CEP – Uma proposta de gerenciamento de identidades em *Cloud Computing* utilizando OpenAM e Captive Portal

Andreia Rosangela Kessler Mühlbeier
UFSM
andreiarkmuhlbeier@gmail.com

Samuel Stieler
UFSM
samuel.stieler@gmail.com

Felipe Becker Nunes
UFSM
nunesfb@gmail.com

Roseclea Duarte Medina
UFSM
roseclea.medina@gmail.com

Gleizer Bierhalz Voss
UFSM
gleizer.voss@gmail.com

Érico Marcelo Hoff do Amaral
UNIPAMPA
ericohoffamaral@gmail.com

***Resumo* - Este artigo apresenta uma proposta de gerenciamento de identidades em um ambiente de *Cloud Computing* utilizando Captive Portal em conjunto com os recursos do OpenAM. O modelo abrange mecanismos de controle de acesso baseado no contexto de papéis provendo uma Camada Extra de Proteção (CEP) a um ambiente de computação em nuvem.**

I. INTRODUÇÃO

A Computação em Nuvem (*Cloud Computing*) vem se tornando um paradigma da área computacional com grande evolução no cenário atual. O aumento de usuários no decorrer da última década propiciou o desenvolvimento de alternativas positivas para a resolução de questões importantes relacionadas às pessoas e empresas.

Esse paradigma aparece com o intuito de disponibilizar serviços de tecnologia da informação sob demanda, no qual o pagamento é baseado conforme a sua utilização. No entanto, há algumas preocupações relacionadas à segurança que precisam ser resolvidas, por exemplo, confidencialidade, autenticidade e integridade das informações e aplicações armazenadas em nuvem [1].

Usuários carecem de garantias rígidas que suas informações estarão bem protegidas pela empresa responsável. Em conformidade com essa situação, diversas pesquisas e propostas têm sido apresentadas [2] [3] [4] na tentativa de quebrar este cenário de desconfiança e assim buscar fornecer maiores níveis e garantias de segurança.

Diante deste contexto, este trabalho apresenta uma proposta de controle de acesso para um ambiente de *Cloud Computing*, com a utilização dos mecanismos OpenAM e Captive Portal. Estes realizam a autenticação e autorização dos usuários no ambiente, fornecendo assim uma Camada Extra de Proteção (CEP).

Este trabalho está organizado da forma que segue: na seção 2 são apresentados os trabalhos relacionados; a seção 3 descreve questões sobre segurança; o gerenciamento de identidades é descrito na seção 4; na seção 5 são descritos os métodos de controle de acesso OpenAM e Captive Portal e dados sobre a implementação desta proposta; por fim na seção 6 é apresentado as considerações finais sobre a pesquisa realizada.

II. TRABALHOS RELACIONADOS

O trabalho descrito em [2] apresenta uma solução alternativa a um *Identity as a Service* (IDaaS), com o objetivo de fornecer um gerenciamento de identidades baseado no conceito de identidade digital federada. A solução utiliza o Shibboleth, uma ferramenta baseada em *Security Assertion Markup Language (*SAML), que fornece apoio às tarefas de autenticação, autorização e federação de identidades. Desta forma, os autores destacam que é possível oferecer um serviço que permite ao mesmo tempo acesso público, nos casos de acesso apenas para leitura, ou pode exigir credenciais, solicitando ao usuário uma conexão validada para alterar documentos.

A proposta de [3] apresenta uma solução de segurança para o serviço de nuvem baseado em *Usage Control* (UCON), um modelo apenas conceitual, e sem especificação de realização concreta. O modelo UCON é composto por seis partes: Assuntos, Direitos, Objetos, Autorizações, Obrigações e Condições. Como trabalho futuro, os autores pretendem criar o protótipo do sistema de segurança para o serviço de *cloud*.

O trabalho descrito em [4] propõe a criação de identidade centralizada para o gerenciamento de identidade na nuvem. A abordagem é baseada em pacotes ativos e identificação anônima. Destacam-se nessa solução: a independência de terceiros, o fornecimento de informações mínimas ao *Service Provider* (SP) e a capacidade de usar dados de identidade em *hosts* não confiáveis.

Pode-se relacionar ainda o trabalho de [5], que apresenta a proposta de uma arquitetura para uma nova abordagem de "proteção mútua". Ela é baseada no conceito de confiança mútua e na especificação de perfis definidos na forma de um vetor matricial. Por fim tem-se [6] que apresenta um protótipo, utilizando a tecnologia de agentes como uma forma de oferecer privacidade aos dados dos clientes em um ambiente de *cloud computing*. A abordagem utiliza predicados sobre os dados criptografados e computação segura *multi-party computation* (MPC) para a negociação na utilização de um serviço na nuvem.

III. SEGURANÇA EM *CLOUD COMPUTING*

Quando o assunto de computação em nuvem é abordado, a segurança é uma das objeções mais frequentemente citadas, apesar da maioria das empresas terceirizarem os pagamentos e utilizarem serviços de *e-mail* externos [8]. Dentre as principais preocupações dos usuários e empresas, estão questões referentes sobre quais usuários terão acesso às informações e quais os riscos que existem em utilizar uma aplicação armazenada em um servidor na nuvem. O fato de que uma nuvem é composta por um aglomerado de informações pode marcá-la como um alvo propício para ataques de potenciais invasores [7].

Deste modo, as organizações precisam verificar os riscos existentes em disponibilizar suas informações e aplicações em servidores na nuvem em relação as suas necessidades, avaliando dessa forma a solução que irá proporcionar um maior número de vantagens para o seu negócio.

IV. GERENCIAMENTO DE IDENTIDADES

Em ambientes que possuam restrições de acesso, por exemplo, uma nuvem que hospeda aplicações e recursos, o usuário terá que realizar a sua identificação junto da entidade de verificação desse ambiente, conforme as permissões concedidas pela entidade.

Uma identidade pode ser definida como uma representação de um elemento, de forma que seja possível identificá-la dentro de um contexto em particular [9]. Um usuário ou aplicação necessitará de um identificador, como: CPF ou *e-mail*, para ser reconhecido no ambiente.

Conforme este contexto, a gerência de identidades realiza o processo de controle, com o objetivo de permitir que uma entidade possa entrar no ambiente desejado, de acordo com suas permissões de acesso.

A. Shibboleth

Shibboleth [10] é a solução de identidade federada mais utilizada mundialmente, sua arquitetura é composta por componentes livres e *open source*, principalmente o *Security Assertion Markup Language* (SAML). O SAML é um padrão aberto no formato *eXtensible Markup Language* (XML) para a troca de dados de autenticação e autorização em um domínio de segurança. O Shibboleth fornece um mecanismo de *Single Sign-On* (SSO) [2], que conforme a definição do *Opengroup* é um mecanismo que permite um usuário cadastrado acessar por meio de uma única autenticação, todos os sistemas a que tenha permissão de acesso.

B. Higgins

Higgins [11] é um *Personal Date Service* (PDS) que permite ao usuário controlar como e com quem os seus dados serão compartilhados, seja com amigos ou organizações em que confia. Esse projeto foi desenvolvido pela Eclipse Foundation, e possui quatro partes principais: o PDS, que é o *Back-end* de serviços de apoio ao cliente *Web*; o *Attribute Data Storage* (ADS), que expõe os dados para o Portal e o *Higgins Browser Extension* (HBX), que utiliza uma interface de mensagens HTTP/Comet; Cliente que é a interface que permite ao usuário ver e editar os atributos; e por fim o HBX que é a extensão que carrega os programas *JavaScript* para o PDS e roda-os no navegador.

C. OpenAM

O OpenAM [12] foi inicialmente denominado de OpenSSO e desenvolvido pela Oracle. Porém com a aquisição desta pela Sun Microsystems, ele passou a ser desenvolvido pela ForgeRock sob o nome de OpenAM.

Ele é um sistema de código aberto (*open source*), que prove os serviços de autenticação, autorização, verificação de validade de *tokens, login* e provisão de identidades. O OpenAM possui três formas de acesso: utilização de uma requisição HTTP, por meio dos serviços *Web* ou com a utilização de um agente. Ele pode ser considerado um IDaaS que realiza a provisão de identidades e controle de acesso [13]. Sua arquitetura é subdividida em 3 camadas: interface do cliente, núcleo e a camada de integração.

D. Captive Portal

Segundo [14], o Captive Portal funciona como um roteador ou *gateway*, não permitindo que haja tráfego de informações antes da autenticação do usuário. Ele obriga este a visualizar uma página de login ou acesso, onde o usuário deve realizar uma autenticação para obter acesso ao local que deseja, por exemplo, aplicações e informações armazenadas em uma nuvem.

Esta tecnologia faz o monitoramento de pacotes quando um usuário realiza o acesso a uma aplicação ou base de dados, sendo a conexão deste redirecionada para uma página de *login*, na qual ocorre a autenticação/autorização de acesso. A transmissão de informação entre o usuário e a página de acesso é criptografada com a utilização do protocolo *Secure Socket Layer* (SSL) em ambas as direções. [15].

E. Mecanismos escolhidos

A Tabela I apresenta um comparativo dos mecanismos de controle de acesso mencionados anteriormente. Podem-se observar as principais diferenças com relação ao suporte de padrões abertos e também sobre o tipo de paradigma que cada um possui.

Tabela I
Comparativo dos mecanismos de controle de acesso

Mecanismos / Soluções	Shibboleth 2x	Higgins 2.0	OpenAM 10
SSO	X	X	X
SAML 2.0	X	X	X
OpenID 2.0	-	-	Plugin
XACML 2.0	Extenção	-	X
OAuth 1.0	-	-	Plugin
S.Agentes	-	-	X
Paradigma	Federado	C. Usuário	Federado

Fonte - Adaptado de Feliciano 2011.

Baseado na tabela comparativa apresentada anteriormente, foi escolhido o mecanismo OpenAM em relação ao Shibboleth e o Higgins. Os motivos para esta escolha são os seguintes:

- O OpenAM permite a implementação de novos padrões por meio da instalação de *plugins*, por exemplo, o OpenID e OAuth.
- Ele possui uma arquitetura flexível e paradigma centrado no usuário, que possibilita o compartilhamento de informações de identidades.
- Está sendo utilizado de forma crescente pelos usuários e possui bastante tempo de desenvolvimento. Além disso, teve sua última versão lançada no mês de junho de 2012, sendo esta a escolhida para ser utilizada.

Adicionalmente ao mecanismo do OpenAM, será utilizado o Captive Portal com o objetivo de fornecer uma camada extra de proteção ao ambiente da nuvem. Com isso, é realizado o gerenciamento de identidades e o controle de acesso de forma conjunta. O Captive Portal foi escolhido pelo fato de poder ser integrado ao OpenAM, impedindo os usuários de acessar o ambiente caso não tenham realizado a autenticação.

V. Controle de acesso utilizando OpenAM e Captive Portal

É apresentado um controle de acesso para realizar a autenticação/autorização dos usuários em um ambiente de nuvem, na qual está hospedado o ambiente virtual de aprendizagem Moodle, softwares para a gerência de rede e um laboratório virtual. Além destes serviços, é realizado o controle de papéis dos usuários, de forma a restringir o acesso às páginas somente para perfis específicos, impedindo o acesso não autorizado.

O controle de acesso utiliza os serviços do Captive Portal, que adiciona uma camada extra de proteção, trabalhando em conjunto com o OpenAM para atender aos requisitos de segurança citados anteriormente. Simultaneamente realiza o gerenciamento de perfis, permitindo acesso somente aos pontos que lhe são liberados no ambiente.

A. Arquitetura de Controle

Com o desafio de idealizar um controle de acesso diferenciado, a arquitetura precisa ser também um diferencial. A arquitetura de controle da *Cloud* é simples, porém rica em detalhes demonstrando o seu diferencial quando se trata em segurança, conforme mostra a Figura 1. Ela é definida a partir de três elementos: o Captive Portal, o OpenAM e o servidor de banco de dados e *login*.

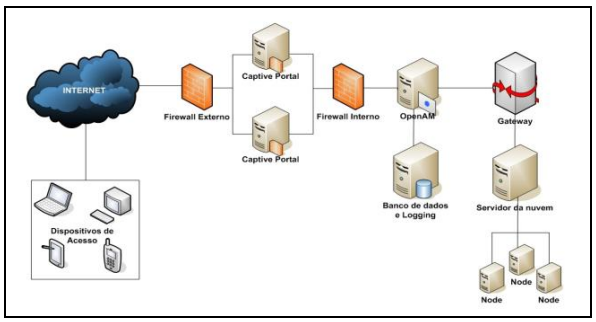

Figura 1. Visão geral da arquitetura de controle do sistema CEP.

O Captive Portal irá atuar como um agente, com o papel de verificar se o usuário que está solicitando o acesso aos recursos disponíveis na *Cloud* já realizou a sua autenticação no ambiente. Caso contrário, será redirecionado para uma página especial de *login*, onde deverá inserir seus dados, por exemplo, CPF e RG. A conexão entre o usuário e a página de acesso é criptografada com a utilização do protocolo SSL em ambas as partes. O desenvolvimento do Captive Portal será feito com a utilização da linguagem de programação PHP e com o sistema gerenciador de banco de dados MySQL.

O OpenAM é responsável pela parte de autenticação e autorização dos usuários no ambiente, além de realizar o gerenciamento dos perfis. Ele funciona com um mecanismo de autenticação *Single Sign-On* (SSO), conforme mencionado anteriormente, que faz a troca de mensagens via XML e certificação via SAML, sendo esses os principais padrões do mecanismo. O OpenAM gerencia o uso de agentes, que serão os responsáveis por enviar a requisição via Captive Portal para o OpenAM, em que é realizada a consulta das identidades de cada usuário.

O servidor de banco de dados e *login* tem a função de armazenar as informações dos usuários cadastrados no ambiente da nuvem, para que a entidade que for realizar a identificação verifique se um usuário é legítimo ou não. Ele também possui uma tabela que deve armazenar os *log's* gerados por cada acesso ou tentativa mal sucedida de acesso ao ambiente, para quando necessário analisar o comportamento deste. Outro detalhe é a existência de uma *flag* no banco de dados, com valores de V (Ativo) ou F (Inativo), que indicam se a sessão está ativa ou não.

B. Funcionamento do controle de acesso

Para que os usuários de uma *Cloud* obtenham permissão de acesso aos serviços hospedados, é necessária a realização de algumas etapas de autenticação e autorização. Na Figura 2 pode ser visualizado o fluxograma da proposta, no qual é apresentado as etapas de funcionamento do controle de acesso, descritas de forma detalhada.

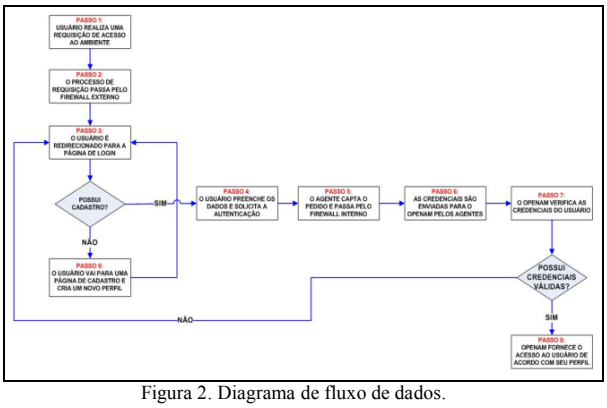

Figura 2. Diagrama de fluxo de dados.

No passo 1, o usuário realiza o acesso ao endereço do serviço na nuvem pelo *browser*, utilizando qualquer equipamento, por exemplo, *notebook*, *desktop* ou dispositivo móvel; No passo 2, o processo de requisição passa primeiramente por um *firewall* externo, que tem como função principal a verificação de segurança por meio das regras de acesso pré-definidas pelo administrador de rede, protegendo a nuvem de ataques maliciosos externos;

No passo 3, o *firewall* externo redireciona o usuário para a tela do Captive Portal, que exibe uma página especial criada para realizar o *login* (passo 4). Caso o usuário não possua cadastro, ele é redirecionado a uma página para a criação de um novo perfil no ambiente (passo 9). No passo 4, o usuário deve preencher os

campos necessários (número de matrícula, senha, instituição/curso) e solicitar a autenticação pelo Captive Portal.

No passo 5, o agente irá interceptar a requisição no momento da autenticação passando por um *firewall* interno, que terá a função de proteger as aplicações instaladas;

No passo 6, as credenciais do usuário são encaminhadas pelo agente para o OpenAM; No passo 7, o OpenAM verifica se o usuário possui as credenciais necessárias para o acesso com uma consulta ao banco de dados;

No passo 8, o OpenAM autoriza o acesso a nuvem, restringindo a navegação somente às páginas que o perfil do usuário tenha permissão. Caso não possua as credenciais de acesso válidas, é redirecionado para a página de *login* do Captive Portal por meio do agente.

Para a criação de um novo perfil (passo 9), o usuário deverá efetuar o cadastro, recebendo as credenciais de acesso e sendo redirecionado para a página inicial.

O novo perfil de usuário é definido de forma padrão (básico), tendo acesso somente aos recursos permitidos para este perfil. Os outros dois tipos são denominados de intermediário e avançado, cada um com recursos específicos, conforme Tabela II. O tipo de perfil poderá ser alterado através de solicitação ao administrador do sistema, que analisará a situação e concederá ou não a alteração.

O acesso as páginas é autorizado conforme o tipo do perfil que o usuário possui, desta forma, este poderá desempenhar determinadas ações, por exemplo, leitura e modificação, caso o seu perfil possibilite realizá-las. Esta forma de controle permite que arquivos e informações confidenciais sejam acessados somente por perfis que tenham permissão, possibilitando assim uma maior segurança em relação ao funcionamento do ambiente.

No momento em que um usuário cadastrado for realizar o acesso ao ambiente, será gerado um *log* de acesso criado por um *script* de rotina do banco de dados, que armazena esses *logs* no mesmo servidor. Ao mesmo tempo, é incrementado o valor de uma *flag* localizada na base de dados criando uma sessão. O valor desta pode ser V (Ativa) ou F (Inativa), indicando se o usuário pode navegar pelo ambiente. Um *time out* controla o tempo ocioso do usuário, obrigando-o a realizar a autenticação novamente, caso tenha expirado o limite estabelecido pelo administrador.

Tabela II
Tipo de Níveis de Permissões de Acesso

Tipo de Permissões	Básica	Média	Avançada
Ler	X	X	X
Enviar Arquivos	X	X	X
Copiar Arquivos	X	X	X
Modificar/Excluir Arquivos		X	X
Criar/Excluir Tarefas e Fóruns		X	X
Alterar dados perfil	X	X	X
Modificar permissões			X
Acesso Total			X

VI. CONSIDERAÇÕES FINAIS

A computação em nuvem é um paradigma relativamente recente, cujo objetivo é disponibilizar serviços de tecnologia da informação sob demanda. Desta forma, diversas propostas vêm surgindo com o objetivo de apresentar soluções e melhorias para garantir a segurança nestes ambientes.

Com base na análise descrita na seção 2, foi apresentada a proposta de utilização de uma camada extra de proteção com o uso do Captive Portal associado ao mecanismo OpenAM, que tem como funções: autenticação/autorização e o gerenciamento dos perfis dos usuários no ambiente da nuvem.

Após a descrição da proposta e do funcionamento de sua arquitetura, espera-se que seja possível realizar a inserção da Camada Extra de Proteção (CEP) ao controle de acesso. Com isso, busca-se unir todos os recursos oferecidos pelo mecanismo do OpenAM às funcionalidades que o Captive Portal disponibiliza, agregando um maior nível de segurança ao controle de acesso no ambiente da nuvem.

Como trabalho futuros, propõe-se a implementação dos testes e validação dos resultados obtidos com a aplicação, a fim de verificar a consistência da proposta.

REFERÊNCIAS

[1] F. R. C. Souza, L. O. Moreira; J. C. Machado. Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios, ERCEMAPI, 2009.

[2] M. A. P. Leandro, T. J. Nascimento, D. R. dos Santos,; C. M. Wetphall, C. B. Wetphall. Multi-Tenancy Authorization System with Federated Identity for Cloud-Based Environments Using Shibboleth. ICN 2012: The Eleventh International Conference on Networks.

[3] C. Danwei, H. Xiuli, R. Xunyi. Access control of cloud service based on UCON. CloudCom, 2009.

[4] P. Angin, L. B. O. Lilien, M. Lindermna. An Entity-centric Approach for Privacy and Identity Management in Cloud Computing. SRDS 2010, 29th IEEE International Symposium on Reliable Distributed Systems.

[5] A. Albeshri, W. Caelli. Mutual Protection in a Cloud Computing Environment. HPCC 2010, 12th IEEE International Conference on High Performance Computing and Communications.

[6] R. Ranchal, B. Bhargava, L. B. Othmane, L. Lilien, A. Kim, M. Kang, M. Linderman. Protection of Identity Information in Cloud Computing without Trusted Third Party. SRDS 2010, 29th IEEE International Symposium on Reliable Distributed Systems.

[7] R. C. C. Castro, V. L. P. Sousa. Segurança em Cloud Computing: governança e gerenciamento de riscos de segurança. Info Brasil, 2010.

[8] M. A. Armbrust et al. A view of Cloud Computing. Communications of the ACM, v. 53, no. 4, April, 2010.

[9] T. El Maliki, J. M. Seigneur. A Survey of User-centric Identity Management Technologies. In Proceedings of the The International Conference on Emerging Security Information, Systems, and Technologies, 2007.

[10]SHIBBOLETH - site oficial. Disponível em:<http://shibboleth.net/>. Acesso em: Jun. 2012.

[11]HIGGINS - Personal Data Service.Disponível em:<http://www.eclipse.org/higgins/>. Acesso em: Jun. 2012.

[12] OpenAM Project. Disponível em: <http://openam.forgerock.org/>. Acesso em: Jul. 2012.

[13] ForgeRock. Disponível em: <http://openam.forgerock.org/>. Acesso em: Jul. 2012.

[14] K. J. Hole, E. Dyrnes, P. Thorsheim. Securing Wi-fi networks - Captive Portals. IEEE Computer Society, v. 38, pages 28-34, July, 2005.

[15] L. G. Machado. "CPAut" Uma Arquitetura de Controle de Acesso para o CRSPE/INPE - MCT. Trabalho de Graduação apresentado ao Curso de Graduação em Ciência da Computação Bacharelado, da Universidade Federal de Santa Maria (UFSM, RS),2006.