DOI: 10.5748/9788599693131-14CONTECSI/RF-4954

IMPLEMENTATION OF A SINGLE SIGN-ON SOLUTION FOR IMPROVED PASSWORD MANAGEMENT IN CORPORATE ENVIRONMENTS

Dante Minucci (Institute for Technological Research of the State of São Paulo, São Paulo, Brasil) - dminucci@gmail.com
Eduardo Takeo Ueda (Institute for Technological Research of the State of São Paulo, São Paulo, Brasil) - edutakeo@usp.br

Application service providers have provided a way for other Web applications to achieve Web Single Sign-On (SSO) accepting user credentials from their domain, browser-supported login solutions and browser add-on also offer Web SSO when visiting relying parties. This work investigates the challenges and concerns corporations face when adopting SSO because many Web applications they use do not support Web SSO protocols and identify what changes in the login flow Web SSO solution. This work describes the architecture, design, and implementation of a proposed prototype SSO solution to improve the corporations' experience and adoption incentives in a way that third-party Web-based application does not need to develop chances in their system to achieve SSO. This study found out that participants were having several behaviors and concerns using the proposed SSO solution: (a) 87% expressed satisfaction when using the SSO prototype; (b) 53% prefer to use their own password management strategies instead of using SSO; (c) 80% of end-users said that the prototype demonstrated reliability and safety; (d) 73% prefer to use our proposed SSO in a single dialog window instead of keeping launching lots of Web browser. The results also show that when using the proposed prototype SSO solution user´s password reset dropped to zero, users that use the same Web applications have an average of 0,14 reset required. This study involves a contribution to the SSO authentication process without changing websites, enhancing user´s secure password management.

Keywords: Information Security, Authentication, Password, Web Components, Systems Automation.

1. Introdução

A utilização de sistemas com autenticação de acesso para autorizar a navegação faz com que os usuários tenham que memorizar diversas senhas, seguindo regras estabelecidas para criá-las. Assim, o usuário é o responsável pela escolha de cada uma de suas senhas de acesso. Narayanan (2005) afirma que senhas sempre serão vulneráveis a ataques inteligentes enquanto seus proprietários precisarem memorizá-las.

A quantidade de sistemas faz com que o usuário tenha que gerenciar diversas informações de maneira independente, pois os dados podem estar armazenados em repositórios distintos. Portais Web concentram serviços dentro de sua própria plataforma (Sun, 2009). Para utilizar diferentes portais, o usuário precisa manter cópias separadas da identidade, das relações sociais e das políticas de controle de acesso. Desta maneira, o usuário precisará efetuar manualmente o processo de autenticação em cada um dos portais que acessar. Se o sistema possui controle de sessão, essa ação poderá ser realizada mais de uma vez ao longo de um dia.

Muitas aplicações podem não oferecer alternativa para o compartilhamento de dados ou serviços de integração. Uma aplicação que controla seu próprio conjunto de usuários e emprega mecanismo de controle de acesso para proteger o conteúdo pessoal é denominado "jardim murado" (Mostarda, 2009).

Sistemas de conexão única (SSO – *Single Sign-On*) são serviços que fornecem autenticação para vários portais. A dificuldade de compartilhar e integrar conteúdo entre sistemas independentes gera problemas para a criação de uma autenticação única entre as aplicações. Os usuários autenticam em um provedor de SSO, que verifica suas credenciais retransmitindo o resultado ao *site* ao qual foi feita a requisição (Stobert, 2014).

O mecanismo de SSO pode ser usado para aplicações geridas por diversas áreas ou fornecedores, em ambientes fisicamente e logicamente segregados, Intranet, Extranet ou Internet. Os proprietários destes sistemas podem não utilizar protocolos de autenticação ou não fornecer APIs (*Application Programming Interfaces*) para acesso aos dados dos usuários, não permitindo integração ou reutilização de recursos de forma fácil de diferentes domínios de aplicação (Geambasu, 2008).

Usuários podem adotar estratégias não seguras para criação de senhas ao escolher combinações de fácil memorização, e que são facilmente decifradas por ataques ao reutilizar as mesmas senhas em diversos portais e ao adotar formas não seguras para lembrete das senhas, como, por exemplo, anotações em cadernos e blocos de notas (Cahill, 2011).

Ações dos usuários na tentativa de resgatar ou acertar uma credencial pode gerar problemas como o bloqueio do acesso ao sistema, tendo sucessivas digitações inválidas. Como consequência, pode ser necessária a ação de desbloqueio do acesso, o envio de uma confirmação positiva para troca da senha ou até mesmo ações de outros grupos externos responsáveis pelas aplicações.

Surge, portanto, a necessidade de tratar o problema de conexão única em portais que

utilizam navegador Web e não estão integrados com SSO, sem a necessidade de alterar o código fonte destes sistemas.

O objetivo deste trabalho é propor e desenvolver um protótipo de sistema para autenticar usuários de portais Web com uma única credencial pessoal. Como premissa o usuário cadastrará no protótipo cada uma de suas credenciais com senha, criando relacionamento com os portais Web. Com os dados pessoais dos usuários armazenados no banco de dados do protótipo, a recuperação das informações e a autenticação são realizadas toda vez que o usuário solicitar.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 define os mecanismos de Autenticação e *Single Sign-On*; a Seção 3 apresenta os requisitos e arquitetura da solução *Single Sign-On* proposta; a Seção 4 aborta aspectos importantes da implementação da solução proposta; a Seção 5 discute a avaliação da aplicação da solução deste trabalho em um ambiente corporativo; e por fim, a seção 6 apresenta as conclusões e possibilidades de trabalhos futuros.

2. Autenticação e Single Sign-On

Mustafic (2011) afirma que a autenticação é um processo de confirmação da identidade de uma pessoa, com isto é possível responder: "Quem é a pessoa?" ou "É a pessoa a que ele/ ela afirma ser?". Em relação ao modo como uma pessoa informa a sua identidade, existem três classes de autenticação e estas são baseadas nos seguintes fatores: (1) Conhecimento (*PIN*, senha), o que a pessoa sabe; (2) Propriedade (*Smart Card*, *Key*), o que a pessoa tem; (3) Biometria (impressão digital, íris, voz), o que a pessoa é.

Radha (2012) realizou um estudo referente às técnicas de SSO (*Single Sign-On*), que contextualiza e apresenta os conceitos necessários para o entendimento dos pormenores deste trabalho. Ele define SSO como um mecanismo que utiliza uma única ação de autenticação, para permitir que um usuário autorizado acesse todos os sistemas de software ou aplicações relacionadas, mas independentes, sem ter que efetuar uma nova autenticação durante uma sessão particular.

2.1. Autenticação Baseada em Conhecimento

Suoranta (2014) aborda sistemas Web com SSO que permitem que os usuários se autentiquem a vários aplicativos *online* de uma única vez e com uma única credencial. Utilizando o gerenciamento de identidades federadas (FIM – *Federated Identity Management*), os provedores dos serviços de autenticação (SP) delegam a função de autenticar o usuário a um provedor de identificação (IdP), também denominado provedor de serviço de identificação. Os serviços e o provedor de identificação formam uma federação. Os exemplos apresentados são elaborados a partir do projeto Shibboleth SSO. O Shibboleth (ShibbolethConsortium, 2017), desenvolvido pela Internet2, oferece um sistema de código aberto que usa mensagens SAML (*Security Assertion Markup Language*) para a implementação do IdP e do SP. Ele exige que os IdPs e SPs formem uma federação antes que o IdP possa autenticar os usuários nos aplicativos desejados.

Stobert (2014) diz que os problemas dos usuários com as senhas são bem conhecidos: aquelas que são seguras tornam-se difíceis de lembrar, os usuários possuem muitas senhas

para gerenciar e têm dificuldades para lembrar a combinação, contas de acesso e respectivas senhas. A autora desenvolveu um protótipo que incorpora elementos chave de gerenciadores de senha e palavras-passe gráficas. Esta solução evita problemas existentes com a memorização da senha e sua associação à conta do usuário. Em vez de lembrar senhas, o usuário lembra pistas na imagem que é apresentada durante o acesso ao sistema, são senhas gráficas. Estas pistas ajudam os usuários a melhor lembrar suas senhas, conectando mais facilmente as senhas e as contas. Também facilita a reutilização de senha segura, permitindo que os usuários usem uma mesma sugestão de imagem para várias contas.

2.2. Autenticação Baseada em Propriedade

Stajano (2011) diz que do ponto de vista da usabilidade, senhas e números de identificação pessoal (PIN - *Personal Identification Number*), alcançaram o fim da sua vida útil, mas se contradiz afirmando que não se pode abandoná-los até que se apresente um método alternativo de autenticação do usuário que seja útil e seguro. Ele também afirma que mesmo sendo conveniente aos desenvolvedores e arquitetos a implementação de rotinas de autenticação, o aumento de senhas gera para os usuários uma situação cada vez mais sem controle. As exigências de combinações para formar uma senha não são mais razoáveis quando o usuário tem de gerir dezenas delas.

Stajano apresentou um projeto chamado Pico, onde foi proposto um sistema alternativo baseado em hardware, um *token*, que alivia o usuário da tarefa de lembrar suas senhas e PINs. Assim um aplicativo com o qual o usuário pode se autenticar continuamente. O *token* tem dois botões importantes chamados "principal" e "emparelhamento", um pequeno visor, uma câmera 2D e uma interface de rádio bidirecional de curto alcance. Entre o sistema e o aplicativo existe comunicação sem fio criptografada. A autenticação é realizada durante o emparelhamento do aplicativo com o *token* e a autenticação nos demais aplicativos não requer senha. A solução apresentada reconhece o sistema por meio da câmera e inicia troca de chaves para validação de autenticidade; quando comprovada, libera o acesso.

Cahill (2011) afirma que a autenticação com senha por si só não garante que a pessoa com quem o aplicativo está se comunicando é o usuário legítimo. O fato de digitar a senha correta não comprova isto. O acesso a informação de usuário e senha, de forma consensual ou não, pode fazer com que qualquer pessoa se passe por outra durante o processo de autenticação sistêmica.

De acordo com Cahill a autenticação sofre problemas não resolvidos em segurança e usabilidade. No primeiro existem ocorrências de ataque para roubar credenciais de usuário, incluindo: "*phishing"*, com o formulário de senha vulnerável a ataques, o usuário pode ser enganado ao acessar um *site* que imita o original, e digitar uma senha (Yee, 2006); "*malware"*, uma coleção de vírus e muitos sistemas maliciosos que podem recuperar senhas armazenadas ou interceptar ao serem digitadas, danificar programas ou o sistema operacional (Sahu, 2014); e ataques contra prestadores de serviços. Medidas práticas de segurança com utilização de políticas de formatação de senha e o tempo limite de uma sessão ociosa muitas vezes comprometem a usabilidade. Com isso, Cahill criou um sistema de autenticação baseado no microcomputador, onde é instalado um hardware seguro que contém um repositório para armazenar as credenciais do usuário. A autenticação pode ser

realizada em prestadores de serviços locais e remotos, sem liberar as credenciais do usuário. Utiliza SAML para troca de mensagens entre os aplicativos e o sistema criado.

2.3. Autenticação Baseada em Biometria

Mustafic (2011) adapta soluções de SSO existentes, combinando múltiplos fatores comportamentais de autenticação baseados em biometria. No artigo ele apresenta a concepção de uma conexão única persistente (PSSO – *Persistent Single Sign-On*) para ambientes domésticos onipresentes, envolvendo as capacidades de comportamento biométrico para verificar a identidade do usuário continuamente, porém de uma forma discreta. Mustafic afirma que para simplificar a complexidade de gerenciar várias contas com credenciais diferentes, soluções de SSO foram introduzidas. No entanto, uma única senha para várias contas representa um único ponto de falha. Além disso, a sessão SSO, uma vez iniciada, pode representar um risco potencial quando a estação de trabalho, ou microcomputador, é abandonada sem que o usuário realize o bloqueio da sessão.

No projeto, Mustafic utilizou o conceito de autenticação forte em provedores de conexão única (SASSO – *Strong Authentication for Single Sign-On*). A ideia principal do SASSO é a utilização de um telefone celular para realização do SSO, sem a necessidade de instalação de hardware no microcomputador. A solução segue uma abordagem baseada em padrões com SAML 2.0. O perfil SAML Web SSO é limitado a aplicativos baseados em navegador e não se destina a SSO em ambientes heterogêneos, assim o SASSO implementa o provedor de identidade (IdP) diretamente no telefone móvel, o que não impede problemas de uso indevido, ataques e mesmo a dependência de um dispositivo externo. O sistema usou como fator de verificação a dinâmica de digitação do usuário, sendo esta a ação utilizada para provar que o usuário que utiliza o sistema é o mesmo desde o início da sessão. Quando o sistema detecta que o usuário mudou, inicia uma chamada no SAML para realizar a desconexão de todo o sistema (SLO – *Single Logout*). Como restrição, esta proposta funciona apenas em sistemas que utilizam digitação.

3. Arquitetura da Solução

Esta seção apresenta os requisitos e a arquitetura do protótipo para automatização da autenticação em portais Web. O método escolhido para especificação foi o NDT (*Navigational Development Techniques*), pois ele captura requisitos específicos que compõe o ambiente de sistemas Web.

A Figura 1 ilustra o fluxo do processo NDT que inicia com a definição dos objetivos, para então os requisitos serem coletados e definidos. Os requisitos são classificados de acordo com sua natureza: requisitos de armazenamento; requisitos dos atores; requisitos funcionais; requisitos de interação; requisitos não funcionais. Após elaboração dos requisitos, o fluxo do NDT continua com a definição dos modelos de conteúdo, com diagrama de classes, do modelo de interação, que mostra como os usuários podem navegar pelo sistema e o modelo de interface, que mostra a interface abstrata do sistema. Por último, é elaborada a prototipagem para validação dos requisitos.

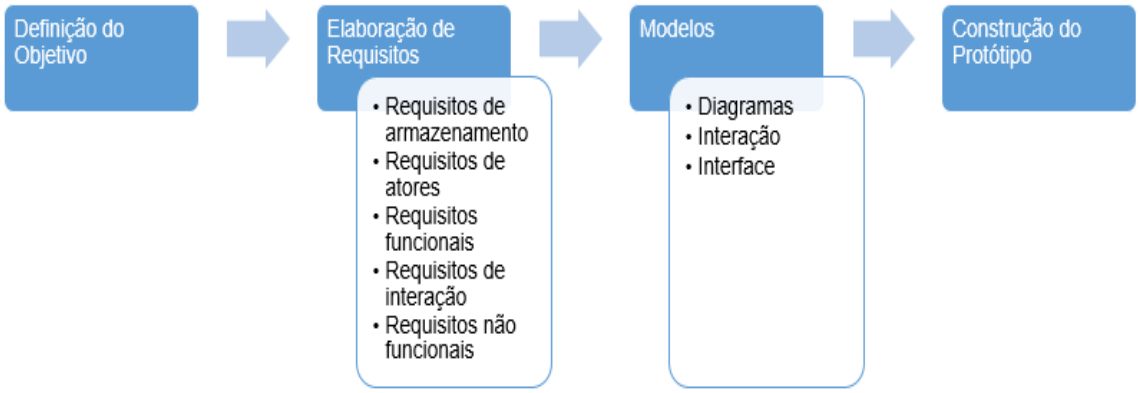

Figura 1 - Fluxo do Processo NDT adaptado de Escalona (2008).

Os objetivos (OBJ), segundo o NDT, cobrem a captura, a definição e a validação dos requisitos. A Tabela 1 a seguir apresenta a descrição dos objetivos do protótipo.

Tabela 1 - Objetivos do Protótipo (Elaborado pelos Autores)

OBJ.01	Autenticação
Descrição	Autenticar os usuários usando um provedor de identidade.

OBJ.02	Autorização
Descrição	Criar perfil de acesso para administrador e usuário.

OBJ.03	Administração
Descrição	Gerenciar usuários e gerar relatórios.

OBJ.04	Segurança
Descrição	Proteger os dados dos usuários contra acesso não autorizado.

OBJ.05	Implantação
Descrição	O Protótipo ser integrado a ambientes corporativos – ESSO (*Enterprise Single Sign-On*).

OBJ.06	Automatização
Descrição	Integrado a portais Web autorizados, automatizando a autenticação dos usuários.

3.1. Requisitos

Sommerville (2011) define os requisitos de armazenamento (RA) como os dados que devem ser mantidos em um local seguro e que pode ser gerenciado, sendo este acessível a todos os envolvidos no processo de engenharia de requisitos. A Tabela 2 a seguir descreve os requisitos de armazenamento que adotados para o armazenamento de informações do protótipo.

Tabela 2 - Requisitos de Armazenamento (Elaborado pelos Autores)

RA.01	Usuário
Objetivo associado: - OBJ.01 - OBJ.02	Armazenar as informações de: - Identificador do usuário (*Login* no Provedor de Identidade); - Identificador de perfil de acesso (administrador ou usuário); - Nome do usuário;

	- Situação (ativo ou inativo).

RA.02	Sistemas
Objetivo associado: - OBJ.03	Armazenar as informações de: - Identificador do portal Web; - URL do portal Web.

RA.03	Credenciais
Objetivo associado: - OBJ.01 - OBJ.02 - OBJ.04 - OBJ.05	Armazenar as informações de: - Identificador do usuário; - Identificador do portal Web; - Credencial de acesso (usuário e senha); - Data de criação e expiração da credencial.

RA.04	Histórico
Objetivo Associado: - OBJ.03 - OBJ.04	Armazenar as informações de: - Identificador do usuário; - Data e hora do acesso; - Ação executada.

Sommerville (2011) define os Atores (AT) como qualquer entidade que interage com o sistema e assim pode ser identificado. A Tabela 3 a seguir descreve os requisitos dos atores do Protótipo. Cada requisito descrito refere-se a um ator identificado.

Tabela 3 - Requisitos dos Atores (Elaborado pelos Autores)

AT.01	Usuário (Administrador e Operador)
Objetivo associado: - OBJ.01 - OBJ.02 - OBJ.03 - OBJ.06	Aquele que se autentica e utiliza as funcionalidades do protótipo.

AT.02	Provedor de serviço
Objetivo associado: - OBJ.02 - OBJ.04	Provedor de serviço Web que transporta as informações armazenadas no protótipo, responsável pelos módulos Serviço e Integrador do protótipo.

AT.03	Provedor de identidade
Objetivo associado: - OBJ.01 - OBJ.04	Provedor de identidade que garante a autenticidade do usuário que está acessando o protótipo.

Os requisitos funcionais (RF) de um sistema descrevem o que este deve fazer, a função do sistema em detalhes, suas entradas e saídas, exceções e assim por diante. Estes requisitos dependem do tipo de *software* que está sendo desenvolvido, do tipo de usuários que o utilizará e da abordagem adotada pela organização (Sommerville, 2011). A Tabela 4 a seguir descreve os requisitos funcionais do protótipo e contém a descrição dos casos de uso considerados para implementação.

Tabela 4 - Requisitos Funcionais (Elaborado pelos Autores)

RF.01	Autenticar
Objetivo associado:	Usuário informa sua credencial para autenticar no protótipo que

- OBJ.01 - OBJ.04	consulta o Provedor de Identidade para comprovar sua autenticidade.
Ator Associado: - AT.01 (Usuário: Operador) - AT.03 (Provedor de Identidade)	

RF.02	Consultar
Objetivo associado: - OBJ.03	Usuário consulta os dados armazenados no sistema, conforme critério informado pelo administrador, as informações são transportadas pelo provedor de serviço.
Ator Associado: - AT.01 (Usuário: Administrador) - AT.02 (Provedor de Serviço)	

RF.03	Gerenciar
Objetivo associado: - OBJ.03 - OBJ.04	Administrador seleciona ação cadastral para incluir, alterar ou revogar usuários ou portais Web.
Ator Associado: - AT.01 (Usuário: Administrador) - AT.02 (Provedor de Serviço)	

RF.04	Autorizar
Objetivo associado: - OBJ.01 - OBJ.04	Provedor de Serviço consulta dados armazenados e recebe lista de portais Web autorizados para o usuário.
Ator Associado: - AT.02 (Provedor de Serviço)	

RF.05	Acessar
Objetivo associado: - OBJ.04 - OBJ.06	Provedor de Serviço recebe o portal Web que o usuário deseja acessar, o protótipo realiza a abertura do navegador e realiza a autenticação automatizada.
Ator Associado: - AT.01 (Usuário: Operador) - AT.02 (Provedor de Serviço)	

Escalona (2008) diz que os requisitos interativos (RI) devem ter definidos seus critérios de acesso aos dados, recuperação ou pesquisa. Estes requisitos também definem como os dados serão disponibilizados em tela ao usuário e como será sua navegação. A Tabela 5 descreve os requisitos interativos do protótipo. Estes requisitos contêm as informações que serão recuperadas e disponibilizadas em tela para os usuários do protótipo e como estes navegam de uma funcionalidade para outra.

Tabela 5 - Requisitos Interativos (Elaborado pelos Autores)

RI.01	Recuperação de portais
Objetivo associado: - OBJ.03 - OBJ.06	Protótipo mostra os portais que o usuário pode acessar.
Ator Associado: - AT.01 (Usuário: Operador)	

RI.02	Recuperação de credenciais
Objetivo associado: - OBJ.06	Protótipo recupera as informações do endereço (URL), do usuário e da senha do usuário e mostra o portal autenticado.
Ator Associado: - AT.01 (Usuário: Operador) - AT.02 (Provedor de Serviço)	

RI.03	Recuperação de dados por tipo
Objetivo associado: - OBJ.03	Protótipo disponibiliza dos dados armazenados no sistema seguindo o critério de seleção utilizado.
Ator Associado: - AT.01 (Usuário: Administrador) - AT.02 (Provedor de Serviço)	

RI.04	Informação de portais
Objetivo associado: - OBJ.03	Mostra os endereços (URL) de cada um dos portais.
Ator Associado: - AT.01 (Usuário: Administrador) - AT.02 (Provedor de Serviço)	

RI-05	Informação de usuários
Objetivo associado: - OBJ-03	Mostra informações cadastrais dos usuários e perfil de acesso.
Ator Associado: - AT.01 (Usuário: Administrador) - AT.02 (Provedor de Serviço)	

RI.06	Informação histórica
Objetivo associado: - OBJ.04	Mostra a rastreabilidade da ação realizada pelo usuário.
Ator Associado: - AT.01 (Usuário: Administrador) - AT.02 (Provedor de Serviço)	

Como o nome sugere, os requisitos não funcionais (RNF) não estão diretamente relacionados com as funções específicas que o sistema deve fazer. Eles podem estar relacionados com as propriedades do sistema, tais como confiabilidade, tempo de resposta e pico de ocupação (Sommerville, 2011). A Tabela 6 a seguir atribui para cada requisito não funcional identificado um nome e uma descrição.

Tabela 6 - Requisitos Não Funcionais (Elaborado pelos Autores)

RNF.01	Confidencialidade e integridade
Objetivo associado: - OBJ.04	Assegurar que os dados armazenados no protótipo são protegidos e acessados somente pelo usuário autorizado.

RNF.02	Autorização
Objetivo associado: - OBJ.02	Toda requisição realizada no protótipo deve ser autorizada pelo usuário.

RNF.03	Navegador Web

Objetivo associado: - OBJ.06	Deverá realizar a automatização do acesso utilizando o navegador embarcado no protótipo.

RNF.04	Disponibilidade
Objetivo associado: - OBJ.04 - OBJ.05	Considerar que os serviços utilizados no protótipo estejam disponíveis para uso quando requisitados.

RNF.05	Escalabilidade
Objetivo associado: - OBJ.04 - OBJ.05	Garantir o crescimento do desempenho do protótipo quando receber uma quantidade maior de requisições.

3.2. Visão Geral da Arquitetura

A arquitetura do protótipo é baseada no Microsoft Windows e seu ambiente de desenvolvimento integrado, IDE – *Integrated Development Environment*, uma vez que o objetivo deste trabalho é a implementação em ambiente corporativo e dados de janeiro de 2016 do *Net Market Share* (Netmarketshare, 2017) apontam que 90,60% dos *desktops* rodam o sistema operacional Windows.

A arquitetura do protótipo deste trabalho pode ser separada em quatro grupos principais: Apresentação; Provedor de Serviço; Provedor de Identidade; Repositório de Dados. Por meio de um diagrama, a Figura 2 ilustra uma visão geral dos componentes, camadas e relacionamentos existentes no protótipo.

No primeiro grupo, Apresentação, o usuário (operador ou administrador) acessa o protótipo. Neste estão inseridos itens como o sistema operacional e a camada de apresentação. A aplicação do usuário – sistema com o qual o usuário realiza o acesso e autoriza as funcionalidades conforme perfil de acesso – está neste grupo. Conforme perfil cadastrado na aplicação o usuário poderá acessar funcionalidades administrativas ou de operador para acesso automatizado aos portais Web que possui autorização (SSO).

O segundo grupo, Provedor de Serviço, é o local que hospeda o portal de acesso do Administrador, o serviço que executa as transações para acesso aos recursos protegidos do protótipo, com os componentes de negócio e de persistência. O Provedor de Serviço recebe a credencial do usuário que está solicitando autenticação no protótipo, como este serviço está registrado no Provedor de Identificação, ele verifica sua autenticidade e permite o acesso aos serviços. Os dicionários e funções para ler e escrever informações nos repositórios de dados são fornecidos por este serviço.

Figura 2 - Arquitetura do Protótipo (Elaborado pelos Autores)

Por sua vez, o terceiro grupo, Provedor de Identidade, é o responsável por verificar se o usuário que está solicitando acesso é legítimo e possui autorização para ingressar no domínio e ter concedido o acesso às informações. Por último, o Repositório de Dados, armazena as informações administrativas, individuais e de acesso ao protótipo.

O diagrama de sequência da Figura 3 ilustra a interação das camadas que compõe o protótipo, cada camada fornece serviços para a camada anterior e utiliza os serviços da camada subsequente.

Figura 3 - Interação das Camadas do Protótipo (Elaborado pelos Autores)

A construção em camadas proporciona sistemas mais modulares e flexíveis, divide a funcionalidade de manutenção e apresentação dos dados para minimizar o grau de acoplamento entre os objetos e esta redução entre as classes do sistema pode facilitar as atividades de manutenção (Gui, 2009). Sendo assim, a principal característica de um sistema em camadas é a separação lógica de negócio, processamento do servidor e a interface com o usuário em componentes distintos.

A Figura 4 ilustra a interface de acesso ao protótipo, garantindo que toda requisição de autenticação no protótipo será manual, ou seja, o usuário deverá digitar sua credencial na abertura do aplicativo.

Figura 4 - Acesso ao Protótipo (Elaborado pelos Autores)

A Figura 5 ilustra a arquitetura de segurança envolvendo os componentes do protótipo e os atributos considerados, detalhes específicos de implementação são apresentados na próxima seção.

Figura 5 - Arquitetura de Segurança (Elaborado pelos Autores)

4. Implementação

Esta seção apresenta os detalhes referentes a implementação do protótipo da solução SSO em ambiente Windows, considerando os componentes Aplicação do Usuário, Provedor de Serviço, Provedor de Identidade e Repositório de Dados, apresentados na seção anterior.

O protocolo HTTPS foi utilizado na comunicação entre o Provedor de Serviço e a aplicação do usuário. O HTTPS é um canal de comunicação seguro usado para trocar informação entre o aplicativo utilizado pelo usuário e o servidor que hospeda o Provedor de Serviço e ele usa Secure Sockets Layer (SSL). Para habilitar o SSL, é necessário obter um certificado que é usado para codificar e decodificar a informação trafegada pela Intranet onde o protótipo é utilizado. O protótipo utiliza um certificado gerado por uma Autoridade Certificadora Interna (CA – *Certificate Authority*), e este é configurado no servidor Web do Provedor de Serviço e no sistema operacional dos usuários.

Para a utilização do Provedor de Serviço é necessário que o usuário solicitante tenha autorização para utilizar o Provedor de Identidade, pois o seu acesso é configurado para a realização de autenticação integrada com o domínio, como indicado na Figura 6 a seguir.

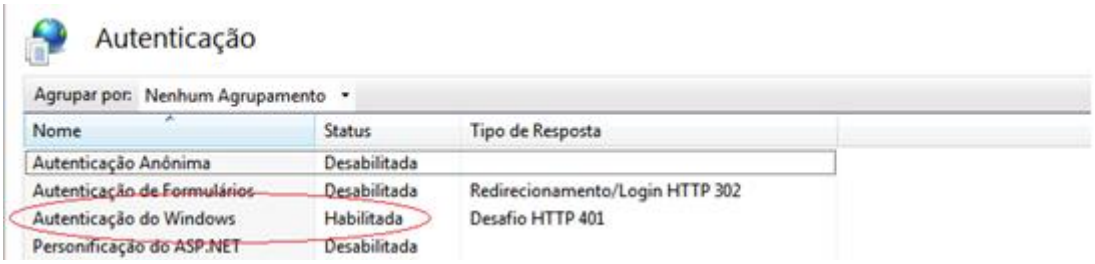

Figura 6 - Configurações do servidor de aplicação Microsoft IIS (Elaborado pelos Autores)

A utilização do Provedor de Identidade no protótipo provê uma terceira parte confiável para autenticação. O IdP utilizado no protótipo foi o *Active Directory Domain Services* (AD DS). O método para autenticação do protótipo do Usuário (Operador e Administrador) junto ao IdP foi o LDAP ("*Lightweight Directory Access Protocol*"), baseada em formulário. O usuário fornece o nome da conta e a prova de conhecimento da senha e estas informações são submetidas para validação contra a credencial que está armazenada no AD DS. A Figura 7 apresenta o código do método de chamada do protótipo ao Provedor de Identidade.

```
public string Autentica(string IpServer, string User, string Senha)
{
    string retorno = string.Empty;
    try
    {
        DirectoryEntry objAD = new DirectoryEntry("LDAP://" + IpServer, User, Senha);

        retorno = objAD.Name;
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
    return retorno;
}
```

Figura 7 - Verificação do Usuário no Provedor de Identidade (Elaborado pelos Autores)

A autenticação integrada configurada no Provedor de Serviço foi baseada em declarações no Windows, o aplicativo obtém um *token* de segurança do usuário, e não as credenciais, e usa as informações dentro das declarações para determinar o acesso aos recursos. Qualquer falha ou negativa do AD DS na etapa de autenticação, o protótipo ou o Provedor de Serviço emitirá mensagem de acesso negado, não permitindo o acesso à aplicação.

O acesso ao repositório/banco de dados que hospeda as informações dos usuários deve ser realizado e autorizado somente a partir do Provedor de Serviço, ou seja, apenas o IP e *Hostname* deste componente poderá estabelecer conexão com o servidor que hospeda o banco de dados. Não deve ser estabelecida comunicação direta da aplicação do usuário com o banco de dados. A Figura 8 apresenta o código do método de comunicação do *Web Service* com o componente de repositório de dados com criptografia da informação.

```
SqlConnection con;
private SqlConnection GetConnection()
{
    if (con == null || (con.State == ConnectionState.Closed))
    {
        Crypt crypto = new Crypt();

        string ConnectionString = crypto.DeCryptBanco(ConfigurationManager.ConnectionStrings["ConnectionStringSql"].ConnectionString);
        con = new SqlConnection();
        con.ConnectionString = ConnectionString;
        con.Open();
    }

    return con;
}

private void CloseConnection(SqlConnection con)
{
    try
    {
        if (con.State == ConnectionState.Open)
        {
            con.Close();
        }
    }
    catch
    { }
    finally
    {
```

Figura 8 – Comunicação entre Web Service e Banco de Dados (Elaborado pelos Autores)

A criptografia para entrada e saída de dados do Provedor de Serviço para o Banco de

Dados utilizou duas chaves. Para a comunicação entre estes dois servidores foram utilizadas chaves fixas, uma delas permaneceu compilada no aplicativo em execução no servidor *do* Provedor de Serviço e a outra hospedada no "*Service Provider"* do mesmo servidor, marcada como não exportável.

E foi utilizada criptografia simétrica do tipo AES com 256 bytes. As informações antes de serem armazenadas no banco de dados de forma criptografada são novamente criptografadas, porém utilizando uma chave simétrica hospedada no próprio banco de dados, protegida por senha e certificado, com algoritmo AES 256.

Para assegurar a origem dos acessos, foi desenvolvido no serviço de consulta uma "*White List*", ou lista de endereçamento de máquinas, para que apenas determinados intervalos de IPs, ou mesmos IPs fixos, tenham acesso ao Provedor de Serviço.

5. Avaliação

Esta seção apresenta a avaliação de aceitação da solução implementada em um ambiente corporativo. A primeira etapa formou um grupo focal para compreender os desafios, preocupações e benefícios percebidos com a utilização de sistemas com SSO, assim como realizado por Sun (2013). A segunda etapa da avaliação foi um levantamento (*Survey*) aplicado aos usuários do protótipo.

Foi formado um grupo focal com 9 (nove) participantes, sendo 5 (cinco) funcionários das áreas de tecnologia e 4 (quatro) da área operacional de uma empresa que terceiriza processos de negócio (*Business Process Outsourcing*).

Os participantes do grupo foram submetidos a utilizar dois sistemas comuns a todos, o Web e-mail e o sistema de abertura de chamados para problemas administrativos e sistêmicos. Inicialmente os participantes realizaram o acesso aos portais digitando manualmente suas credenciais de acesso, ao final desta tarefa executaram o encerramento da sessão (*logoff*). Finalizado tal procedimento, os participantes foram submetidos a utilizar o protótipo desenvolvido neste trabalho. Inicialmente informaram as credenciais para que o sistema pudesse conduzir a autenticação automatizada. Após a conclusão da atividade, os participantes foram submetidos a uma entrevista contextual, a fim de entender os problemas encontrados, bem como as suas potenciais preocupações e benefícios percebidos, detalhando assim a experiência de utilizar os sistemas com e sem o a solução SSO. Com os aspectos positivos e negativos capturados foi possível elaborar um *Survey* com 5 questões para aplicar em um grupo maior de usuários.

Na segunda etapa da avaliação o *Survey* foi aplicado, por meio de questionário estruturado, para um grupo de 397 usuários que utilizaram o protótipo e possuíam experiência em sistemas com autenticação. Outro grupo com 397 usuários tiveram seu comportamento observado quando a troca de senhas, nos mesmos sistemas, mas sem a utilização da solução SSO. Ambos os grupos foram observados durante 30 dias, mas somente o grupo que utilizou o protótipo respondeu o formulário de levantamento, uma vez que as questões eram referentes ao protótipo.

A Tabela 7 apresenta a quantidade de pedido de troca de senha registrado durante os três meses de observação do grupo de 397 usuários que não utilizaram a solução SSO.

Tabela 7 - Quantidade de Pedidos de Troca de Senha do Grupo sem SSO

Mês	Usuários	Pedidos	Percentual
1	397	85	21%
2	397	43	11%
3	397	36	9%
Média	397	55	14%

Os participantes do grupo que utilizou o protótipo da solução SSO não registraram nenhum pedido de troca de senha por esquecimento ou ter excedido ou violado regras de tentativas inválidas, estes dados foram obtidos com os administradores do sistema que realizavam a gestão de acessos e confirmados por não ter nenhum registro de alteração de senha na base de dados.

Os resultados observados com a aplicação do *Survey* no grupo que utilizou a solução SSO sugerem que os usuários de portais Web corporativos consideraram sua experiência com o mecanismo de SSO adotado no protótipo, como a abertura e autenticação sem precisar digitar as credenciais, valiosa e amigável. As respostas obtidas na Questão 1 apresentaram percentuais de pessoas satisfeitas (53%) ou muito satisfeitas (34%) superiores aos de rejeição (2%) ou indiferença (12%).

Apesar da maioria ter gostado da experiência com o SSO apresentado no protótipo, quando os participantes foram questionados se preferem utilizar o sistema apresentado para controlar suas credenciais ou realizar este controle pessoalmente, as respostas indicaram que a maioria dos participantes (Questão 2 – 53%) prefere o acesso pelo protótipo, porém a quantidade de pessoas que ainda desejam controlar seus usuários e senhas manualmente é alta (47%).

Ao serem questionados sobre requisitos de segurança apresentados no protótipo (Questão 3) como a confidencialidade e integridade das informações, disponibilidade do sistema e políticas de segurança para impedir o acesso não autorizado, a percepção de segurança dos participantes muito satisfeitos (48%) e satisfeitos (32%) foi superior a quantidade de pessoas que declararam indiferença quanto ao uso e sua segurança (19%). Os usuários do protótipo insatisfeitos e muito insatisfeitos somaram 4 pessoas.

Foi observado também que a maioria dos usuários (Questão 4 – 73%) responderam que preferem e sentiram maior segurança no modelo de abertura dos portais Web corporativos dentro do próprio protótipo. Os participantes que declararam preferência por acessar os sistemas Web em diversas janelas do browser sem o protótipo somaram 27% da população alvo do *Survey*.

Na última questão apresentada (Questão 5), os participantes foram informados que nenhum portal Web corporativo foi alterado e o protótipo realizou três ações para eles: abertura do portal Web; preenchimento do usuário e senha; envio da requisição de autenticação para acessar o portal Web corporativo. As respostas obtidas sugerem que a proposta do sistema apresentado aos participantes do *Survey* foi satisfatória, pois apresentou percentuais de pessoas satisfeitas (51%) ou muito satisfeitas (38%) superiores aos de rejeição (2%) ou indiferença (11%).

6. Conclusão e Trabalhos Futuros

Por meio de uma prova de conceito este trabalho mostrou que é possível, considerando uma solução SSO (*Single Sign-On*), garantir a viabilidade técnica da implementação de SSO em portais Web corporativos não integrados a provedores de identidade e assim verificar o cumprimento dos requisitos especificados na Seção 3.

Muitos portais possuem verificações durante a etapa de autenticação, solicitando ao usuário requisitante uma contraprova além da credencial informada no formulário de senha. Nossa proposta não abrange o escopo de portais que utilizam recursos e testes de *Captcha* ou formulários de senha com imagens.

A implementação de *Single Sign-On* em sistemas Web e o gerenciamento de recursos dos usuários possuem grande potencial para o desenvolvimento de trabalhos futuros. Como recomendações de continuidade desta pesquisa destacamos:

(a) Desenvolvimento de um *framework* para a inclusão de novos portais Web, sem a necessidade da intervenção de profissionais com conhecimento em linguagem de programação;
(b) Utilização de um padrão de interface binária para o *browser* permitir o acoplamento entre aplicações nos diversos processos dos sistemas Web, não apenas na etapa de autenticação;
(c) Uso de componentes compatíveis para que o modelo proposto não seja limitado somente a portais Web, mas que também possa ser compatível com sistemas executáveis (Windows Forms);
(d) Criação de *White List* de acesso de usuários, assim, além da validação do usuário origem o sistema poderá bloquear acessos a partir de locais não autorizados.

Referências

Cahill C.P., Martin J., Phegade V., Rajan A., Pagano M.W. Client-based authentication technology: user-centric authentication using secure containers. In: DIM'11: proceedings of the 7th ACM workshop on Digital Identity Management, p.83-92, ACM, 2011.

Escalona, J. M.; Aragon, G. NDT - A Model-Driven Approach for Web Requirements. Software Engineering, IEEE Transactions on, v.34, n.3, p.377-390, jun. 2008.

Geambasu, R. Cheung, C; Moshchuk, A; Gribble, S.D.; Levy, H. M. Organizing and Sharing Distributed Personal Web-Service Data. 17TH International Conference on World Wild Web. Proceedings..., WWW 2008. New York, USA: ACM, 2008.

Gui, G; Scott, P. D. Measuring Software Component Reusability by Coupling and Cohesion Metrics. Journal of Computers, v. 4, n. 9, p. 18–21, 2009.
Mostarda, M.; Palmisano, D.; Zani, F.; TripodI, S. Towards an OpenID-based solution to the Social Network Interoperability Problem. In W3C Workshop on the Future of Social Networking. p.15-16, jan. 2009.

Mustafic, T.; Messerman, A.; Camtepe, S. A.; Schmidt, A. D.; Albayrak, S. Behavioral

biometrics for persistent single sign-on. In DIM'11 Proceedings of the 7th ACM workshop on Digital Identity Management. ACM, 2011. p.73-82.

Narayanan, A.; Shmatikov, V. Fast dictionary attacks on passwords using time-space tradeoff. In Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS 2005), EUA: ACM, 2005.

Netmarketshare. Net Market Share. http://goo.gl/UNknh6. Acessado em: 21/02/2017.

Radha, V.; Reddya, D. H. A Survey on Single Sign-On Techniques. 2nd International Conference on Computer, Communication, Control and Information Technology (C3IT-2012), Procedia Technology, v.4, p.134–139, Elselvier, fev. 2012.

Sommerville, I. Software Engineering. Addison-Wesley Publishers Limited. 9. ed. Londres, 2011. 840p.

Sahu, M. K.; Ahirwar, M.; Hemlata, A. A Review of Malware Detection Based on Pattern Matching Technique. IJCSIT INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGIES, v.5(1), p.944-947, 2014.

Shibboleth Consorthium. Shibboleth Internet 2. http://shibboleth.net/about. Acessado em: 10/02/2017.

Stajano, F. Pico: No More Passwords!*. In: CHRISTIANSON, B. et al. (Ed.). Security Protocols XIX. Heidelberger: Springer-Verlag, 2011. p.49-81.

Stobert, E. Biddle, R. A Password Manager that Doesn't Remember Passwords. In NSPW '14: Proceedings of the 2014 workshop on New Security Paradigms Workshop. ACM, set. 2014.

Sun, S. T.; Hawkey, K.; Beznosov, K. Security Web 2.0 Content Sharing Beyond Walled Gardens. Computer Security Applications Conference, 2009 (ACSAC 09). Proceedings... Honolulu, Hawaii: IEEE Computer Society, dez 2009.

Sun, S. T; Pospisil, E.; Muslukhov, I. Dindar, N. Hawkey, K. Beznosov, K. Investigating Users' Perspectives of Web Single Sign-On: Conceptual Gaps and Acceptance Model. Transactions on Internet Technology (TOIT). v.13, n.1. ACM, nov. 2013.

Suoranta, S. et al. Logout in single sign-on systems: Problems and Solutions. Journal of information security and applications. v.19, n.1, p.61-77, ACM, fev. 2014.

Yee, K. P.; Sitaker, K. Passpet: convenient password management and phishing protection. In SOUPS '06 Proceedings of the Second Symposium on Usable Privacy and Security, 2006, p.32-43, ACM, New York, NY, 2006.