DANIEL RICARDO DOS SANTOS

GERENCIAMENTO DE IDENTIDADES E PRIVACIDADE EM AMBIENTES DE COMPUTAÇÃO EM NUVEM

Florianópolis

Julho de 2011

DANIEL RICARDO DOS SANTOS

GERENCIAMENTO DE IDENTIDADES E PRIVACIDADE EM AMBIENTES DE COMPUTAÇÃO EM NUVEM

Monografia apresentada na Universidade Federal de Santa Catarina para obtenção do título de Bacharel em Ciências da Computação.

Orientadora:

Profa. Dra. Carla Merkle Westphall

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

Florianópolis

Julho de 2011

Título: Gerenciamento de Identidades e Privacidade em ambientes de Computação em Nuvem

Autor: Daniel Ricardo dos Santos

Banca Examinadora:

Profa. Dra. Carla Merkle Westphall
Universidade Federal de Santa Catarina - UFSC

Prof. Dr. Carlos Becker Westphall
Universidade Federal de Santa Catarina - UFSC

Marcos Aurélio Pedroso Leandro
Universidade Federal de Santa Catarina - UFSC

Shirlei Aparecida de Chaves
Universidade Federal de Santa Catarina - UFSC

Quis custodiet ipsos custodes?

(Decimus Iunius Iuvenalis)

SUMÁRIO

RESUMO

Com o crescimento da computação em nuvem e a tendência de migração de serviços para esse novo paradigma, torna-se necessário investigar questões de segurança que possam comprometer seu uso.

O gerenciamento de identidades é um campo da segurança da informação que se preocupa com o gerenciamento de usuários e seus dados, envolvendo autenticação, autorização e liberação de atributos.

Uma das questões mais preocupantes quando se envolvem dados de usuários é a privacidade na coleta, manipulação, armazenamento e destruição desses dados.

Este trabalho apresenta uma proposta de aplicação de gerenciamento de identidades com proteção à privacidade dos usuários implementada em um ambiente de computação em nuvem.

Palavras-chave: Identidade digital, Gerenciamento de Identidades, Privacidade, Computação em Nuvem

ABSTRACT

Due to the continued growth in the use of cloud computing and the tendency to migrate services to this new paradigm, it becomes necessary to investigate security issues that might compromise its use.

Identity Managament is an area in information security that is concerned with the management of users and their data, involving authentication, authorization and attribute release.

One of the biggest issues when users' data are involved is privacy in the collection, manipulation, storage and destruction of these data.

This paper presents a proposal for an identity management application with users' privacy protection implemented in a cloud computing environment.

Keywords: Digital identity, Identity Management, Privacy, Cloud Computing

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

AA	*Attribute Authority*,	p. 36
ACS	*Assertion Consumer Service*,	p. 36
AMI	*Amazon Machine Image*,	p. 39
API	*Application Programming Interface*,	p. 44
AR	*Attribute Requester*,	p. 36
AWS	Amazon Web Services,	p. 39
DNS	*Domain Name System*,	p. 39
EBS	*Elastic Block Storage*,	p. 35
EC2	*Amazon Elastic Compute Cloud*,	p. 26
FIPs	*Fair Information Practice Principles*,	p. 21
HS	*Handle Service*,	p. 36
HTTP	*Hypertext Transfer Protocol*,	p. 29
HTTPS	*Hypertext Transfer Protocol Secure*,	p. 44
IaaS	*Infrastructure as a Service*,	p. 27
IdP	Provedor de Identidades,	p. 15
IMS	*Identity Management Systems*,	p. 13
IP	*Internet Protocol*,	p. 13
JDBC	*Java Database Connectivity*,	p. 44
LDAP	*Lightweight Directory Access Protocol*,	p. 40
MAC	*Media Access Control*,	p. 21
OASIS	*Organization for the Advancement of Structured Information Standards*,	p. 17
P3P	*Platform for Privacy Preferences Project*,	p. 23
PaaS	*Platform as a Service*,	p. 27
PII	*Personally Identifiable Information*,	p. 20
RDP	*Remote Desktop Protocol*,	p. 39
RFID	*Radio-frequency identification*,	p. 21
RM	*Resource Manager*,	p. 36

SaaS	*Software as a Service*,	p. 27
SAML	*Security Assertion Markup Language*,	p. 17
SLA	*Service Level Agreement*,	p. 29
SP	Provedor de Serviços,	p. 15
SQL	*Structured Query Language*,	p. 30
SSL	*Secure Sockets Layer*,	p. 39
SSO	*Single Sign-on*,	p. 16
W3C	*World Wide Web Consortium*,	p. 23
WAYF	*Where are you from?*,	p. 36
XML	*Extensible Markup Language*,	p. 17
XSS	*Cross-site scripting*,	p. 30

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Computação em nuvem é uma nova forma de se pensar sobre a venda e a utilização de recursos computacionais. Apoiada em conceitos como virtualização, compartilhamento de recursos e escalabilidade em massa a computação em nuvem entrega esses recursos através da internet e com transparência para os usuários.

Atualmente a migração de serviços para as nuvens computacionais é uma tendência crescente e a computação em nuvem já é considerada a nova revolução no mercado de Tecnologia da Informação. As nuvens apresentam diversas vantagens para usuários e empresas. Ainda existem, porém, certas questões a serem abordadas nesse período de transição e dentre essas questões a segurança é um fator importante para garantir o sucesso e uso correto de ambientes de nuvem. Com o advento das nuvens um dos tópicos que ganha destaque nas discussões é a proteção à privacidade, especialmente porque dados sensíveis agora passam a ficar sob a custódia de terceiros.

O gerenciamento de identidades cresce em importância conforme crescem os serviços que precisam utilizar autenticação e controle de acesso de ususários. Esse é o caso de muitos serviços que rodam em nuvens e precisam estabelecer a identidade de seus usuários ao mesmo tempo que protegem sua privacidade.

O problema apresentado é a falta de respeito à privacidade em sistemas de gerenciamento de identidades e a solução proposta é a implantação de uma estrutura de gerenciamento de identidades utilizando um provedor de serviços implementado com o *software* Shibboleth, combinado com o *plugin* de privacidade uApprove em uma máquina virtual sendo executada no ambiente de nuvem provido pela Amazon.

1.2 OBJETIVO GERAL

O objetivo geral deste trabalho é inserir o gerenciamento de identidades com proteção à privacidade em ambientes de computação em nuvem.

1.3 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho que podem ser listados são:

- Descrever o que são identidades digitais e como é feito seu gerenciamento;
- Demonstrar que a privacidade é um conceito importante ao se tratar de identidades;
- Introduzir o paradigma de computação em nuvem e relatar alguns desafios de segurança;
- Implantar um provedor de identidades em uma máquina na nuvem;
- Garantir um aspecto de privacidade nesse provedor de identidades;
- Demonstrar alguns casos de uso da aplicação.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está dividido em sete capítulos: (1) Introdução, (2) Identidade e Gerenciamento de Identidades, (3) Privacidade, (4) Computação em Nuvem, (5) Desafios do Gerenciamento de Identidades na Computação em Nuvem, (6) Desenvolvimento prático e (7) Conclusões.

O Capítulo 1 descreve uma visão geral do trabalho, apresentando a motivação e os objetivos, além de apresentar a divisão do texto como um todo.

O Capítulo 2 apresenta os conceitos básicos de identidades digitais e do gerenciamento de identidades, além de mostrar exemplos de algumas tecnologias e ferramentas atualmente utilizadas nesta área, como Shibboleth e OpenID.

O Capítulo 3 apresenta os conceitos básicos da privacidade no mundo digital, além de descrever os requisitos legais e técnicos para a implantação da privacidade.

O Capítulo 4 apresenta uma visão geral da computação em nuvem, como se organiza e quais tipos de serviço pode oferecer.

O Capítulo 5 introduz e detalha os desafios do gerenciamento de identidades em ambientes de computação em nuvem.

O Capítulo 6 detalha o desenvolvimento prático do trabalho, apresentando a proposta, a instalação das ferramentas e os resultados.

O Capítulo 7 apresenta as conclusões obtidas do desenvolvimento do trabalho e possibilidades de trabalhos futuros na área.

2 IDENTIDADE E GERENCIAMENTO DE IDENTIDADES

Neste capítulo define-se o que é uma identidade digital, o que são *Identity Management Systems* (IMS) e como funcionam. Também são discutidos modelos de gerenciamento de identidade e citados exemplos de IMS.

2.1 CONCEITOS BÁSICOS

Com o advento e crescimento do uso da internet e serviços *online*, especialmente negócios e saúde, a questão da identidade digital se torna cada vez mais importante. É necessário que ambos usuários e prestadores de serviços estabeleçam suas identidades e tenham confiança mútua para que possam trocar informações.

De acordo com (WINDLEY, 2003) identidades digitais são coleções de dados que representam atributos, preferências e traços de uma entidade. Os atributos são características associadas a essa entidade, como histórico de compras, acessos a serviços e outras. Preferências representam os desejos dessa entidade, enquanto os traços são características permanentes como a data de nascimento.

Uma identidade digital representa uma entidade qualquer, seja uma pessoa, uma empresa ou qualquer outra organização e, tecnicamente, é relacionada a um identificador, que pode ser, entre outros, um nome de usuário, um número de identificação ou um endereço *Internet Protocol* (IP).

Identidades digitais funcionam como um meio de autenticação e autorização e são utilizadas em diversos cenários, entre os quais pode-se destacar: comércio eletrônico, governo eletrônico, redes sociais, *e-mail*, finanças e saúde *online*.

No contexto social é normal uma pessoa apresentar várias identidades diferentes dependendo do ambiente em que se encontra. Num banco, por exemplo, pode-se apresentar suas informações financeiras e num consultório médico suas informações médicas, mas seria incomum liberar informações financeiras para um médico e informações médicas para um banco. No contexto digital a mesma ideia se aplica, por isso uma entidade pode ter diversas ”identidades parciais”que são utilizadas em diferentes ocasiões.

2.1.1 CICLO DE VIDA DE UMA IDENTIDADE

Uma identidade digital normalmente passa pelo seguinte ciclo de vida:

Fornecimento O processo de registro do usuário e criação de sua identidade relacionada, pode incluir uma prova de identidade e atribuição de privilégios. É o momento em que o usuário fornece suas informações pessoais e essas são relacionadas a identidade sendo criada.

Propagação, uso e manutenção Fase em que a identidade criada se encontra na maior parte do tempo. Essa identidade deve ser propagada para que possa ser utilizada nos sistemas em que for solicitada e deve ser mantida com um armazenamento seguro no IMS até que sua remoção seja solicitada.

Destruição Nesse momento a identidade é destruída e a partir de então não poderá mais ser utilizada. É importante que não só a identidade, mas também contas relacionadas, atributos e informações atreladas, bem como privilégios sejam todos removidos.

Auditoria Pode ser necessária durante qualquer momento no ciclo de vida de uma identidade, por isso é importante que sejam gerados *logs* de toda operação realizada com ou sobre a identidade.

2.2 SISTEMAS DE GERENCIAMENTO DE IDENTIDADES

Com o crescimento do uso de identidades em inúmeros serviços distribuídos pela Internet cresce também a necessidade de gerenciamento dessas identidades, principalmente por questões de segurança, mas também para a redução de custos (GROß, 2003).

Além disso, muitas vezes as informações associadas a uma identidade são sensíveis, como no caso de dados médicos e financeiros. Existe uma classe especial de informações privadas que são conhecidas como *Personally Identifiable Informations* (PII), definidas como informações que podem identificar unicamente uma entidade ou que, em conjunto com outras informações podem ser usadas para essa identificação única. O conceito de PII é melhor descrito na Seção 3.1. As PII devem ser protegidas acima de tudo pois representam as informações mais confidenciais de uma entidade.

Por causa da natureza privada e sensível dessas informações associadas os sistemas de gerenciamento de identidades devem ser seguros e confiáveis.

Segundo (LEE; JEUN; JUNG, 2009) um serviço de gerenciamento de identidades pode ser definido como ”o processo de criação, gerenciamento e utilização de identidades de usuários e a infraestrutura que suporta esse processo.”

Os seguintes papéis são desempenhados num sistema de gerenciamento de identidades:

Usuário É a entidade que possui uma identidade e utiliza os serviços tanto do provedor de identidades quanto do provedor de serviços.

Provedor de Identidades (IdP) É aquele que fornece os serviços de gerenciamento de identidades, necessários para que o usuário utilize o provedor de serviços.

Provedor de Serviços (SP) É aquele que fornece os serviços que o usuário efetivamente deseja utilizar, por exemplo, *e-mail*, *e-commerce* e outros. O provedor de serviços delega a autenticação e autorização dos usuários que acessam seus serviços a um IdP e, por isso, o provedor de serviço é conhecido como parte confiante.

É possível que o provedor de serviços seja também o provedor de identidades, ou que ambos façam parte do mesmo domínio, mas isso não é necessário e é detalhado na seção 2.3.

2.2.1 REQUISITOS DE UM IMS

Os requisitos mais importantes de um IMS são funcionalidade, segurança, privacidade e interoperabilidade.

Na parte de funcionalidade cabe ao sistema fazer aquilo a que se propõe, ou seja, gerenciar a criação, uso, manutenção e destruição de identidades e informações relacionadas a essas identidades, permitindo ao usuário que interfira quando necessário ou desejado. Deve, também, oferecer as mesmas funções para identidades parciais, que possam ser utilizadas em diferentes contextos.

Quanto à segurança é importante que o sistema atenda aos requisitos de disponibilidade, integridade e confidencialidade. É de extrema importância que o sistema opere respeitando esses requisitos, devido à importância dos serviços providos e da confidencialidade das informações manipuladas.

O requisito de privacidade, que é um dos focos deste trabalho é detalhado no capítulo 3, mas é importante que os usuários tenham acesso e controle sobre suas informações, que o sistema recolha e armazene seguramente essas informações e que apenas o conjunto mínimo de informações sobre uma entidade seja armazenado.

Em relação à interoperabilidade é importante que o sistema seja compatível com outros semelhantes disponíveis, respeite padronizações e implemente interfaces compatíveis com essas padronizações.

2.3 MODELOS DE GERENCIAMENTO DE IDENTIDADES

Além de fornecer maior segurança na manipulação de identidades a principal vantagem do uso de um IMS é a possibilidade de *Single Sign-on* (SSO). SSO é um acesso único que um usuário faz em um serviço e, a partir daí, está autenticado para utilizar outros serviços no mesmo domínio.

Existe também o conceito de *Single Sign-off*, que é a ação reversa ao *Single Sign-on*, ou seja, a possibilidade de encerrar todas as sessões de acesso a diferentes serviço a partir de um acesso único.

(LEE; JEUN; JUNG, 2009) define os seguintes modelos de sistemas de gerenciamento de identidade:

Silo O modelo mais comum e mais fácil de ser implementado. O próprio provedor de serviços gerencia as identidades utilizadas e que são válidas apenas no seu domínio de serviço. Nesse caso o usuário deve ter uma identidade diferente para cada provedor de serviços;

Centralizado Nesse modelo um único provedor de identidades gerencia identidades que podem ser usadas em diferentes serviços providos num mesmo domínio. É o modo mais simples de se implementar o SSO;

Federado No modelo federado o IdP compartilha as identidades entre provedores de serviço incluídos num círculo de confiança. Esse círculo é montado através de acordos prévios entre os provedores de serviços. Esse modelo também provê SSO, com a vantagem de estender esse serviço para organizações em diferentes domínios;

Centrado no usuário Nesse último modelo o próprio usuário gerencia as políticas de uso de suas identidades e informações, bem como controla a criação, uso e remoção dessas mesmas informações.

Todos os modelos apresentados enfrentam problemas de segurança. No primeiro ocorre a replicação de dados em provedores diferentes e o uso de diferentes identidades. Ao usar diferentes identidades é comum que os usuários precisem anotar diferentes senhas ou criar senhas mais fáceis de lembrar e, geralmente, mais fracas.

Enquanto isso, nos modelos que proveem SSO existe a possibilidade de um atacante obter ou falsificar uma única identidade e assim conseguir acesso a vários serviços.

O foco deste trabalho é o modelo federado. De acordo com (BENANTAR, 2005) uma federação se manifesta pelos mecanismos utilizados para permitir que uma organização participante forneça serviços para entidades registradas em outras organizações que sejam membros da mesma federação.

O modelo federado apresenta algumas vantagens em relação aos outros, como a extensão do uso de SSO para além das fronteiras de uma única organização. Além disso esse modelo apresenta as vantagens de um modelo de gerenciamento distribuído (WINDLEY, 2005).

2.4 FERRAMENTAS E PADRÕES

Atualmente existem diversas ferramentas e padrões de gerenciamento de identidades disponíveis no mercado. Entre essas pode-se destacar algumas: SAML, Shibboleth, OpenID e Windows CardSpace. Apesar de um pouco antigo, um bom estudo comparativo entre sistemas de gerenciamento de identidades pode ser encontrado em (EU, 2003).

2.4.1 SAML

A *Security Assertion Markup Language* (SAML) é uma linguagem de marcação baseada em *Extensible Markup Language* (XML) para troca de informações de segurança e identidade entre diferentes domínios. É um padrão definido pela *Organization for the Advancement of Structured Information Standards* (OASIS) (OASIS, 2010).

SAML permite que os membros de uma federação criem e transmitam asserções sobre entidades, que são usadas para permitir o acesso a diferentes serviços. Foi projetado tendo em vista segurança e extensibilidade e é reconhecido como uma das melhores soluções para federação de identidades.

Entre as possibilidades do uso de SAML estão o estabelecimento de transações de alto nível entre participantes, autenticação iniciada pelo provedor de identidade ou pelo provedor de serviços, autorização baseada em atributos, *Single Sign-off* e administração centralizada de identidades.

É utilizada em diversas ferramentas como Shibboleth da Internet2 e InfoCard da Microsoft entre outros.

2.4.2 SHIBBOLETH

Segundo (INTERNET2, 2010a) o Shibboleth é um *software* de código aberto que permite *Single Sign-On* pela *web* dentro de ou entre organizações.

Além do SSO, o Shibboleth permite o compartilhamento de atributos das identidades e autorização de acesso federada. Para esses propósitos o Shibboleth utiliza vários padrões, especialmente o SAML.

É utilizado principalmente em universidades na Europa e nos Estados Unidos, mas existe também um projeto de federação brasileira utilizando Shibboleth, a federação CAFe (RNP, 2010).

Nas universidades é bastante usado para prover serviços restritos, como acesso à livros digitais e cursos à distância para alunos e professores de diferentes instituições que façam parte da mesma federação.

Entre as federações mais conhecidas que utilizam Shibboleth pode-se destacar: InCommon nos Estados Unidos, UK Federation no Reino Unido e SURFnet na Holanda (INTERNET2, 2010b).

2.4.3 OPENID

O OpenID é outro protocolo de federação de identidades bastante utilizado e que tem crescido muito nos últimos anos. Atualmente existem mais de 1 bilhão de identificadores de usuários e mais de 50000 *websites* aceitando OpenID como opção de *login* (OpenID Foundation, 2010)

Não se trata de um *software* específico ou mesmo uma biblioteca, mas sim de especificações sobre as interações e informações necessárias para se utilizar SSO. O OpenID oferece um conjunto de regras para provedores de serviço e de identidade, mas não força a utilização de ferramentas específicas.

Alguns exemplos de empresas que emitem ou aceitam OpenIDs em seus *websites* são: Google, Facebook, Yahoo, Microsoft e Sun (OpenID Foundation, 2010).

2.4.4 WINDOWS CARDSPACE

Windows CardSpace é um componente da plataforma .Net para gerenciamento de identidades baseada em cartões pessoais, de acordo com o padrão InfoCards da Microsoft.

Permite a federação de identidades centrada no usuário e não estabelece conexões entre o provedor de serviços e o provedor de identidades, deixando essa tarefa para um seletor de identidades. O seletor de identidades é um programa cliente que pode ser utilizado pelo usuário para conter suas informações e utiliza cartões de identidade que podem ser auto-assinados ou administrados. Cartões auto-assinados são gerados pelo próprio usuário, enquanto cartões administrados são providenciados por um provedor de identidades e permitem que o CardSpace se conecte a eles (MICROSOFT, 2011).

3 PRIVACIDADE

Neste capítulo define-se o que é privacidade e como o conceito se relaciona com o gerenciamento de identidades, quais são os princípios da privacidade e quais o requisitos legais e técnicos que devem ser cumpridos para a garantia da privacidade.

3.1 CONCEITOS BÁSICOS

A privacidade é um direito humano fundamental, presente no artigo 12 da Declaração Universal dos Direitos Humanos (ONU, 1948). A privacidade pode ser encarada de diversas formas, de acordo com o contexto em que se apresenta.

Segundo (GOLDBERG; WAGNER; BREWER, 1997) privacidade relaciona-se com a capacidade de um indivíduo proteger informações sobre si. Nesse contexto, anonimidade é a privacidade da identidade.

Como privacidade é um termo muito amplo, faz sentido separá-lo em diferentes categorias, o que é feito em (SHOSTACK; SYVERSON, 2003):

Inobservabilidade Quando o indivíduo não pode ser observado.

Não rastreamento Quando não se pode rastrear um mesmo indivíduo entre várias identidades.

Autodeterminação de informação Quando um indivíduo sabe que a informação que ele fornece será usada de maneiras que ele conhece e aprova.

Anonimidade Quando não há nenhum identificador do indivíduo.

Neste trabalho tratamos da privacidade em relação à informações pessoais. Em (PEARSON, 2009) essas informações são separadas em cinco tipos:

***Personally Identifiable Information* (PII)** qualquer informação que pode ser usada para identificar ou localizar uma entidade, ou informação que relacionada a outras pode levar a identificação de uma entidade. Exemplos: nome, endereço, endereço IP

Informações sensíveis Informações sobre saúde, finanças, religião e outras que são consideradas privadas.

Dados de uso Dados coletados a partir do uso de dispositivos como computadores e impressoras ou dados comportamentais como padrões de acesso a informações, histórico do *browser* e outros.

Identificadores de dispositivos Outras informações que podem ser rastreadas para um dispositivo específico como etiquetas *Radio-frequency identification* (RFID) e endereços *Media Access Control* (MAC).

3.2 PRINCÍPIOS DE PRIVACIDADE

O *Fair Information Practice Principles* (FIPs) é um conjunto de regras básicas para manipulação de informações com proteção à privacidade criado pela Comissão de Comércio Americana (*Federal Trade Comission*) que regula o uso de informações privadas nos Estados Unidos e serve de base para regras de todo o mundo (Federal Trade Comission, 2010).

Os FIPs consistem em cinco princípios básicos, que são descritos a seguir:

ADVERTÊNCIA/CONSCIÊNCIA

O princípio mais fundamental. Os usuários devem ser avisados das práticas de uma entidade que irá manipular informações sensíveis antes que qualquer informação seja coletada. Se o usuário não tem consciência de quais informações são liberadas e como serão utilizadas ele não tem como decidir sobre a liberação dessas informações. A maioria dos outros princípios de privacidade só faz sentido quando esse está presente.

Alguns dos itens mais importantes que devem ser informados ao usuário são:

- Identificação da entidade coletando os dados;
- Identificação de como esses dados serão utilizados;
- Identificação de quem possivelmente receberá esses dados;
- Se a coleta dos dados é obrigatória ou opcional;
- Como a entidade vai garantir a confidencialidade, integridade e qualidade dos dados.

Uma boa prática desse princípio na internet requer que essas informações sejam dadas aos usuários de maneira clara e colocadas de forma visível e de fácil acesso a partir de qualquer página em que se colete informações do usuário, além de ser inevitável que o usuário seja informado quando estiver liberando informações.

ESCOLHA/CONSENTIMENTO

Escolha significa dar aos usuários opções sobre como quaisquer dados coletados serão utilizados. A escolha se refere principalmente aos usos secundários dos dados coletados. Usos secundários são aqueles além do necessário para a utilização de um serviço, por exemplo, inserir o usuário numa lista de *e-mails* ou a transferência de informações para parceiros.

Geralmente a escolha se dá por meio de uma decisão de sim ou não para a liberação de informações para um uso específico. Na internet isso pode ser alcançado através do uso simples de caixas de seleção que permitam ao usuário decidir se deseja liberar suas informações sendo coletadas.

ACESSO/PARTICIPAÇÃO

O acesso consiste na possibilidade de um usuário acessar as informações coletadas sobre si e poder alterar esses valores caso encontrem-se errados ou incompletos. Para um uso eficiente deve ser possível ao usuário acessar as informações de maneira rápida e fácil e contestar a validade das mesmas.

INTEGRIDADE/SEGURANÇA

A integridade é alcançada quando as entidades que coletam dados o fazem a partir de fontes confiáveis, provendo acesso dos usuários a esses dados e destruindo os que estejam incorretos.

A segurança se refere a medidas técnicas e administrativas de proteção quanto à perda ou acesso, uso, destruição ou liberação não autorizados dos dados.

CUMPRIMENTO/REPARAÇÃO

Os princípios de proteção a privacidade são efetivos quando há um mecanismo que os faça serem cumpridos, senão tratam-se apenas de conselhos e não precisam ser respeitados. Existem duas formas principais de garantir o respeito aos princípios de privacidade: a auto-regulação da indústria e a legislação.

3.3 MECANISMOS DE REGULAMENTAÇÃO

3.3.1 LEGISLAÇÃO

Em diversos países existem leis específicas de proteção à privacidade de consumidores e usuários de serviços que coletam informações pessoais. Entre as principais, pode-se destacar: *Canadian Personal Information Protections and Electronic Documents Act* no Canadá, *European Data Protection Directive* na União Europeia e *Gramm-Leach-Bliley Act* nos Estados Unidos. Essas leis ajudam a garantir que as entidades que lidam com informações pessoais mantenham uma política de privacidade e cumpram-na. No Brasil, a privacidade é uma garantia constitucional, mas não existe uma lei específica sobre a proteção de dados. Isso levou à criação, por membros do Ministério da Justiça e da FGV de um anteprojeto de lei, que atualmente encontra-se em debate. O projeto propõe inclusive a criação de um Conselho Nacional de Proteção de Dados Pessoais (CULTURADIGITAL, 2011).

3.3.2 POLÍTICAS DE PRIVACIDADE

Uma política de privacidade é um documento que expressa a forma como uma entidade coleta, utiliza, administra e libera informações de seus usuários. O conteúdo específico de uma política de privacidade depende de quais exigências ela precisa atender.

As políticas são a principal fonte de informação de um usuário que deseja utilizar um serviço que exija a liberação de informações. Com o avanço da regulamentação em relação à privacidade os serviços devem sempre adequar suas políticas à novas regras.

Existem iniciativas que preveem formas automatizadas de declaração de políticas de privacidade e aceitação por parte dos usuários, como o *Platform for Privacy Preferences Project* (P3P) da *World Wide Web Consortium* (W3C)(W3C, 2011)

3.3.3 SELOS DE PRIVACIDADE

Programas de certificação *online* ou selos de privacidade são uma forma de auto-regulação da indústria quanto aos princípios de privacidade. Normalmente um programa de certificação exige que um serviço ou empresa atenda à determinadas exigências quanto à obtenção e ao tratamento dado as informações pessoais e é comum que haja um monitoramento constante quanto a continuação do cumprimento às regras. Alguns exemplos de selos de privacidade são: TRUSTe, o primeiro a ser criado e o maior programa de certificação ativo, com mais de 3000 *sites* certificados (TRUSTE, 2010), BBBOnline e EuroPriSe. Quando um serviço está

em conformidade com as exigências definidas pelo programa de certificação, pode ostentar um certificado de aprovação.

3.4 AMEAÇAS À PRIVACIDADE

Atualmente a preocupação com a proteção à privacidade é crescente, como pode ser notado em diversas manifestações de acadêmicos e pessoas ligadas à indústria. Entre essas pode-se destacar a carta enviada ao CEO da Google Eric Schmidt que expressa preocupações quanto a política de segurança dos serviços em nuvem da empresa e é assinada por 38 acadêmicos e pesquisadores. (APPELBAUM et al., 2010)

Alguns exemplos mais conhecidos de vazamento de informações pessoais são dados em (BRAGHIN; CREMONINI; ARDAGNA, 2009), que descreve como esses incidentes ocorrem nos mais diversos tipos de instituições, como organizações governamentais, instituições de ensino e empresas privadas.

Uma lista bastante extensa e atualizada de casos de vazamento de informações pode ser encontrada em (Open Security Foundation, 2010)

4 COMPUTAÇÃO EM NUVEM

Neste capítulo define-se o que é computação em nuvem e quais seus modelos de funcionamento. Também são dados exemplos de serviços de computação em nuvem.

4.1 CONCEITOS BÁSICOS

Computação em nuvem é o conceito de entregar recursos computacionais compartilhados, sejam de armazenamento, processamento ou mesmo de *software* para usuários através da Internet. O nome é derivado dos diagramas de rede onde a Internet é usualmente representada como uma nuvem. Uma analogia recorrente é que com a computação em nuvem os recursos computacionais são entregues aos usuários pela internet como a eletricidade é entregue pela rede elétrica.

A definição de (MATHER; KUMARASWAMY; LATIF, 2009) de computação em nuvem é baseada em cinco atributos:

Recursos compartilhados Os recursos de computação são compartilhados entre diversos usuários em vários níveis, como infraestrutura ou aplicação.

Escalabilidade em massa Proporciona a possibilidade de escalar dezenas de milhares de sistemas, além de ser possível escalar taxa de transmissão e armazenamento.

Elasticidade Os usuários podem aumentar ou diminuir seus recursos computacionais de acordo com a necessidade.

Pagamento por demanda Os usuários pagam apenas pelos recursos e pelo tempo efetivamente utilizados.

Auto-provisionamento Os próprios usuários alocam seus recursos.

A computação em nuvem é uma nova mudança de paradigma na forma como os usuários acessam e utilizam recursos computacionais, assim como houve a mudança dos mainframes para o modelo cliente/servidor.

As principais vantagens da computação em nuvem são a redução de custos e a facilidade de adaptação, que ocorrem em grande parte pelo uso de recursos compartilhados. Entretanto, alguns problemas também decorrem desse uso, entre eles o principal sendo a preocupação com a segurança.

No modelo de computação em nuvem os dados, ou mesmo os aplicativos que manipulam esses dados, não se encontram mais armazenados com o usuário ou a empresa que utiliza os serviços, mas sim em *datacenters*, muitas vezes de localização desconhecida, operados pelas empresas fornecedoras de serviços.

Existem quatro modelos de nuvem que podem ser aplicados em uma empresa que deseja utilizar esse tipo de serviço:

Nuvem pública Ocorre quando uma empresa terceirizada fornece serviços de computação em nuvem para vários clientes. Esse é o tipo mais comum de nuvem e como exemplos podem ser citados o *Amazon Elastic Compute Cloud* (EC2) (AMAZON, 2010) e o Salesforce.com (SALESFORCE, 2010). Nesse tipo de modelo os usuários têm pouco controle sobre as questões de segurança da nuvem, já que a empresa responsável opera toda a estrutura necessária para o funcionamento do serviço.

Nuvem privada No segundo modelo uma empresa monta uma estrutura de nuvem a partir da sua rede privada e com o objetivo de utilizar essa nuvem internamente. Esse modelo de nuvem não traz as maiores vantagens do modelo mais usual, como a grande redução de custos e menor preocupação com manutenção, mas em contrapartida também ameniza as desvantagens do modelo anterior. Já que a estrutura da nuvem fica sob a responsabilidade da própria empresa, ela mesmo pode garantir a sua segurança e confiabilidade.

Nuvem comunitária Nesse modelo várias organizações que têm interesses e requisitos em comum juntam a sua infra-estrutura para formar uma única nuvem. As principais vantagens são que os custos são divididos entre todas as organizações que participam da nuvem comunitária e as preocupações com segurança e privacidade são menores do que em uma nuvem pública.

Nuvem híbrida O quarto modelo é uma mistura dos três modelos anteriores. É possível que uma empresa utilize uma nuvem pública para serviços não essenciais e também mantenha, por questões de segurança, uma nuvem privada para serviços essenciais.

4.2 TIPOS DE SERVIÇOS DE COMPUTAÇÃO EM NUVEM

Três tipos diferentes de serviços são mencionados quando se considera computação em nuvem: *Software as a Service* (SaaS), *Platform as a Service* (PaaS) e *Infrastructure as a Service* (IaaS).

4.2.1 SAAS

Tradicionalmente quando uma empresa deseja utilizar um novo *software*, essa empresa compra uma licença de uso desse novo *software* e então o carrega no seu próprio hardware e, se desejar, pode adquirir um serviço de suporte. No modelo SaaS a empresa assina um serviço de uso do *software* que funciona como um aluguel, tanto do *software* como de toda a estrutura necessária para executá-lo, podendo acessá-lo a partir de diferentes dispositivos.

Um exemplo de SaaS é o Google Apps, que entrega aos usuários aplicativos de escritório, como e-mail e agenda, a partir de uma assinatura anual de US$50,00 por conta de usuário (GOOGLE, 2010).

4.2.2 PAAS

No modelo PaaS o vendedor do serviço oferece aos clientes uma plataforma de desenvolvimento de aplicativos, que o usuário utiliza tanto no desenvolvimento quanto na posterior disponibilização do serviço *online*. É comum que junto com a plataforma de desenvolvimento o vendedor forneça partes comuns de código já prontas para integração com o aplicativo do cliente.

Alguns exemplos de PaaS são o Windows Azure e o Google AppEngine. O Azure oferece uma plataforma para desenvolvimento em linguagens como PHP, Ruby e Java, além de bancos de dados e a possibilidade de executar essas aplicações nos *datacenters* da Microsoft (MICROSOFT, 2010)

4.2.3 IAAS

No caso do IaaS o que o cliente procura é a própria infra-estrutra de computação, em forma de poder de processamento, capacidade de armazenamento e taxa de transmissão. Nesse tipo de serviço geralmente o cliente tem controle total sobre a máquina fornecida através de acesso remoto.

Um exemplo de IaaS é o Amazon Elastic Compute Cloud (EC2), onde é possível alugar instâncias de máquinas virtuais com diferentes configurações, focadas em processamento, memória ou armazenamento.

Com a popularização dos serviços de nuvem, muitas novas definições de tipos de serviços têm surgido. Um exemplo relevante a esse trabalho é o IDaaS, um serviço de gerenciamento de identidades para a nuvem, definido em (Cloud Security Alliance, 2010).

5 DESAFIOS DO GERENCIAMENTO DE IDENTIDADES NA COMPUTAÇÃO EM NUVEM

Como descreve (HUANG; ZHANG; HOU, 2009) atualmente assistimos a migração de muitas aplicações *desktop* para a nuvem, onde os dados de perfis e contexto dos usuários devem ser armazenados baseados em identidades digitais.

O uso da computação em nuvem traz novos desafios à segurança da informação e, mais especificamente, ao gerenciamento de identidades. Nas próximas seções alguns desses desafios são descritos.

O *Service Level Agreement* (SLA) é uma parte do contrato de serviços entre um usuário e um provedor de serviços de nuvens, onde o serviço a ser prestado fica formalmente definido. Nesse documento deveriam ficar explícitas as questões de segurança e privacidade. No entanto, essas questões geralmente são difíceis de quantificar. Além disso é importante que os termos do contrato sejam monitorados constantemente, devido à natureza dinâmica das nuvens (TAKABI; JOSHI; AHN, 2010)

5.1 DESAFIOS DE SEGURANÇA

Os serviços de computação em nuvem se utilizam de diversas tecnologias para serem providos. Entre essas pode-se destacar: *Web services*, virtualização e criptografia.

Além disso, esses serviços apresentam características específicas, como definido no capítulo anterior. O uso dessas tecnologias aliado a essas caraterísticas traz diversas vulnerabilidades e desafios de segurança, entre os quais pode-se destacar (GROBAUER; WALLOSCHEK; STOCKER, 2011):

Desvio de Sessão (*Session Hijacking*) O *Hypertext Transfer Protocol* (HTTP) não tem a noção de estado, no entanto algumas aplicações *web* necessitam do uso de sessões, que podem ser implementadas de diversas formas. Muitas dessas implementações são vulneráveis à ataques de desvio de sessão, onde um atacante consegue se apoderar de uma sessão válida para obter acesso não autorizado;

Fuga da virtualização A possibilidade de que um atacante consiga escapar de um ambiente virtualizado e obter acesso à máquina física é inerente à própria virtualização e extremamente perigosa para *clouds*;

Quebra de criptografia Conforme são descobertos avanços em criptoanálise, protocolos e algoritmos criptográficos antes considerados seguros passam a ser obsoletos e vulneráveis à ataques;

Acesso não autorizado à interface de gerência Devido às características de auto-atendimento, o gerenciamento de *clouds* geralmente é feito através de uma interface *web*, sendo mais fácil para um atacante obter acesso não autorizado do que se a interface de gerência estivesse disponível apenas para alguns administradores;

Vulnerabilidades do IP O acesso às *clouds* é geralmente feito via internet, uma rede não confiável, e utilizando protocolos comuns, como o IP, que possuem problemas de segurança;

Vulnerabilidades de recuperação de dados Devido a relocação de recursos é possível que um atacante tenha acesso à recursos de armazenamento e memória previamente utilizados por um usuário, que ainda podem conter dados sensíveis.

Os ataques mais comuns a vulnerabilidades em ambientes *web* são ataques de injeção:

Injeção de SQL Quando um atacante consegue inserir código *Structured Query Language* (SQL) malicioso a ser executado no banco de dados;

Injeção de comandos Quando um atacante consegue inserir comandos a serem executados no sistema operacional;

***Cross-site scripting* (XSS)** Quando um atacante consegue inserir um código JavaScript que será executado pelo navegador da vítima.

Para as outras vulnerabilidades descritas anteriormente existem ataques mais complexos que podem ser realizados. Em (RISTENPART et al., 2009) é descrito um ataque interessante quanto à virtualização, onde é possível um atacante descobrir a provável localização física de uma máquina virtual e instanciar máquinas virtuais co-residentes na mesma máquina física, possibilitando a extração de informações.

Além desses desafios de segurança, existem desafios específicos do gerenciamento de identidades e de privacidade em ambientes de computação em nuvem.

5.2 DESAFIOS DO GERENCIAMENTO DE IDENTIDADES

Como já mencionado, o fato de armazenar dados na nuvem gera uma certa insegurança que é justificada pela falta de controle do usuário sobre seus próprios dados e identidades são peças importantes de informação, pois a partir de uma identidade é possível descobrir outras informações sobre uma entidade.

Quando não se utiliza um IMS, conforme cresce o número de serviços disponíveis é normal que cresça também o número de contas que um mesmo usuário deve gerenciar, o que pode levar à problemas de segurança. De acordo com (ZARANDIOON; YAO; GANAPATHY, 2009) quando um usuário tem que gerenciar várias contas ele acaba usando senhas fracas ou anotando essas senhas em algum lugar.

Os principais desafios do gerenciamento de identidades na computação em nuvem passam pela autenticação e pelo controle de acesso de usuários. Autenticação é o processo de validar as credenciais fornecidas por um usuário. As credenciais fornecidas podem ser, por exemplo, *login* e senha ou um certificado digital.

Alguns desafios da autenticação em ambientes de nuvem são (Cloud Security Alliance, 2010):

Proteção As credenciais devem ser protegidas, seja durante a comunicação, utilizando protocolos seguros, ou durante o armazenamento, utilizando criptografia. Também devem ser protegidas de ataques de força bruta ou ataques à mecanismos como reinicialização das senhas;

Personificação Se um usuário utiliza as mesmas credenciais para vários serviços, quando um atacante obtem essas credenciais ele pode utilizar esses diversos serviços personificando o usuário;

Phishing Atacantes podem convencer os usuários a liberarem as suas credenciais de forma voluntária, sem ter conhecimento do ataque;

Regras Devem ser definidas regras para credenciais fortes, como tamanho e complexidade de senhas, força da chave de certificados digitais e armazenamento apenas do *hash* de senhas.

Um desafio importante da autenticação em ambientes de nuvem é manter a interoperabilidade, já que o uso de diferentes mecanismos e protocolos de autenticação pode levar a falta de compatibilidade entre serviços. Por isso, é importante utilizar um sistema de gerenciamento de identidades que implemente padrões estabelecidos, como o SAML.

Com a autenticação realizada o foco passa para a autorização e o controle de acesso. O maior desafio do controle de acesso em computação em nuvem é a diversidade de serviços que devem ser protegidos. Um mecanismo eficiente de controle de acesso deve lidar com os requisitos de inúmeros tipos de serviços diferentes.

Alguns desafios do controle de acesso em ambientes de nuvem são (Cloud Security Alliance, 2010):

- Controlar o acesso aos serviços baseado em políticas especificadas pelos usuários e pelo nível de serviço contratado;
- Manter o controle de acesso dos usuários aos seus próprios dados, para evitar acessos não autorizados em ambientes compartilhados;
- Notificar aos usuários mudanças em suas permissões, para evitar mudanças não autorizadas;
- Prover *logs* de auditoria contendo informações sobre acesso e uso de serviços;
- Prover soluções para determinar a responsabilidade em caso de problemas.

A pesquisa em gerenciamento de identidades em computação em nuvem é bastante recente e a existência de iniciativas como a *Protect Network* (Protect Network, 2011) e a conferência *Cloud Identity Summit* (Cloud Identity Summit, 2011) indicam que tanto indústria quanto academia investem no tema e geram resultados interessantes.

5.3 DESAFIOS DE PRIVACIDADE

A privacidade é um dos aspectos que mais chama a atenção quando se pensa em segurança na nuvem. Segundo (Marcon Jr. et al., 2010) alguns aspectos que podem ser levantados quando se pesquisa privacidade em ambientes de nuvem são:

Acesso O usuário deve ter o direito de saber quais informações suas estão mantidas na nuvem e solicitar a remoção dessas informações;

Aderência Os provedores de serviços de nuvem precisam seguir leis, normas e regulamentos quando lidam com informações privadas, além de manterem os compromissos estabelecidos no SLA. Como os serviços de nuvem geralmente atravessam diversas jurisdições, uma questão importante é qual é o foro qualificado para julgar eventuais problemas;

Armazenamento Os provedores precisam saber onde e como os dados privados são armazenados e de que forma podem ser transmitidos. Os usuários devem ter garantias de que seus dados são armazenados e transferidos de forma segura;

Retenção Os provedores devem manter políticas que tratem da retenção de dados na nuvem. Deve haver informações acerca de por quanto tempo os dados podem ser retidos e como é feita sua remoção.

Destruição Assim que solicitado pelo usuário, ou que o tempo de retenção tenha expirado, ou por qualquer outro motivo cabível o provedor deve poder destruir os dados que estavam armazenados e garantir que isso seja feito de forma segura e em pouco tempo. Os provedores também devem garantir que não há cópias dos dados armazenados em outros locais após sua destruição;

Auditoria Devem existir formas de monitorar e garantir que os provedores estão cumprindo os requisitos de privacidade. É importante que se mantenham *logs* de acesso a dados;

Violação da privacidade Caso haja um caso de violação de privacidade ou vazamento de informações deve-se saber quem é o culpado, quem é responsável por notificar o evento e por controlá-lo.

No caso de um provedor de identidades na nuvem é importante também que ambos o provedor de identidades e o provedor de serviços de nuvem utilizados tenham uma política de privacidade clara e disponível aos usuários.

6 DESENVOLVIMENTO PRÁTICO

Neste capítulo é descrito o desenvolvimento prático do trabalho, desde a elaboração de uma proposta de aplicação até a obtenção dos resultados, detalhando o processo de instalação e as ferramentas utilizadas.

6.1 PROPOSTA

No capítulo anterior foram vistos alguns desafios do gerenciamento de identidades na computação em nuvem, sendo a privacidade citada como um dos principais. Esse trabalho tem como foco a privacidade dos usuários de serviços na nuvem.

A proposta do trabalho é a utilização de uma aplicação que englobe um provedor de identidades com uma camada de proteção à privacidade executando num ambiente de nuvem. Uma visão geral da proposta pode ser visualizada na Figura 1 e é descrita logo a seguir:

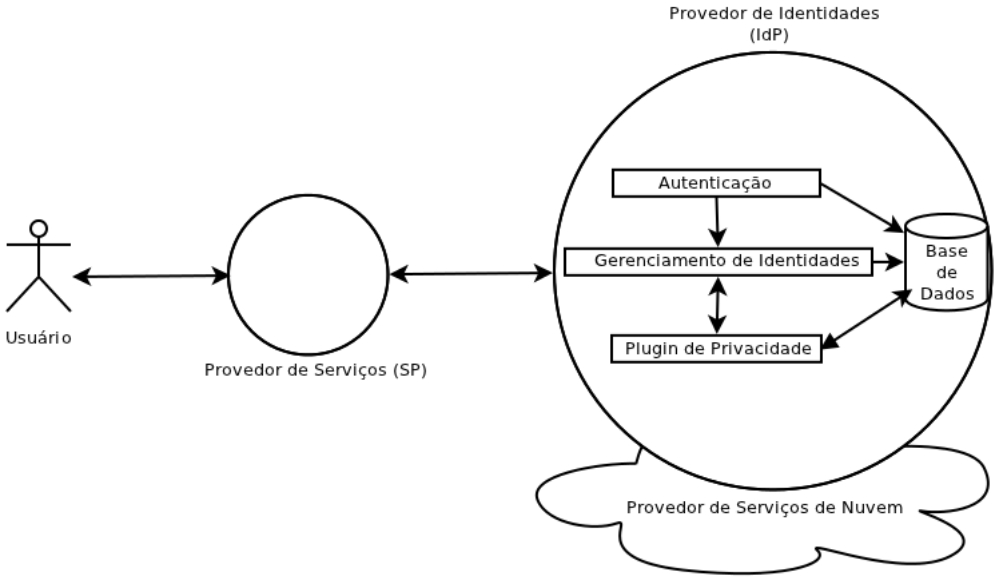

Figura 1: Diagrama geral da proposta

No cenário desta proposta o usuário interessado em utilizar um serviço protegido acessa diretamente o provedor de serviços, o provedor de serviços então o redireciona para seu respectivo provedor de identidades, que pode ser informado pelo usuário e deve ter a confiança do

provedor de serviços. O provedor de identidades está executando em um ambiente de nuvem, o que é transparente para o usuário. O provedor de identidades pede a autenticação do usuário e acessa seus atributos em sua base de dados. Quando o usuário está autenticado e antes de ser novamente redirecionado para o provedor de serviços seus dados passam por um *plugin* de privacidade, onde o usuário fica ciente e deve consentir com a liberação de seus atributos.

6.2 FERRAMENTAS UTILIZADAS

6.2.1 AMAZON EC2

O EC2 foi o provedor de serviços de nuvem utilizado no trabalho. O EC2 provê uma Infraestrutura como um Serviço, em que é possível instanciar máquinas virtuais a partir de imagens pré-definidas. Existem opções de imagens disponíveis de diversos sistemas operacionais e com diferentes aplicações instaladas, mas também é possível o usuário registrar suas próprias imagens.

Além das imagens é possível configurar características da máquina em que a imagem estará sendo executada. Essas características incluem capacidade de processamento, memória e armazenamento.

No EC2 o usuário pode atribuir endereços IP estáticos às máquinas instanciadas e configurar a liberação de portas acesso. A persistência dos dados é feita utilizando-se volumes *Elastic Block Storage* (EBS), que agem como discos rígidos das máquinas. É possível criar *snapshots* desses volumes, para recuperação de dados.

Além disso, o serviço é totalmente configurável através de uma interface *web* e o usuário paga pelo tempo efetivamente utilizado de cada instância. O serviço é compatível com o Eucalyptus, tornando possível a posterior migração de imagens para outros ambientes, incluindo uma nuvem privada.

6.2.2 SHIBBOLETH

Entre os diversos sistemas de gerenciamento de identidades disponíveis, optou-se pelo Shibboleth devido à sua popularidade em ambientes acadêmicos e boa documentação, além de ser um *software* de código aberto.

O Shibboleth é formado por duas partes principais: o IdP e o SP, que se encontram separados, mas se comunicam para prover o acesso seguro aos serviços.

O fluxo de funcionamento do Shibboleth é representado na Figura 2 e descrito a seguir:

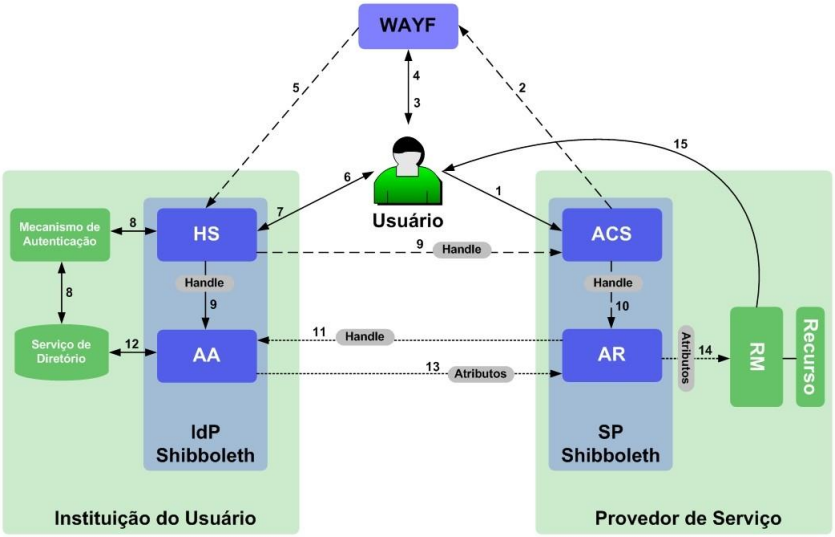

Figura 2: Funcionamento do Shibboleth. (CORDOVA, 2006)

No Passo 1 o usuário navega para o provedor de serviços, onde deseja acessar um recurso protegido. Nos Passos 2 e 3 o Shibboleth redireciona o usuário para a página *Where are you from?* (WAYF), onde ele deve informar qual o seu provedor de identidades. No Passo 4 o usuário informa seu IdP e no Passo 5 ele é redirecionado para o *site*, que é o componente *Handle Service* (HS) do seu IdP. Nos Passos 6 e 7 o usuário informa seus dados e no Passo 8 o componente HS verifica a validade dos seus dados. O HS cria um *handle* para identificar o usuário e registra-o no *Attribute Authority* (AA). No Passo 9 esse *handle* confirma a autenticação do usuário. O *handle* é verificado pelo *Assertion Consumer Service* (ACS) e transferido para o *Attribute Requester* (AR) e no Passo 10 é criada uma sessão. No Passo 11 o AR utiliza o *handle* para requisitar os atributos do usuário ao IdP. No passo 12 o IdP verifica se pode liberar os atributos e no Passo 13 o AA responde com os valores dos atributos. No Passo 14 o SP recebe os atributos e os passa para o *Resource Manager* (RM), que no Passo 15 carrega o recurso solicitado para o usuário (CORDOVA, 2006).

6.2.3 UAPPROVE

Depois de escolhido o sistema de gerenciamento de identidades foi necessário definir quais características de privacidade deveriam ser respeitadas e como atingir esse objetivo. Como apresentado no capítulo 3, os princípios mais importantes da privacidade são a notificação aos usuários e a possibilidade de escolha. Uma ferramenta que implementa esses dois princípios é o uApprove.

O uApprove é um *plugin* de privacidade para o Shibboleth desenvolvido pela rede de universidades suíças SWITCH, para uso em sua federação acadêmica, a SWITCHaai. Atualmente

encontra-se na versão 2.2.1, que foi utilizada neste trabalho.

De acordo com (SWITCH, 2011) o uApprove serve aos seguintes propósitos:

1. Para o usuário implementa um mecanismo que o informa sobre a liberação de atributos para um provedor de serviços;

2. Para o administrador de um provedor de identidades, provê uma ferramenta para implementar leis de proteção à privacidade exigindo o consentimento dos usuários antes da liberação dos atributos e permite coletar informações sobre essa liberação e acessos a um provedor de serviços.

O objetivo do uApprove é garantir que o usuário saiba quais dados seus são liberados e para quem são liberados, além disso o usuário deve concordar com os termos de uso do provedor de serviços. Essas são algumas das características essenciais de privacidade vistas no Capítulo 3.

O uApprove é dividido em três componentes principais: *IdP plugin*, *Viewer* e *Reset approvals*, que são descritos na sequência.

IDP PLUGIN

O componente mais básico do uAppove, é um filtro do Shibboleth, que testa se a ferramenta deve obter o consentimento do usuário para a liberação de seus atributos.

A Figura 3 mostra o fluxo de execução do *plugin* para decidir se o *Viewer* deve ser invocado:

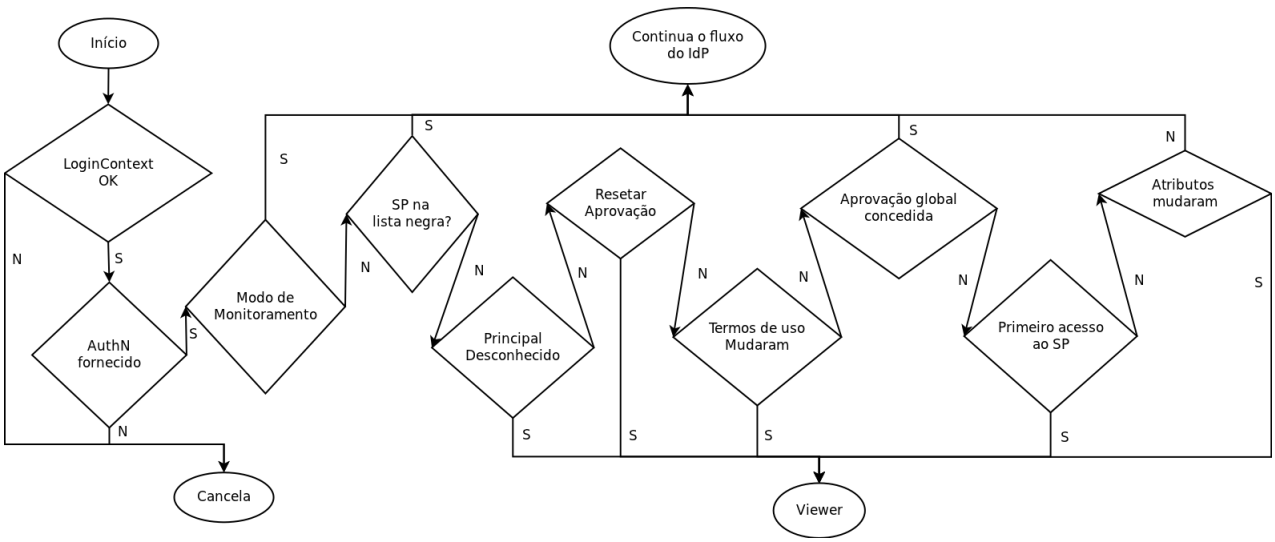

Figura 3: Fluxograma de execução do uApprove. Adaptado de: (SWITCH, 2011)

Primeiramente o *plugin* verifica se o *Login Context* está correto. *LoginContext* é um objeto Java criado quando uma autenticação é requisitada. Caso o *LoginContext* esteja correto é verificado se o AuthN foi fornecido. AuthN é o *Shibboleth Authentication Request*, uma mensagem enviada pelo SP para o IdP para iniciar uma sessão. Se alguma dessas verificações for negativa a execução é cancelada e o processo de autenticação terminado.

Caso as duas primeiras verificações sejam positivas o *plugin* verifica se está executando em modo de monitoramento. Nesse modo ele só monitora os atributos que serão liberados, sem exibir nada para o usuário e continua o fluxo do IdP. Caso esteja em modo de monitoramento o fluxo segue para o IdP, em caso negativo o *plugin* seu fluxo, verificando se o SP se encontra na lista negra. Essa lista na verdade é uma lista branca, ou seja, uma lista de SPs nos quais o uApprove deve assumir automaticamente o consentimento do usuário.

Se o SP se encontrar na lista o fluxo segue para o IdP, se não o *plugin* verifica se o *Principal* é conhecido. *Principal* é o identificador único de um usuário. Se o *Principal* for desconhecido, ou seja, o usuário nunca utilizou o *plugin*, o *Viewer* será invocado.

Se o *Principal* já for conhecido é verificado se o usuário reiniciou seus consentimentos. Caso tenha reiniciado, o Viewer será invocado, caso não tenha o *plugin* continua seu fluxo. A próxima verificação é se os termos de uso foram alterados desde o último acesso do usuário, caso tenham sido o fluxo segue para o *Viewer*, em caso negativo o *plugin* segue.

Após isso é verificado se o usuário concedeu aprovação global para a liberação de seus atributos. Em caso afirmativo o fluxo segue para o IdP, em caso negativo segue para a próxima verificação. Então verifica-se se o usuário está acessando o SP pela primeira vez, em caso afirmativo o *Viewer* é invocado, em caso negativo é feita última verificação.

A última verificação refere-se aos atributos sendo requisitados pelo SP. Se eles tiverem sido alterados o *Viewer* é invocado, se não o fluxo segue para o IdP.

Em todos os casos em que o fluxo for para o IdP a execução do plugin é ignorada pelo usuário. Em todos os casos em que o *Viewer* for invocado, o usuário deve interagir e fornecer seu consentimento.

VIEWER

É o componente principal da ferramenta. Trata-se de um *servlet* que apresenta ao usuário uma página *web* com os termos de uso que o usuário deve aceitar quando utiliza o provedor de identidades. Essa página só é exibida se o usuário estiver utilizando o IdP pela primeira vez ou se os termos de uso tiverem sido alterados. Nessa página o usuário deve marcar uma

opção de aceitação dos termos para continuar. Em seguida é apresentada ao usuário uma tabela com o nome e valor dos atributos que serão liberados para o provedor de serviços. Existe a possibilidade de concordar automaticamente em liberar os atributos, para que o usuário não seja novamente questionado. Caso não concorde com a liberação de seus atributos o usuário será redirecionado para uma página informando que ele não pode acessar o provedor de serviços.

RESET APPROVALS

O componente administrativo da ferramenta, que atualmente é bastante simples. É opcional e permite que o usuário reinicie as liberações que já foram concedidas.

6.3 INSTALAÇÃO DAS FERRAMENTAS

6.3.1 INSTALAÇÃO DA INFRA-ESTRUTRA BÁSICA

A instalação da aplicação começou com a infra-estrutura básica que seria utilizada e o primeiro passo foi definir o serviço de nuvem em que ele estaria disponível. A aplicação deveria utilizar uma Infraestrutura como um Serviço (IaaS) e ter algumas características essenciais como possibilidade de persistência dos dados e possibilidade de utilização de IPs estáticos, além da possibilidade de liberação de portas no *firewall*.

Optou-se então pelo uso de uma *cloud* pública. O serviço escolhido foi o EC2 da Amazon Web Services (AWS). Foi instanciada uma máquina virtual executando Windows Server 2008 (*Amazon Machine Image* (AMI) ID ami-c3e40daa) e utilizado o serviço Elastic IP para obter um IP estático para essa instância. O IP obtido foi o 50.19.108.64, com *Domain Name System* (DNS) público ec2-50-19-108-64.compute-1.amazonaws.com. Para persistência dos dados utilizou-se um volume EBS de 30GB.

As portas liberadas no *firewall* foram: 3306 para acesso ao banco de dados MySQL, 3389 para acesso remoto via *Remote Desktop Protocol* (RDP), 8009 para uso do Shibboleth e 8080 para uso do Tomcat.

Com a máquina instanciada e em execução começou-se a instalação da aplicação. Primeiramente foi instalado o servidor *web* Apache, versão 2.2 (The Apache Software Foundation, 2011a). Foi gerado um certificado digital para a máquina e o servidor Apache foi configurado para permitir o uso de *Secure Sockets Layer* (SSL) com esse certificado. O servidor foi configurado para aceitar na porta 80 conexões não-SSL e nas portas 443 e 8443 conexões SSL.

Depois foi instalado o Java Development Kit (JDK), versão 1.6.0_25 (Oracle Corporation,

2011a), e finalmente o servidor de aplicações Apache Tomcat 6.0.22 (The Apache Software Foundation, 2011b), no qual deveriam ser executadas as aplicações de autenticação, gerenciamento de identidades e o *plugin* de privacidade.

Foi então configurado um *proxy* no Apache para repassar os pedidos dessas aplicações para o Tomcat. Isso foi feito com a criação de um arquivo *httpd-mod-proxy.conf*, com o conteúdo descrito no Apêndice (A.1.1) e habilitando a carga desses módulos no servidor web, no arquivo httpd.conf (A.1.2).

O primeiro serviço a ser instalado foi o mecanismo de autenticação. O serviço escolhido foi o JASIG CAS Server (JASIG, 2011), versão 3.3.2, que realiza a autenticação de usuários através de login e senha e possibilita *Single Sign On* através de uma interface *web* e então repassa os usuários autenticados para o Shibboleth. O CAS foi configurado para procurar os usuários em um diretório *Lightweight Directory Access Protocol* (LDAP). Isso foi feito editando o arquivo *deployerConfigContext.xml* (A.1.3):

Para utilizar esse diretório foi instalado o OpenLDAP (OpenLDAP Foundation, 2011) em uma outra máquina virtual, um Ubuntu 10.10 (AMI ID ami-cef405a7) também executando na nuvem da Amazon.

6.3.2 INSTALAÇÃO DO SHIBBOLETH

Com esses passos iniciais prontos foi possível começar a instalação do provedor de identidades Shibboleth. Primeiramente a aplicação do IdP em si foi configurada e instalada no Tomcat.

O Shibboleth precisa fazer parte de uma federação para que possa ser utilizado com provedores de serviço. A federação escolhida para ser utilizada nesse trabalho foi a TestShib (INTERNET2, 2011b), criada para propósito de testes de configurações do Shibboleth, tanto de SPs quanto de IdPs. Para utilizar o TestShib foi necessário cadastrar o IdP, informando o endereço DNS e o certificado gerado anteriormente.

O Shibboleth foi então configurado para utilizar os metadados do TestShib, o que foi feito no arquivo *relying-party.xml* (A.1.4), nesse arquivo também foi configurado o caminho do certificado gerado anteriormente e utilizado pelo Shibboleth.

Para que o Shibboleth recebesse a autenticação do CAS Server foi utilizado o CAS Client. As alterações necessárias para o uso do CAS Client estão no arquivo *web.xml* do Shibboleth (A.1.5).

Com esses passos prontos o Shibboleth está corretamente instalado e autenticando usuários

a partir do CAS. No entanto o provedor de identidades ainda não liberava nenhum atributo de usuário para o provedor de serviços, o que tornaria seu uso inviável para a maioria das aplicações práticas.

Na liberação dos atributos de usuário é que reside a maior preocupação com a privacidade no gerenciamento de identidades, pois são os atributos que contem os dados sensíveis dos usuários e que devem ser tratados com cuidado na hora em que são criados, armazenados, transferidos ou destruídos.

Os atributos dos usuários seguem um esquema que é definido no diretório LDAP. O esquema mais comum é o eduPerson (INTERNET2, 2011a), que contém atributos comuns à membros de uma federação acadêmica. O esquema utilizado neste trabalho foi o brEduPerson (RNP, 2011), uma extensão do eduPerson para federações brasileiras.

Alguns exemplos de atributos presentes nos esquemas eduPerson e brEduPerson e que foram utilizados nestes trabalho são: eduPersonPrincipalName, geralmente representado pelo nome de usuário utilizado no *Single Sign On*, esse atributo é utilizado quando se necessita um identificador persistente através de vários serviços; eduPersonAffiliation, representa o tipo de afiliação de um usuário com a federação da qual faz parte e brPersonCPF, o número do CPF do usuário.

A configuração da liberação dos atributos dos usuários é feita em dois arquivos: *attribute-resolver.xml* e *attribute-filter.xml*. O primeiro define cada atributo que pode ser liberado pelo IdP e de onde ele é recuperado. O segundo define a política de liberação de atributos do provedor de identidades, tanto de maneira geral quanto de maneira específica para cada atributo.

No *attribute-resolver.xml* (A.1.6) foram criadas definições para quatro atributos que seriam utilizados como exemplo: eduPersonPrincipalName; eduPersonAffiliation; brPersonCPF, o CPF do usuário; cn, o primeiro nome do usuário e sn, o sobrenome do usuário, conforme definidos no esquema brEduPerson. Com exceção do atributo eduPersonAffiliation, que foi definido como um atributo estático, sempre retornando o valor ”member”, todos os outros são pesquisados em um diretório LDAP.

No *attribute-filter.xml* (A.1.7) foram criadas regras de liberação para cada um dos atributos. Como esse provedor de identidades só seria utilizado com o provedor de serviços do TestShib e apenas para fins de teste, sendo o objetivo principal do trabalho demonstrar o uso da camada de privacidade, os atributos podem ser liberados para qualquer provedor de serviço, o que não seria seguro em um ambiente de produção. A seguir o excerto do arquivo *attribute-filter.xml* em que as alterações foram feitas:

Com essas configurações o Shibboleth está pronto para ser usado com um provedor de serviços que requisite a liberação de atributos. O próximo passo é a instalação do plugin de privacidade.

6.3.3 INSTALAÇÃO DO UAPPROVE

Para a instalação do uApprove é necessário utilizar uma base de dados que armazene informações sobre o consentimento dos usuários e a liberação de seus atributos. Para esse armazenamento foi utilizado o sistema de gerenciamento de bancos de dados MySQL, versão 5.5 (Oracle Corporation, 2011b), instalado na mesma máquina do Shibboleth. Primeiramente foi criado um banco de dados e um usuário para o uApprove:

```
mysql -u root -p
mysql>
CREATE DATABASE uApprove;
CREATE USER 'uApprove'@'localhost' IDENTIFIED BY 'uApprove';
GRANT USAGE ON *.* TO 'uApprove'@'localhost';
GRANT SELECT , INSERT , UPDATE , DELETE ON `uApprove`.* TO 'uApprove'@'
    localhost';
ALTER DATABASE uApprove DEFAULT CHARACTER SET utf8 COLLATE
    utf8_general_ci;
```

Em seguida foram criadas as tabelas a serem utilizadas pelo *plugin*:

```
mysql -u root -p
mysql>
use uApprove;

create table ArpUser (
  idxArpUser int unsigned auto_increment primary key,
  auUserName varchar(255) not null,
  auLastTermsVersion varchar(255),
  auFirstAccess timestamp,
  auLastAccess timestamp
);
create index idxUserName on ArpUser (auUserName );

create table ShibProvider (
  idxShibProvider int unsigned auto_increment primary key,
  spProviderName varchar(255)
);

insert into ShibProvider (idxShibProvider) values (1);
```

```
create index idxProvidername on ShibProvider (spProviderName);

create table AttrReleaseApproval (
  idxAttrReleaseApproval int unsigned auto_increment primary key,
  araIdxArpUser int unsigned references ArpUser ( idxArpUser ),
  araIdxShibProvider int unsigned references ShibProvider( idxShibProvider
      ),
  araTimeStamp timestamp not null,
  araTermsVersion varchar(255),
  araAttributes text(2048)
);

create table ProviderAccess (
  idxProviderAccess int unsigned auto_increment primary key,
  paIdxArpUser int unsigned references ArpUser( idxArpUser ),
  paIdxShibProvider int unsigned references ShibProvider( idxShibProvider )
      ,
  paAttributesSent text,
  paTermsVersion varchar(255),
  paIdxAttrReleaseApproval int unsigned references AttrReleaseApproval (
      idxAttrReleaseApproval ),
  paShibHandle varchar(255),
  paTimeStamp timestamp not null
);
```

Foi então gerado um arquivo *terms-of-use.xml* que contém um exemplo de Termos de Uso. Também é necessário definir um segredo compartilhado que será utilizado para criptografar mensagens transmitidas entre o IdP plugin e o Viewer.

Com as configurações prontas é necessário criar um filtro para ativar o uso do IdP plugin com o Shibboleth, isso é feito editando novamente o arquivo *web.xml* do Shibboleth (A.1.5).

Existem algumas configurações adicionais que podem ser realizadas, como a definição de uma lista negra de SPs bloqueados, ou definir a ferramenta no modo de monitoração apenas. Além disso existem configurações que podem ser realizadas nos componentes Viewer e Reset approvals, mas nenhuma delas foi realizada por se afastarem do escopo do trabalho.

6.4 INSTALAÇÃO CONCLUÍDA

Com a instalação concluída, uma visão geral da aplicação pode ser resumida na Figura 4:

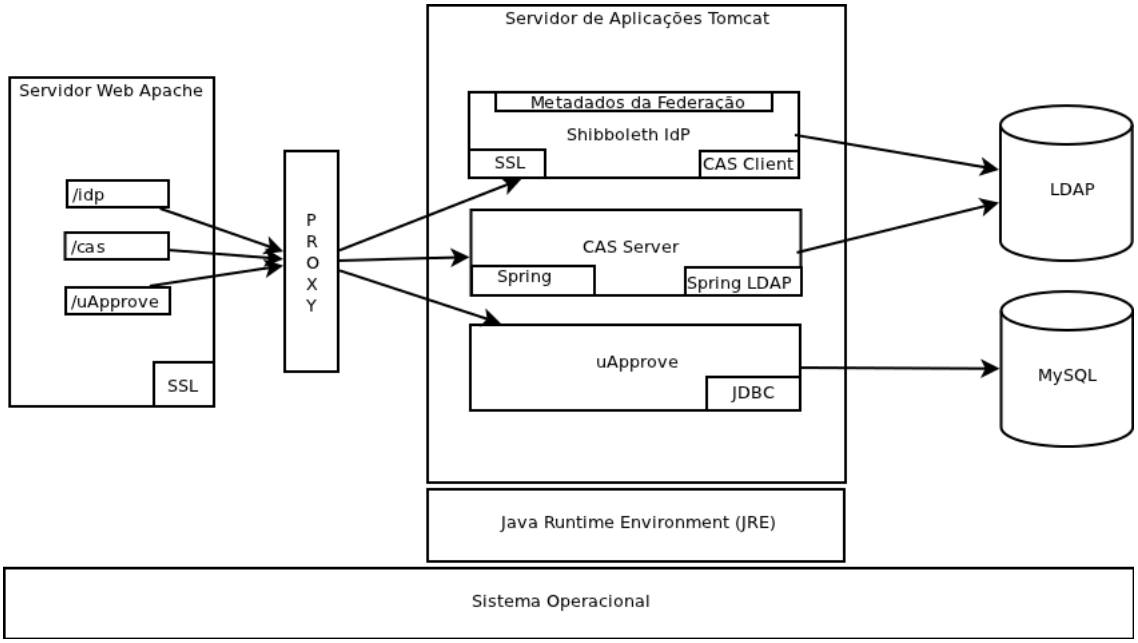

Figura 4: Visão detalhada da aplicação

A Figura 4 representa a visão detalhada da parte do IdP que havia sido descrita na Figura 1. Como ponto de acesso temos o servidor *web* apache, que recebe as requisições *Hypertext Transfer Protocol Secure* (HTTPS). O Apache tem configurado um *proxy* que encaminha essas requisições para o Tomcat, para que sejam recebidas pela aplicação correta. Dentro do Tomcat existem três aplicações sendo executadas:

Shibboleth IdP Configurado com os metadados da federação e utilizando o CAS client para receber a autenticação, é acessado através de /idp;

CAS Server Executa a autenticação e a envia para o IdP. Utiliza o *framework web* Spring e é acessado através de /cas;

uApprove Obtém o consentimento do usuário e libera os atributos para o IdP. Acessa o banco de dados através da *Application Programming Interface* (API) *Java Database Connectivity* (JDBC) e é acessado através de /uApprove.

O IdP e o CAS Server obtem os dados dos usários através de um diretório LDAP e o uApprove utiliza o banco de dados MySQL para registrar as informações de consentimento dos usuários.

6.5 RESULTADOS

Nesta seção são descritos alguns casos de uso da aplicação.

6.5.1 CASO DE USO 1

Esse caso de uso representa o primeiro acesso de um usuário a um SP.

1. Primeiramente o usuário deseja utilizar o serviço disponível em `https://sp.testshib.org/`. Ao acessar a página é pedido para que o usuário informe qual o seu provedor de identidades. Geralmente há uma lista de provedores de identidade disponíveis ao usuário, como as instituições de ensino em uma federação acadêmica. No entanto, como o TestShib mantém um cadastro com muitos provedores é pedido que o usuário escreve o endereço do provedor que deseja utilizar (Figura 5).

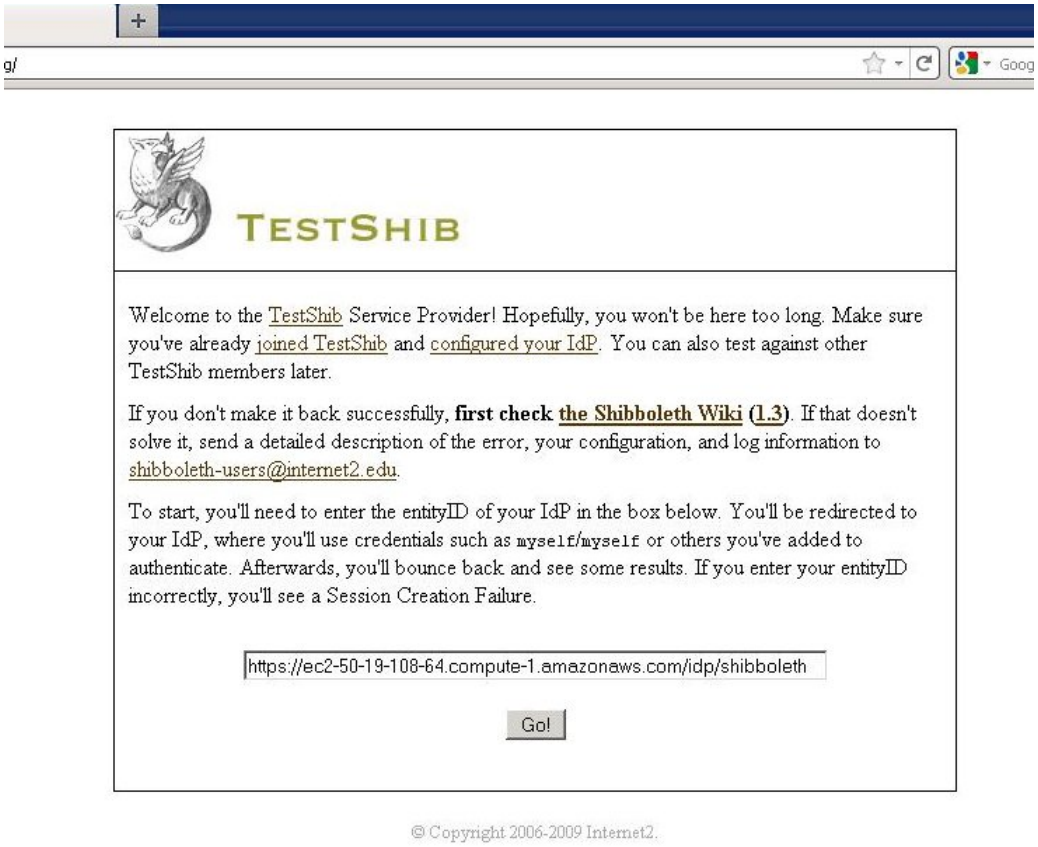

Figura 5: Página inicial do provedor de serviços

O usuário informa o provedor de serviços `https://ec2-50-19-108-64.compute-1.amazonaws.com/idp/shibboleth` e é então redirecionado para a página de autenticação, fornecida pelo mecanismo de autenticação CAS.

2. Nessa página (Figura 6) o usuário faz sua autenticação por *login* e senha, que são buscados no diretório LDAP, instalado em outra máquina (Figura 7). Nesse caso o usuário é jose e sua senha 1234.

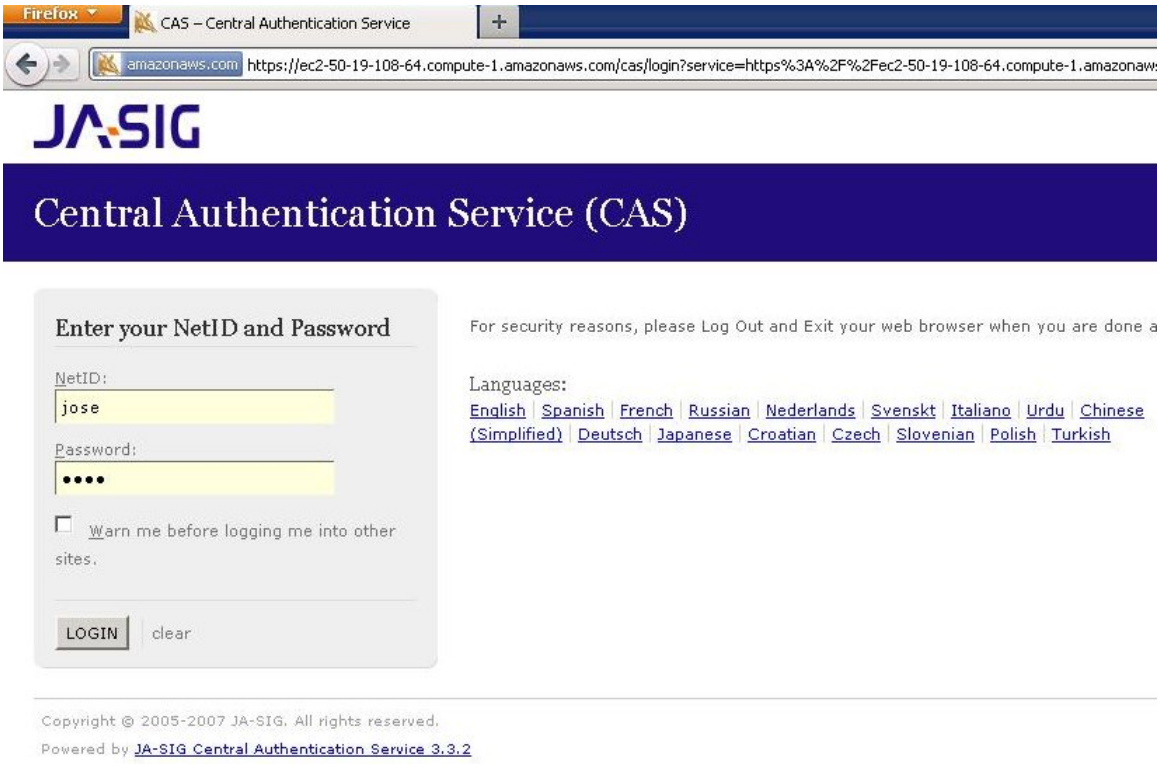

Figura 6: Página do mecanismo de autenticação

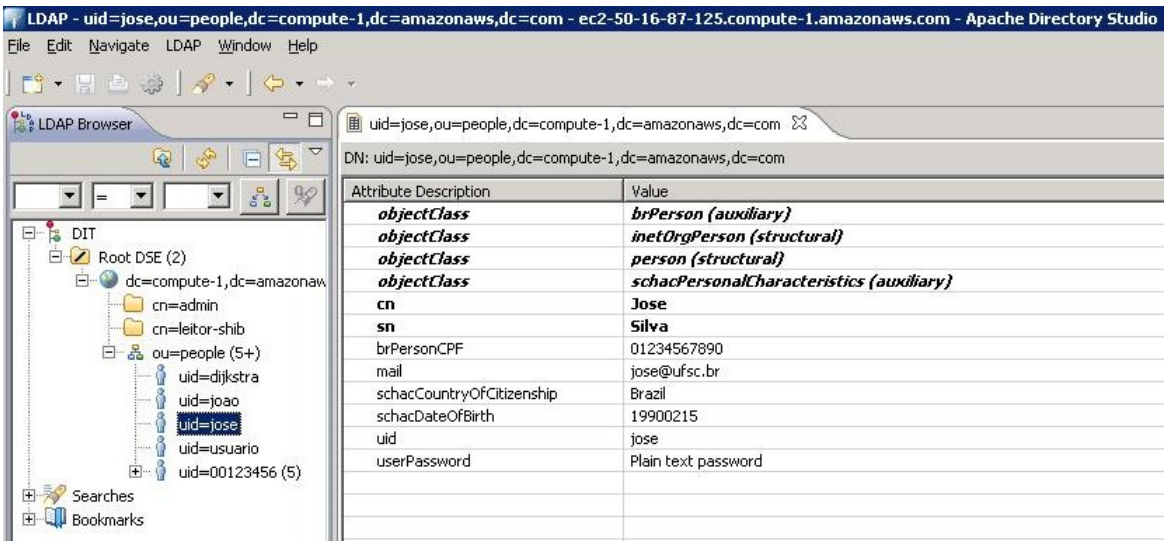

Figura 7: Usuário no diretório LDAP

3. Depois da autenticação o Shibboleth busca no diretório os atributos que devem ser liberados. Nesse momento o filtro do *plugin* uApprove entra em ação e exibe para o usuário uma página contendo os termos de uso do Provedor de Identidades (Figura 8).

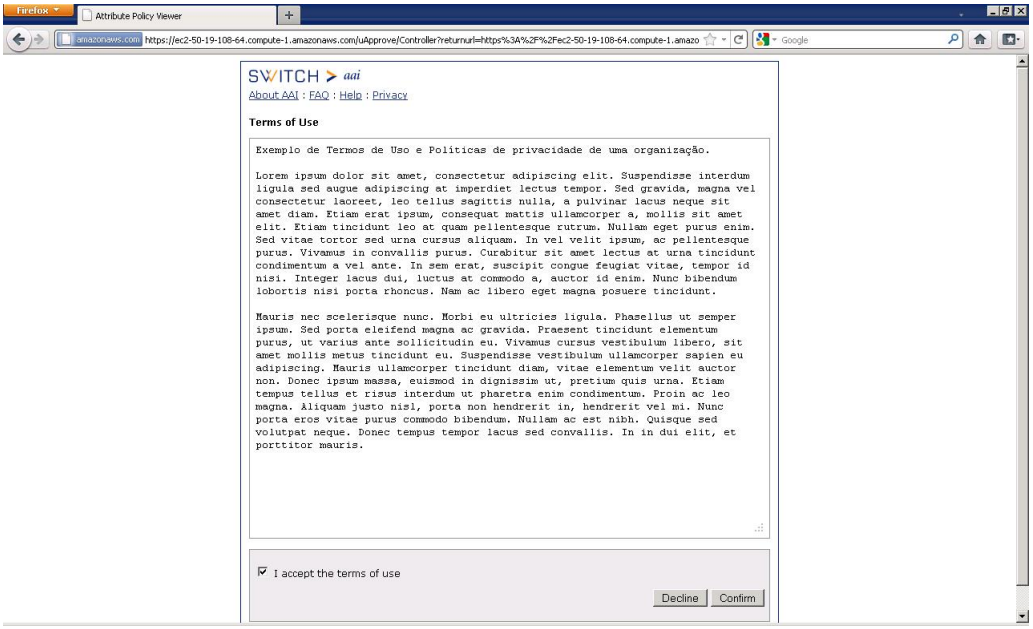

Figura 8: Termos de uso

4. Caso o usuário aceite os termos de uso o *plugin* o redireciona para uma página em que são exibidos os atributos que serão liberados (Figura 9).

Figura 9: Atributos que serão liberados

5. O usuário é novamente requisitado a aceitar a liberação de seus atributos e, caso esteja de

acordo, será levado à página de acesso protegido do provedor de serviços, agora devidamente autenticado (Figura 10).

Figura 10: Página protegida do SP, exibindo os atributos liberados

6.5.2 CASO DE USO 2

Nesse caso de uso o usuário não concorda com os termos de uso de seus dados apresentado pelo provedor de identidades. Os passos de 1 a 3 do caso de uso anterior são idênticos. Entretanto, quando perguntado os termos de uso são exibidos, no passo 3, o usuário escolhe não aceitá-los. O usuário é então redirecionado para uma nova página *web*, onde é informado que o processo de autenticação foi cancelado e ele não pode acessar o serviço protegido (Figura 11).

Figura 11: Página de cancelamento do processo de *login*

6.5.3 CASO DE USO 3

Esse caso de uso representa os acessos subsequentes do usuário ao mesmo SP. Nos acessos subsequentes, desde que não sejam alterados os termos de uso ou os atributos requisitados o usuário não é mais alertado pelo *plugin*, sendo redirecionado diretamente ao provedor de serviços.

6.5.4 OUTROS CASOS DE USO

Alguns outros casos de uso que podem ser mencionados são:

- Acesso do mesmo usuário a outros SPs;
- Acesso de outros usuários ao mesmo SP;
- Mudança nos termos de uso do IdP;
- Mudança nos atributos requisitados pelo SP;

Em todos esses casos de uso a utilização da aplicação será semelhante ao caso de uso 1.

7 CONCLUSÕES E TRABALHOS FUTUROS

Com o que foi relatado nos capítulos iniciais é possível perceber que cada vez mais se torna necessária a utilização de identidades digitais, nos mais diversos cenários. E para dar suporte a essa utilização é necessário que existam sistemas de gerenciamento de identidades seguros e confiáveis.

Entre essas características de segurança e confiabilidade a privacidade desponta como uma das mais preocupantes, por afetar diretamente os usuários dos serviços e pelo grande número de incidentes recentes que podem ser encontrados envolvendo vazamento de informações pessoais.

Com o grande crescimento do número de serviços disponíveis em nuvens computacionais é natural transferir a atenção para os novos problemas que esses serviços apresentam. Como foi descrito no Capítulo 5, muitos desafios ainda se encontram em aberto quando se trata de segurança e, mais especificamente, gerenciamento de identidades em ambientes de computação em nuvem.

7.1 PRINCIPAIS CONTRIBUIÇÕES

Nesse trabalho foi possível identificar problemas específicos de privacidade no gerenciamento de identidades: a falta de consciência dos usuários quanto à liberação de seus atributos para provedores de serviço e a falta de preocupação dos provedores de identidades quanto à apresentação de seus termos de uso.

A proposta de solução, com o uso combinado dos softwares Shibboleth e uApprove, mostrou que é possível resolver os dois problemas de maneira eficiente e sem comprometer a usabilidade da aplicação.

A proposta de solução se mostrou viável e pôde ser implantada em uma nuvem pública, o que garante a possibilidade de replicação da solução e utilização em federações consolidadas.

7.2 TRABALHOS FUTUROS

Como descrito no Capítulo 5, a pesquisa na área ainda é recente e, por isso, muitos problemas ainda se encontram em aberto e existem possibilidades interessantes de trabalhos que ainda podem ser realizados.

Um bom exemplo é o projeto uApprove.jp da federação acadêmica japonesa Gakunin (ORAWIWATTANAKUL et al., 2010). O uApprove.jp é uma extensão do uApprove que permite que os usuário definam especificamente quais dos seus atributos querem que sejam liberados.

Outras iniciativas buscam formas de automatizar a verificação de compatibilidade entre políticas de provedores de serviços e políticas definidas pelos usuários, utilizando protocolos como o P3P, citado no Capítulo 3.

Além disso, ainda existem outros aspectos de privacidade que devem ser considerados, como o emprego de uma legislação eficiente e a forma de armazenamento dos dados sensíveis.

APÊNDICE

A.1 ARQUIVOS DE CONFIGURAÇÃO

Aqui encontram-se os excertos dos arquivos de configuração utilizados no desenvolvimento do trabalho. As partes contendo reticências representam informações que não foram alteradas dos arquivos originais ou não são relevantes.

A.1.1. HTTPD-MOD-PROXY.CONF

```
ProxyRequests Off

ProxyPass         /cas ajp://localhost:8009/cas retry=5
ProxyPass         /idp ajp://localhost:8009/idp retry=5
ProxyPass         /uApprove ajp://localhost:8009/uApprove retry=5
```

A.1.2. HTTPD.CONF

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
...
Include conf/extra/httpd-mod-proxy.conf
```

A.1.3. DEPLOYERCONFIGCONTENT.XML

```
<bean class="org.jasig.cas.adaptors.ldap.BindLdapAuthenticationHandler"
   >
    <property name="filter" value="uid=%u" />
    <property name="searchBase" value="dc=compute-1,dc=amazonaws,dc=com
       " />
    <property name="contextSource" ref="contextSource" />
    <property name="ignorePartialResultException" value="yes" />
</bean>

<bean id="contextSource" class="org.jasig.cas.adaptors.ldap.util.
   AuthenticatedLdapContextSource">
```

```
        <property name="urls">
                <list>
                    <value>ldap://ec2-50-16-87-125.compute-1.amazonaws.com/
                        </value>
                </list>
        </property>

    ...

     <property name="userName" value="cn=admin,dc=compute-1,dc=amazonaws,dc
         =com"/>
          <property name="password" value="1234"/>
          <property name="baseEnvironmentProperties">
```

A.1.4. RELYING-PARTY.XML

```
    ...
    <MetadataProvider id="URLMD" xsi:type="FileBackedHTTPMetadataProvider"
        xmlns="urn:mace:shibboleth:2.0:metadata"
                          metadataURL="http://www.testshib.org/metadata/
                              testshib-providers.xml"
                          backingFile="c:\shibboleth-idp/metadata/testshib-
                              metadata.xml">
        </MetadataProvider>
    ...
```

```
    ...
    <security:Credential id="IdPCredential" xsi:type="
        security:X509Filesystem">
        <security:PrivateKey>c:\PKI/shibidp.key</security:PrivateKey>
        <security:Certificate>c:\PKI/shibidp.crt</security:Certificate>
    </security:Credential>
    ...
```

A.1.5. WEB.XML

```
...
<filter>
   <filter-name>CAS Filter</filter-name>
   <filter-class>edu.yale.its.tp.cas.client.filter.CASFilter</filter-class>
   <!-- URL of login page of CAS Server -->
   <init-param>
```

```
        <param-name>edu.yale.its.tp.cas.client.filter.loginUrl</param-name>
        <param-value>https://ec2-50-19-108-64.compute-1.amazonaws.com/cas/
            login</param-value>
    </init-param>
    <!-- URL to validation URL of CAS Server -->
    <init-param>
        <param-name>edu.yale.its.tp.cas.client.filter.validateUrl</param-
            name>
        <param-value>https://ec2-50-19-108-64.compute-1.amazonaws.com/cas/
            serviceValidate</param-value>
    </init-param>
    <!-- Full hostname with port number to be filtered. The port number is
        not required for standard ports (80,443) -->
    <init-param>
        <param-name>edu.yale.its.tp.cas.client.filter.serverName</param-name
            >
        <param-value>ec2-50-19-108-64.compute-1.amazonaws.com/</param-value>
    </init-param>
    <!-- expose REMOTE_USER (from CAS Client version 2.1.0) -->
    <init-param>
        <param-name>edu.yale.its.tp.cas.client.filter.wrapRequest</param-
            name>
        <param-value>true</param-value>
    </init-param>
     </filter>
         <filter-mapping>
                 <filter-name>CAS Filter</filter-name>
                 <url-pattern>/Authn/RemoteUser</url-pattern>
     </filter-mapping>
...
```

```
    <web-app>
   ...

  <filter>
    <filter-name>uApprove IdP plugin</filter-name>
    <filter-class>ch.SWITCH.aai.uApprove.idpplugin.Plugin</filter-class>
    <init-param>
      <param-name>Config</param-name>
      <param-value>
        C:/uApprove/conf/idp-plugin.properties;
        C:/opt/uApprove/conf/common.properties;
      </param-value>
```

```
    </init-param>
  </filter>

  <filter-mapping>
    <filter-name>uApprove IdP plugin</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

</web-app>
```

A.1.6. ATTRIBUTE-RESOLVER.XML

```
...
   <resolver:AttributeDefinition
               id="eduPersonAffiliation"
               xsi:type="Simple"
               xmlns="urn:mace:shibboleth:2.0:resolver:ad"
               sourceAttributeID="eduPersonAffiliation">
        <resolver:Dependency ref="staticAttributes" />

        <resolver:AttributeEncoder
               xsi:type="SAML1String"
               xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
               name="urn:mace:dir:attribute-def:eduPersonAffiliation" />

        <resolver:AttributeEncoder
               xsi:type="SAML2String"
               xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
               name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
               friendlyName="eduPersonAffiliation" />
    </resolver:AttributeDefinition>

        <resolver:AttributeDefinition
               id="brPersonCPF"
               xsi:type="Simple"
               xmlns="urn:mace:shibboleth:2.0:resolver:ad"
               sourceAttributeID="brPersonCPF">
        <resolver:Dependency ref="MyAD" />

        <resolver:AttributeEncoder
               xsi:type="SAML1String"
               xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
               name="urn:mace:dir:attribute-def:eduPersonAffiliation" />
```

```
    <resolver:AttributeEncoder
            xsi:type="SAML2String"
            xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
            friendlyName="eduPersonAffiliation" />
</resolver:AttributeDefinition>

    <resolver:AttributeDefinition
            id="cn"
            xsi:type="Simple"
            xmlns="urn:mace:shibboleth:2.0:resolver:ad"
            sourceAttributeID="cn">
    <resolver:Dependency ref="MyAD" />

    <resolver:AttributeEncoder
            xsi:type="SAML1String"
            xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:mace:dir:attribute-def:eduPersonAffiliation" />

    <resolver:AttributeEncoder
            xsi:type="SAML2String"
            xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
            friendlyName="eduPersonAffiliation" />
</resolver:AttributeDefinition>

    <resolver:AttributeDefinition
            id="sn"
            xsi:type="Simple"
            xmlns="urn:mace:shibboleth:2.0:resolver:ad"
            sourceAttributeID="sn">
    <resolver:Dependency ref="MyAD" />

    <resolver:AttributeEncoder
            xsi:type="SAML1String"
            xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:mace:dir:attribute-def:eduPersonAffiliation" />

    <resolver:AttributeEncoder
            xsi:type="SAML2String"
            xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
```

```
            friendlyName="eduPersonAffiliation" />
</resolver:AttributeDefinition>

    <resolver:AttributeDefinition
            id="eduPersonPrincipalName"
            xsi:type="Scoped"
            xmlns="urn:mace:shibboleth:2.0:resolver:ad"
     scope="compute-1.amazonaws.com"
            sourceAttributeID="uid">
    <resolver:Dependency ref="MyAD" />

    <resolver:AttributeEncoder
            xsi:type="SAML1ScopedString"
            xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:mace:dir:attribute-def:eduPersonPrincipalName" />

    <resolver:AttributeEncoder
            xsi:type="SAML2ScopedString"
            xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
            friendlyName="eduPersonPrincipalName" />
</resolver:AttributeDefinition>

    <resolver:AttributeDefinition
                    id="eduPersonTargetedID"
                    xsi:type="SAML2NameID"
                    xmlns="urn:mace:shibboleth:2.0:resolver:ad"
                    nameIdFormat="urn:oasis:names:tc:SAML:2.0:nameid-
                       format:persistent"
                    sourceAttributeID="computedID">
    <resolver:Dependency ref="computedID" />

    <resolver:AttributeEncoder
                    xsi:type="SAML1XMLObject"
                    xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
        name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10" />

    <resolver:AttributeEncoder
                    xsi:type="SAML2XMLObject"
                    xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
        name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"
                    friendlyName="eduPersonTargetedID" />
</resolver:AttributeDefinition>
```

```
...

        <resolver:DataConnector
                id="MyAD"
                xsi:type="LDAPDirectory"
                xmlns="urn:mace:shibboleth:2.0:resolver:dc"
        ldapURL="ldap://ec2-50-16-87-125.compute-1.amazonaws.com/"
                baseDN="dc=compute-1,dc=amazonaws,dc=com"
                principal="cn=admin,dc=compute-1,dc=amazonaws,dc=com"
        principalCredential="1234">
       <FilterTemplate>
           <![CDATA[
               (uid=$requestContext.principalName)
           ]]>
       </FilterTemplate>
        </resolver:DataConnector>
...
```

A.1.7. ATTRIBUTE-FILTER.XML

```
...
<AttributeFilterPolicy id="releaseBasicAttributesToAnyone">
        <PolicyRequirementRule xsi:type="basic:ANY" />

        <AttributeRule attributeID="eduPersonAffiliation">
            <PermitValueRule xsi:type="basic:ANY" />
        </AttributeRule>

                <AttributeRule attributeID="eduPersonPrincipalName">
                        <PermitValueRule xsi:type="basic:ANY" />
                </AttributeRule>

                <AttributeRule attributeID="cn">
                        <PermitValueRule xsi:type="basic:ANY" />
                </AttributeRule>

                <AttributeRule attributeID="sn">
                        <PermitValueRule xsi:type="basic:ANY" />
                </AttributeRule>

                <AttributeRule attributeID="brPersonCPF">
                        <PermitValueRule xsi:type="basic:ANY" />
```

```
            </AttributeRule>

            <AttributeRule attributeID="eduPersonTargetedID">
        <PermitValueRule xsi:type="basic:ANY" />
            </AttributeRule>

    </AttributeFilterPolicy>
...
```

REFERÊNCIAS

AMAZON. *Amazon Elastic Compute Cloud*. 2010. Acessado em 20 de junho de 2011. Disponível em: <http://aws.amazon.com/ec2/>.

APPELBAUM, Jacob et al. *Re: Ensuring adequate security in Google's cloud based services*. 2010. Acessado em 20 de junho de 2011. Disponível em: <http://www.cloudprivacy.net/letter/>.

BENANTAR, Messaoud. *Access Control Systems: Security, Identity Management and Trust Models*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005. ISBN 0387004459.

BRAGHIN, CHIARA; CREMONINI, MARCO; ARDAGNA, CLAUDIO AGOSTINO. Net privacy. Morgan Kaufmann, 2009. Disponível em: <http://hdl.handle.net/2434/64439>.

Cloud Identity Summit. *Cloud Identity Summit 2011*. 2011. Acessado em 20 de junho de 2011. Disponível em: <www.cloudidentitysummit.com>.

Cloud Security Alliance. *Domain 12: Guidance for Identity and Access Management V2.1*. 2010.

CORDOVA, André Siqueira de. *Aplicação Prática de um Sistema de Gerenciamento de Identidades*. Itajaí, SC: [s.n.], 2006. Trabalho de Conclusão de Curso, Ciência da Computação, UNIVALI.

CULTURADIGITAL. *Os rumos da lei de proteção de dados*. 2011. Acessado em 20 de junho de 2011. Disponível em: <http://culturadigital.br/dadospessoais/os-rumos-da-lei-de-protecao-de-dados/>.

EU. *Identity Management Systems (IMS): Identification and Comparison Study*. [S.l.], September 2003. Disponível em: <https://www.datenschutzzentrum.de/projekte/idmanage/study.htm>.

Federal Trade Comission. *Fair Information Practice Principles*. 2010. Acessado em 20 de junho de 2011. Disponível em: <http://www.ftc.gov/reports/privacy3/fairinfo.shtm>.

GOLDBERG, I.; WAGNER, D.; BREWER, E. Privacy-enhancing technologies for the internet. In: *Compcon '97. Proceedings, IEEE*. [S.l.: s.n.], 1997. p. 103 –109.

GOOGLE. *Google Apps*. 2010. Acessado em 20 de junho de 2011. Disponível em: <http://www.google.com/apps>.

GROBAUER, Bernd; WALLOSCHEK, Tobias; STOCKER, Elmar. Understanding cloud computing vulnerabilities. *IEEE Security and Privacy*, IEEE Computer Society, Los Alamitos, CA, USA, v. 9, p. 50–57, 2011. ISSN 1540-7993.

GROß, Thomas. Security analysis of the saml single sign-on browser/artifact profile. In: *Proceedings of the 19th Annual Computer Security Applications Conference*. Washington, DC, USA: IEEE Computer Society, 2003. (ACSAC '03), p. 298–. ISBN 0-7695-2041-3. Disponível em: <http://portal.acm.org/citation.cfm?id=956415.956441>.

HUANG, Xin; ZHANG, Tingting; HOU, Yifan. Id management among clouds. In: *Future Information Networks, 2009. ICFIN 2009. First International Conference on*. [S.l.: s.n.], 2009. p. 237 –241.

INTERNET2. *About Shibboleth*. 2010. Acessado em 20 de junho de 2011. Disponível em: <http://shibboleth.internet2.edu/about.html>.

INTERNET2. *Shibboleth Federations*. 2010. Acessado em 20 de junho de 2011. Disponível em: <https://spaces.internet2.edu/display/SHIB/ShibbolethFederations>.

INTERNET2. *eduPerson*. 2011. Acessado em 20 de junho de 2011. Disponível em: <http://middleware.internet2.edu/eduperson/>.

INTERNET2. *TestShib Two*. 2011. Acessado em 20 de junho de 2011. Disponível em: <https://www.testshib.org/testshib-two/index.jsp>.

JASIG. *JASIG CAS*. 2011. Acessado em 20 de junho de 2011. Disponível em: <http://www.jasig.org/cas>.

LEE, Hyangjin; JEUN, Inkyoung; JUNG, Hyuncheol. Criteria for evaluating the privacy protection level of identity management services. *Emerging Security Information, Systems, and Technologies, The International Conference on*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 155–160, 2009.

Marcon Jr., Arlindo et al. Aspectos de segurança e privacidade em ambientes de computação em nuvem. In: *Livro-texto de minicursos do SBSeg 2010*. Porto Alegre, RS: Sociedade Brasileira de Computação, 2010. v. 1, p. 53 –102.

MATHER, Tim; KUMARASWAMY, Subra; LATIF, Shahed. *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*. [S.l.]: O'Reilly Media, Inc., 2009. ISBN 0596802765, 9780596802769.

MICROSOFT. *Windows Azure*. 2010. Acessado em 20 de junho de 2011. Disponível em: <http://www.microsoft.com/windowsazure/resources/default.aspx>.

MICROSOFT. *Introducing Windows CardSpace*. 2011. Acessado em 20 de junho de 2011. Disponível em: <http://msdn.microsoft.com/en-us/library/aa480189.aspx>.

OASIS. *Online Community for the SAML OASIS Standard*. 2010. Acessado em 20 de junho de 2011. Disponível em: <http://saml.xml.org/>.

ONU. *Universal Declaration of Human Rights*. 1948. Acessado em 20 de junho de 2011. Disponível em: <http://www.un.org/en/documents/udhr/>.

Open Security Foundation. *DataLossDB*. 2010. Acessado em 20 de junho de 2011. Disponível em: <http://datalossdb.org/>.

OpenID Foundation. *What is OpenID?* 2010. Acessado em 20 de junho de 2011. Disponível em: <http://openid.net/get-an-openid/what-is-openid/>.

OpenLDAP Foundation. *OpenLDAP*. 2011. Acessado em 20 de junho de 2011. Disponível em: <http://www.openldap.org/>.

Oracle Corporation. *Java SE Overview*. 2011. Acessado em 20 de junho de 2011. Disponível em: <http://www.oracle.com/technetwork/java/javase/overview/index.html>.

Oracle Corporation. *MySQL*. 2011. Acessado em 20 de junho de 2011. Disponível em: <http://www.mysql.com/>.

ORAWIWATTANAKUL, T. et al. User-controlled privacy protection with attribute-filter mechanism for a federated sso environment using shibboleth. In: *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2010 International Conference on*. [S.l.: s.n.], 2010. p. 243 –249.

PEARSON, Siani. Taking account of privacy when designing cloud computing services. In: *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*. Washington, DC, USA: IEEE Computer Society, 2009. (CLOUD '09), p. 44–52. ISBN 978-1-4244-3713-9. Disponível em: <http://dx.doi.org/10.1109/CLOUD.2009.5071532>.

Protect Network. *Protect Network*. 2011. Acessado em 20 de junho de 2011. Disponível em: <www.protectnetwork.org>.

RISTENPART, Thomas et al. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In: *Proceedings of the 16th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2009. (CCS '09), p. 199–212. ISBN 978-1-60558-894-0. Disponível em: <http://doi.acm.org/10.1145/1653662.1653687>.

RNP. *Federação CAFe*. 2010. Acessado em 20 de junho de 2011. Disponível em: <http://www.cafe.rnp.br/>.

RNP. *brEduPerson*. 2011. Acessado em 20 de junho de 2011. Disponível em: <http://wiki.rnp.br/display/cafewebsite/brEduPerson>.

SALESFORCE. *Salesforce*. 2010. Acessado em 20 de junho de 2011. Disponível em: <http://www.salesforce.com>.

SHOSTACK, Adam; SYVERSON, Paul. *What price privacy*. 2003.

SWITCH. *uApprove*. 2011. Acessado em 20 de junho de 2011. Disponível em: <http://www.switch.ch/aai/support/tools/uApprove.html>.

TAKABI, Hassan; JOSHI, James B.D.; AHN, Gail-Joon. Security and privacy challenges in cloud computing environments. *IEEE Security and Privacy*, IEEE Computer Society, Los Alamitos, CA, USA, v. 8, p. 24–31, 2010. ISSN 1540-7993.

The Apache Software Foundation. *Apache HTTP Server*. 2011. Acessado em 20 de junho de 2011. Disponível em: <http://projects.apache.org/projects/http_server.html>.

The Apache Software Foundation. *Apache Tomcat*. 2011. Acessado em 20 de junho de 2011. Disponível em: <http://projects.apache.org/projects/tomcat.html>.

TRUSTE. *TRUSTe*. 2010. Acessado em 20 de junho de 2011. Disponível em: <http://www.truste.com/>.

W3C. *P3P: The Platform for Privacy Preferences*. 2011. Acessado em 20 de junho de 2011. Disponível em: <http://www.w3.org/P3P/>.

WINDLEY, Phillip. *Digital Identity*. [S.l.]: O'Reilly Media, Inc., 2005. ISBN 0596008783.

WINDLEY, Phillip J. *Understanding Digital Identity Management*. 2003.

ZARANDIOON, Saman; YAO, Danfeng; GANAPATHY, Vinod. Privacy-aware identity management for client-side mashup applications. In: *Proceedings of the 5th ACM workshop on Digital identity management*. New York, NY, USA: ACM, 2009. (DIM '09), p. 21–30. ISBN 978-1-60558-786-8. Disponível em: <http://doi.acm.org/10.1145/1655028.1655036>.