram pontos importantes que nortearam os princípios deste trabalho.

O mecanismo em nuvem de monitoramento a dados sensíveis tem como objetivo ser ofertado como um serviço que melhora o controle sobre o dado liberado pelo seu proprietário. A abordagem permite interceptar os dados sensíveis com base em um padrão, usando-se anotações, garantindo a interoperabilidade e permitindo analisar os acessos com o uso de cenários de detecção para auxiliar na decisão sobre como melhorar as políticas destinadas a liberar dados com base na compreensão dos acessos e identificação do mau uso.

O mecanismo acaba utilizando uma adaptação da auditoria de pseudônimo, proposta neste trabalho. Tal adaptação é aplicada em um cenário bem diferente da proposta original da auditoria de pseudônimo. Assim como a proposta original, também garante a proteção da identidade do usuário contido no log, com uma mistura da chave criptográfica do usuário que efetuou o acesso e da chave do proprietário do dado, diferenciando-se da proposta original.

Os ganhos com essa adaptação é que a identidade do log é protegida pela chave do administrador e pela chave do usuário. Apesar de pedaços da chave serem gravados no log para fins do

uso do cenário de detecção, a descoberta só é possível com a reversão do pseudônimo, usando-se a chave privada do usuário e a chave simétrica do proprietário do dado. Mesmo que alguém acesse a base de dados, esse tipo de manipulação torna-se inviável porque a chave privada do usuário na tabela na qual se concentram todos os usuários do mecanismo é cifrada com a chave simétrica do proprietário.

O mecanismo em nuvem de monitoramento a dados sensíveis foi implementado na forma de um serviço a ser requisitado por aplicações que precisam ter o acesso ao dado auditado. Consequentemente, um monitoramento do acesso é obtido com os logs captados e armazenados na base do mecanismo. Para essa validação foi implementado um protótipo do Portal da Transparência do governo brasileiro, que possui um problema no qual o mecanismo se enquadrava. Dessa forma, o portal requisita o mecanismo na forma de um serviço.

O uso do serviço de auditoria e prestação de contas chamado LogCloud teve sua viabilidade avaliada por testes de desempenho. Para isso, foram aplicados testes, tais como o *load test* e *baseline test*. Procedeu-se a uma avaliação do tempo de resposta, usando-se o serviço implantado na nuvem. Então foi medido o tempo de resposta por meio de um *baseline test*, no qual foi possível observar um impacto mínimo do mecanismo de auditoria sobre a aplicação. E o *load test* mostra as configurações mínimas em que o plugin do LogCloud pode trabalhar.

O serviço LogCloud possui algumas limitações, entre as quais a necessidade de uma padronização inicial na forma de captar os dados definidos como sensíveis. Além disso, o foco inicial deste projeto são somente dados, não incluindo documentos do tipo foto ou multimídia. Outra questão é a dos cenários de detecção montados com base no padrão de captação dos dados sensíveis. Entretanto, por se tratar de um serviço a ser ofertado para várias aplicações, foi percebida uma necessidade de padronização, pois existe uma gama enorme de tipos e conjuntos de dados que o serviço poderia receber.

O serviço é um passo inicial que visa melhorar o controle sobre o que é liberado, auxiliando, consequentemente, na percepção de mau uso e na melhoria da compreensão do acesso por meio de uma visão geral dos acessos. Sendo assim, acredita-se na possibilidade de trabalhos futuros na criação de serviços que possibilitem ao proprietário do dado bloquear o acesso a usuários nos cenários de detecção de intrusão.

Como proposta de trabalhos futuros indicam-se a criação de um módulo de regras que gere notificações de tempos em tempos, analisando a base de dados. Tal regra teria a função de alertar ao proprietário do dado que é dono do dado e a pessoa que tem os dados visualizados. Outras propostas estariam relacionados a criação de um modelo que estenderia a funcionalidade do mecanismo no sentido de poder ser aplicado a qualquer tipo de aplicação computacional, ou

seja, tanto ambiente *web* ou móvel. Atualmente com tanto aplicativos móveis requisitando acesso sob os dados pessoais em dispositivos celulares parece ser um caminho interessante para estender as funcionalidade de um serviço de monitoramento.

O software como serviço LogCloud implementado armazena os logs criados a partir de requisições de aplicações, garantindo uma forma interoperável de troca de dados. Além disso, permite ao proprietário do dado analisar os logs por intermédio dos cenários de detecção, ajudando a compreender o acesso a dados que irá conduzir a decisão sobre a sua liberação. Também ajuda a garantir um acesso transparente, i.e., o usuário tem o livre arbítrio de acessar dados sensíveis e terá preservada a privacidade do acesso. Mas o proprietário do dado pode revelar a identidade quando julgar necessário, baseado na análise feita por meio dos cenários. Para isso se fez necessária uma adaptação na auditoria de pseudônimo.

A contribuição do trabalho se situa no limiar de outro tema, o de auditar com garantia de privacidade. E a sua contribuição é a proposta de um mecanismo em nuvem de monitoramento de dados sensíveis que permite ao proprietário monitorar o acesso a dados por meio do armazenamento de logs dos acessos aos dados em um ambiente externo à aplicação: a nuvem. Essa proposta protege a privacidade da identidade do acesso de terceiros, garante um julgamento imparcial e possibilita acessar os dados de forma monitorada e transparente.

Auditoria e prestação de contas como serviços podem liberar a equipe para se concentrar no que é importante em um aplicativo. O serviço se incumbe de fazer o trabalho, que é monitorar o acesso e garantir a prestação de contas.

O serviço pode melhorar o acesso a dados sensíveis de uma organização, oferecendo conhecimento para compreender o uso de dados. Isto pode ajudar a conduzir as decisões de liberação de conteúdo. Em geral, a adoção de um mecanismo de auditoria tem alguns custos, compensados com os ganhos que, no caso do LogCloud, permitem simultaneamente auditoria e preservação da privacidade do usuário.

REFERÊNCIAS

ACESSOAINFORMACAO.GOV.BR. *Anteprojeto de Lei de Proteção de Dados Pessoais*. Abril 2014. Disponível em: <http://www.acessoainformacao.gov.br/acessoainformacaogov/publicacoes/anteprojeto-lei-protecao-dados-pessoais.pdf>.

ACQUISTI, A.; ADJERID, I.; BRANDIMARTE, L. Gone in 15 seconds: The limits of privacy transparency and control. *Security Privacy, IEEE*, v. 11, n. 4, p. 72–74, 2013. ISSN 1540-7993.

ALMULLA, S.; YEUN, C. Y. Cloud computing security management. In: *Engineering Systems Management and Its Applications (ICESMA), 2010 Second International Conference on*. [S.l.: s.n.], 2010. p. 1 –7.

ANTON, A.; EARP, J.; YOUNG, J. How internet users' privacy concerns have evolved since 2002. *Security Privacy, IEEE*, v. 8, n. 1, p. 21–27, 2010. ISSN 1540-7993.

BISHOP, M. A. *The Art and Science of Computer Security*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002. ISBN 0201440997.

BISKUP, J.; BRÜGGEMAN, H. H. The personal model of data:: Towards a privacy-oriented information system. *Computers & Security*, v. 7, n. 6, p. 575 – 597, 1988. ISSN 0167-4048. Disponível em: <http://www.sciencedirect.com/science/article/pii/0167404888900090>.

BLAZEMETER. *Performance Metrics for Websites*. Abril 2014. Disponível em: <http://community.blazemeter.com/knowledgebase/articles/34412-performance-metrics-for-websites>.

BRANDIMARTE, L.; ACQUISTI, A.; LOEWENSTEIN, G. Misplaced confidences: Privacy and the control paradox. *Social Psychological and Personality Science*, v. 4, n. 3, p. 340–347, 2013. Disponível em: <http://spp.sagepub.com/content/4/3/340.abstract>.

BRASIL. *Constituição da República Federativa do Brasil de 1988 - Art.5º Inc X*. Abril 2014. Disponível em: <http://presrepublica.jusbrasil.com.br/legislacao/91972/constituicao-da-republica-federativa-do-brasil-1988#art-5–inc-X>.

BRASIL. *Lei nº 12.527, de 18 de novembro de 2011*. Abril 2014. Disponível em: <http://www.acessoainformacao.gov.br/acessoainformacaogov/acesso-informacao-brasil/legislacao-integra-completa.asp#8>.

CHEN, L.; HOANG, D. Novel data protection model in healthcare cloud. In: *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on*. [S.l.: s.n.], 2011. p. 550 –555.

CHENG, Y. et al. Privacy in machine-to-machine communications a state-of-the-art survey. In: *Communication Systems (ICCS), 2012 IEEE International Conference on*. [S.l.: s.n.], 2012. p. 75–79. ISSN Pending.

DARWISH, A.; ZARKA, A.; ALOUL, F. Towards understanding phishing victims' profile. In: *Computer Systems and Industrial Informatics (ICCSII), 2012 International Conference on*. [S.l.: s.n.], 2012. p. 1–5.

FISCHER-HÜBNER, S. *IT-security and privacy: design and use of privacy-enhancing security mechanisms*. Berlin, Heidelberg: Springer-Verlag, 2001. ISBN 3-540-42142-4.

GHANI, N. A.; SIDEK, Z. M. Controlling your personal information disclosure. In: *Proceedings of the 7th WSEAS International Conference on Information Security and Privacy*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2008. (ISP'08), p. 23–27. ISBN 978-960-474-048-2. Disponível em: <http://dl.acm.org/citation.cfm?id=1576645.1576649>.

GLOSSARY, I. S. *Internet Security Glossary*. Maio 2000. Disponível em: <http://www.ietf.org/rfc/rfc2828.txt>.

GOMES, H. S. *Governo finaliza projeto que regula a exploração de dados pessoais*. Abril 2014. Disponível em: <http://www1.folha.uol.com.br/tec/2013/02/1224492-governo-finaliza-projeto-que-regula-a-exploracao-de-dados-pessoais.shtml>.

GOWRIGOLLA, B.; SIVAJI, S.; MASILLAMANI, M. Design and auditing of cloud computing security. In: *Information and Automation for Sustainability (ICIAFs), 2010 5th International Conference on*. [S.l.: s.n.], 2010. p. 292 –297.

GRANDISON, T.; THORPE, S.; STENNETH, L. Simultaneously supporting privacy and auditing in cloud computing systems. In: *Services (SERVICES), 203 IEEE Ninth World Congress on*. [S.l.: s.n.], 2013. p. 290–297.

GUNASEKERA, S. *Android Apps Security*. [S.l.]: Apress, 2012.

HADNAGY, C. *Social Engineering - The Art of Human Hacking*. Indianapolis, Indiana, USA: Wiley, 2011.

HAMLEN, K. et al. Identity management for cloud computing: developments and directions. In: *Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research*. New York, NY, USA: ACM, 2011. (CSIIRW '11), p. 32:1–32:1. ISBN 978-1-4503-0945-5. Disponível em: <http://doi.acm.org/10.1145/2179298.2179333>.

HANNE, M. et al. Analysis of vulnerability to facebook users. In: *Proceedings of the 18th Brazilian symposium on Multimedia and the web*. New York, NY, USA: ACM, 2012. (WebMedia '12), p. 335–342. ISBN 978-1-4503-1706-1. Disponível em: <http://doi.acm.org/10.1145/2382636.2382707>.

HILL, R. et al. Introducing cloud computing. In: *Guide to Cloud Computing*. Springer London, 2013, (Computer Communications and Networks). p. 3–19. ISBN 978-1-4471-4602-5. Disponível em: <http:dx.doi.org10.10079781447146032_1>.

HONG, J. *How Social Engineering Works*. Janeiro 2013. Disponível em: <http://blog.wombatsecurity.com/howsocialengineeringworks/>.

HOULIHAN, R.; DU, X. An effective auditing scheme for cloud computing. In: *Global Communications Conference (GLOBECOM), 2012 IEEE*. [S.l.: s.n.], 2012. p. 1599–1604. ISSN 1930-529X.

HUMPHREYS, L. Who's watching whom? a study of interactive technology and surveillance. *Journal of Communication*, Blackwell Publishing Ltd, v. 61, n. 4, p. 575–595, 2011. ISSN 1460-2466. Disponível em: <http://dx.doi.org/10.1111/j.1460-2466.2011.01570.x>.

ISHITANI, L. *Uma Arquitetura para Controle de Privacidade na Web*. Tese (Doutorado) — Universidade Federal de Minas Gerais, Dezembro 2003.

IVES, B.; WALSH, K. R.; SCHNEIDER, H. The domino effect of password reuse. *Commun. ACM*, ACM, New York, NY, USA, v. 47, n. 4, p. 75–78, abr. 2004. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/975817.975820>.

JMETER. *JMeter*. Dezembro 2013. Disponível em: <http://jmeter.apache.org/>.

JORDÃO, R. *Acesso à informação pública: Uma introdução à Lei n° 12.527, de 18 de novembro de 2011*. Brasil: Imprensa Nacional, 2011.

JOSANG, A.; POPE, S. User centric identity management. In: . [S.l.: s.n.], 2005. (AusCERT'05).

KOCH, H. S. Computer auditing and control. In: *Proceedings of the 1979 annual conference*. New York, NY, USA: ACM, 1979. (ACM '79), p. 188–. ISBN 0-89791-008-7. Disponível em: <http://doi.acm.org/10.1145/800177.810063>.

KOWTKO, M. Securing our nation and protecting privacy. In: *Systems, Applications and Technology Conference (LISAT), 2011 IEEE Long Island*. [S.l.: s.n.], 2011. p. 1–6.

MASSONET, P. et al. A monitoring and audit logging architecture for data location compliance in federated cloud infrastructures. In: *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*. [S.l.: s.n.], 2011. p. 1510 –1517. ISSN 1530-2075.

MAYER, J.; MITCHELL, J. Third-party web tracking: Policy and technology. In: *Security and Privacy (SP), 2012 IEEE Symposium on*. [S.l.: s.n.], 2012. p. 413–427. ISSN 1081-6011.

MILITÃO, E. *Câmara e Senado exigem CPF para revelar salário de servidor*. Janeiro 2012. Disponível em: <http://congressoemfoco.uol.com.br/noticias/camaraesenadoexigemcpfpara-revelarsalariodeservidor/>.

MOLYNEAUX, I. *The Art of Application Performance Testing*. [S.l.]: O'Reilly, 2009.

OLIVEIRA, R. R. de. Mestrado, *Avaliação de manutenibilidade entre as abordagens de web services RESTful e SOAP-WSDL*. São Carlos, SP, BR: [s.n.], Junho 2012.

PAUTASSO, C.; ZIMMERMANN, O.; LEYMANN, F. Restful web services vs. "big"' web services: making the right architectural decision. In: *Proceedings of the 17th international conference on World Wide Web*. New York, NY, USA: ACM, 2008. (WWW '08), p. 805–814. ISBN 978-1-60558-085-2. Disponível em: <http://doi.acm.org/10.1145/1367497.1367606>.

PEARSON, S. Taking account of privacy when designing cloud computing services. In: *Software Engineering Challenges of Cloud Computing, 2009. CLOUD '09. ICSE Workshop on*. [S.l.: s.n.], 2009. p. 44–52.

PEARSON, S. et al. Accountability for cloud and other future internet services. In: *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*. [S.l.: s.n.], 2012. p. 629–632.

PFITZMANN, A.; HANSEN, M. *A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management*. 2009.

PLANALTO.GOV.BR. *Lei n. 12.527, de 18 de Novembro de 2011*. Janeiro 2012. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm>.

PORTALTRANSPARENCIA.GOV.BR. *Sobre o Portal*. Abril 2014. Disponível em: <http://www.portaltransparencia.gov.br/faleConosco/perguntas-tema-sobre-o-portal.asp>.

POWELL, G. *Beginning XML Databases*. [S.l.]: Wiley Publishing Inc., 2007.

PRIVACILLA. *Privacy fundamentals*. 2003.

SAKURAGUI, R. R. M. Doutorado, *Gerenciamento de identidades com privacidade do usuário em ambiente web*. São Carlos, SP, BR: [s.n.], Janeiro 2012.

SCHIFFMAN, J. et al. Cloud verifier: Verifiable auditing service for iaas clouds. In: *Services (SERVICES), 203 IEEE Ninth World Congress on*. [S.l.: s.n.], 2013. p. 239–246.

SHAIKH, F.; HAIDER, S. Security threats in cloud computing. In: *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*. [S.l.: s.n.], 2011. p. 214–219.

SHINDER, D. L.; CROSS, M. *Scene of the Cybercrime*. Burlington, MA: Elsevier, 2008.

SMITH, R.; XU, J. A survey of personal privacy protection in public service mashups. In: *Service Oriented System Engineering (SOSE), 2011 IEEE 6th International Symposium on*. [S.l.: s.n.], 2011. p. 214–224.

SOBIREY, S. F.-H. M.; RANNENBERG, K. *Pseudonymous Audit for Privacy Enhanced Intrusion Detection*. 1997.

SPENCER, T. Identity in the cloud. *Computer Fraud & Security*, v. 2012, n. 7, p. 19 – 20, 2012. ISSN 1361-3723. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1361372312700751>.

SPRING, J. Monitoring cloud computing by layer, part 2. *Security Privacy, IEEE*, v. 9, n. 3, p. 52 –55, may-june 2011. ISSN 1540-7993.

SRINIVASAN, M. K. et al. State-of-the-art cloud computing security taxonomies: a classification of security challenges in the present cloud computing environment. In: *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*. New York, NY, USA: ACM, 2012. (ICACCI '12), p. 470–476. ISBN 978-1-4503-1196-0. Disponível em: <http://doi.acm.org/10.1145/2345396.2345474>.

STOLFO, S.; SALEM, M.; KEROMYTIS, A. Fog computing: Mitigating insider data theft attacks in the cloud. In: *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*. [S.l.: s.n.], 2012. p. 125 –128.

SUMTER, L. Cloud computing: security risk. In: *Proceedings of the 48th Annual Southeast Regional Conference*. New York, NY, USA: ACM, 2010. (ACM SE '10), p. 112:1–112:4. ISBN 978-1-4503-0064-3. Disponível em: <http://doi.acm.org/10.1145/1900008.1900152>.

SUNDARESWARAN, S. et al. Promoting distributed accountability in the cloud. In: *Cloud Computing CLOUD, 2011 IEEE International Conference on*. [S.l.: s.n.], 2011. p. 113 –120. ISSN 2159-6182.

TAN, Y. S. et al. Tracking of data leaving the cloud. In: *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*. [S.l.: s.n.], 2012. p. 137 –144.

TAVANI, H.; GRODZINSKY, F. Cyberstalking, personal privacy, and moral responsibility. *Ethics and Information Technology*, Kluwer Academic Publishers, v. 4, p. 123–132, 2002. ISSN 1388-1957. Disponível em: <http://dx.doi.org/10.1023/A3A1019927824326>.

VELTE, T.; VELTE, A.; ELSENPETER, R. *Cloud Computing, A Practical Approach*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 2010. ISBN 0071626948, 9780071626941.

W3C. *Web services architecture*. Junho 2012. Disponível em: <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>.

WALTERS, R. The cloud challenge: realising the benefits without increasing risk. *Computer Fraud & Security*, v. 2012, n. 8, p. 5 – 12, 2012. ISSN 1361-3723. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1361372312700829>.

WANG, H.; LEE, M. K. O.; WANG, C. Consumer privacy concerns about internet marketing. *Commun. ACM*, ACM, New York, NY, USA, v. 41, n. 3, p. 63–70, mar. 1998. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/272287.272299>.

WANG, J. et al. Providing privacy preserving in cloud computing. In: *Human System Interactions (HSI), 2010 3rd Conference on*. [S.l.: s.n.], 2010. p. 472–475.

WARREN, S. D.; BRANDEIS, L. D. *The right to privacy*. [S.l.]: Harvard Law Review, 1890.

WESTIN, A. F. Social and political dimensions of privacy. *Journal of Social Issues*, Blackwell Publishing, v. 59, n. 2, p. 431–453, 2003. ISSN 1540-4560. Disponível em: <http://dx.doi.org/10.1111/1540-4560.00072>.

ZARGARI, S. A.; SMITH, A. Policing as a service in the cloud. In: *Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on*. [S.l.: s.n.], 2013. p. 589–596.

ZHOU, M. et al. Security and privacy in cloud computing: A survey. In: *Semantics Knowledge and Grid (SKG), 2010 Sixth International Conference on*. [S.l.: s.n.], 2010. p. 105–112.