Muriel Canever

Implantação de um controle centralizado de usuários utilizando o protocolo LDAP.

São José-SC

Dezembro/2016

Muriel Canever

Implantação de um controle centralizado de usuários utilizando o protocolo LDAP.

Monografia apresentada à Coordenação do Curso Superior de Tecnologia em Sistemas de Telecomunicações do Instituto Federal de Santa Catarina para a obtenção do diploma de Tecnólogo em Sistemas de Telecomunicações.

Orientador:
Prof. Luciano Barreto

Curso Superior de Tecnologia em Sistemas de Telecomunicações
Instituto Federal de Santa Catarina

São José-SC
Dezembro/2016

Monografia sob o título *"Implantação de um controle centralizado de usuários utilizando o protocolo LDAP"*, defendida por *Muriel Canever* e aprovada em Dezembro de 2016, em São José, Santa Catarina, pela banca examinadora assim constituída:

Prof. Luciano Barreto
Orientador - IFSC

Prof. Odilson Tadeu Valle
IFSC

Prof. Ederson Torresini
IFSC

Agradecimentos

Primeiramente à minha família, que sempre esteve ao meu lado. Um agradecimento em especial para minha namorada que foi uma grande incentivadora para o término deste projeto.

Quero agradecer também os meus companheiros de trabalho da empresa ServerDo.in, que disponibilizaram os servidores e serviços (sem a estrutura física nada seria possível) a serem integrados, e também colaboraram com o amadurecimento da idéia inicial do projeto.

Ao meu orientador Luciano Barreto, que soube planejar com experiência os prazos a serem cumpridos durante nossas reuniões, assim como outros professores da Instituição, Ederson Torresini, que se mostrou interessado em ajudar em algumas dificuldades do projeto.

Resumo

A metodologia se caracteriza através da apresentação de um protocolo padrão de comunicação que utiliza um serviço de diretório, através de uma abordagem teórica e técnica. A abordagem principal utiliza tanto aplicações em código aberto, como soluções proprietárias, discorrendo sobre o protocolo LDAP - *Lightweight Directory Access Protocol*, protocolo leve para acesso de diretórios. Os conceitos teóricos apresentados nos primeiros capítulos são de extrema importância para se criar uma base de entendimento inicial sobre o assunto, para posteriormente integração do ambiente de estudo e análise. O presente trabalho pretende adaptar aplicações com suporte ao protocolo LDAP, formando um gerenciamento centralizado de usuários. As aplicações irão utilizar o sistema de diretório transparentemente, mantendo a informação em um único repositório.

Esta integração das aplicações foi implementada na ServerDo.in, empresa que fornece soluções em computação em nuvem e hospedagem de sites, onde foi identificado uma carência de soluções de gerenciamento que atendessem as demandas da empresa. Para isso, a opção preferencial para este gerenciamento, foi realizada através da uma ferramenta de e-mail corporativo chamada Zimbra, que já utiliza o protocolo para gerenciamento de suas contas.

Palavras-chave: *LDAP, OpenLDAP, centralização de usuários.*

Abstract

The methodology is characterized by the presentation of a standard communication protocol that uses a directory service, through a theoretical and technical approach. The main approach uses both open source and proprietary solutions addressing the LDAP - Lightweight Directory Access Protocol. The theoretical concepts presented in the first chapters are of extreme importance in order to create an initial understanding base on the subject, to integrate the study environment and analysis later. The present work intends to adapt applications with support to the LDAP, forming a centralized management of users. The applications will use the directory system transparently, keeping the information in a unique repository.

This integration of application was implemented at ServerDo.in, a company that provides solutions in cloud computing and web hosting, where it was identified a deficiency of management solutions that met the demands of the company. For this, the preferred solution for this management was made through a corporate email tool called Zimbra, which already uses the LDAP protocol for managing its accounts.

Key-words: *LDAP, OpenLDAP, centralized user management.*

Sumário

Lista de Figuras

Lista de Abreviaturas

AWS *Amazon Web Services*

CLI *Command Line Interface*

CMS *Content Management System*

CN *Common Name*

DAP *Directory Access Protocol*

DC *Domain Control*

HTTP *Hypertext Transfer Protocol*

LDAP *Lightweight Directory Access Protocol*

NSS *Name Server Switch*

OU *Organizational Unit*

PAM *Pluggable Authentication Module*

PHP *Hypertext Preprocessor*

SASL *Simple Authentication and Security Layer*

SSH *Secure Shell*

SSL *Secure Sockets Layer*

TI *Tecnologia da informação*

UID *User Id*

VPS *Virtual Private Server*

1 Introdução

O uso e aplicação de tecnologias vêm desenvolvendo um papel importante dentro das empresas, interconectando sistemas em rede com o objetivo de oferecer uma melhor experiência e serviços aos seus clientes. As tendências para integrar as informações e os dados acerca de usuários, senhas e diretórios, surgem como uma forma de reduzir transtornos dentro dos ambientes corporativos. Soluções para prover a diminuição dos gastos utilizando serviços de código aberto, ganham espaço no mercado e são explorados por profissionais da área. O protocolo que será discutido e posteriormente implementado nas aplicações, está em destaque quando o assunto é a utilização do serviço de diretório para o gerenciamento de infraestruturas de redes. Segundo TUTTLE (et al., 2004, p.3) [5], para aumentar a funcionalidade e o uso, e permitir um custo benefício para administrar aplicações distribuídas, informações sobre os serviços, usuários, e outros objetos acessíveis vindo das aplicações, precisam ser organizadas de uma maneira simples e consistente.

Comumente usado por profissionais da área de Tecnologia da Informação, para autenticação e autorização de usuários em aplicações corporativas, o LDAP, surge a partir de uma deficiência, onde se fez necessário integrar as informações (credenciais e atributos) dos usuários, fazendo com que os dados fiquem centralizados em um único servidor base. Se tornando uma ferramenta estratégica dentro de ambientes corporativos, CARTER [30] reitera sobre o potencial do LDAP para consolidar serviços existentes em um único diretório, que pode ser acessado por clientes LDAP, sendo eles: navegadores de internet, clientes de e-mail, servidores de e-mail e uma variedade de outros aplicativos. CARTER [30] acredita que como mais e mais aplicativos usam diretórios LDAP, fazer um investimento na criação de um servidor LDAP terá um retorno financeiro (*payoff*) enorme a longo prazo.

Analisado as aplicações desejadas com suporte ao protocolo LDAP, e tendo em vista as vantagens após integração com as mesmas, decidiu-se implementar este gerenciamento utilizando o protocolo LDAP para troca de dados/informações. Isso irá unificar os esforços

de criação e manutenção da base de informações.

Vale lembrar que o conceito de uma base LDAP não é novidade no ambiente acadêmico, porém a falta de experiência em como realizar a implementação deste controle centralizado de usuários, não induz à procura desta solução de código aberto para resolver problemas de gerenciamento. Porém, conforme DONLEY (2003, p.10) [2], onde afirma já em 2003, que o LDAP é uma tecnologia madura utilizada por uma ampla variedade de aplicações para muitos fins críticos. Estas aplicações incluem diversas funcionalidades desde autenticação, autorização e gestão de usuários. Segundo HELMKE[31], se você estiver usando LDAP, estará ciente de seu imenso poder e flexibilidade, todo o trabalho duro que você colocar vale a pena, porque o funcionamento do LDAP melhora imensamente a experiência em rede. Dessa forma, novas aplicações são desenvolvidas frequentemente com suporte ao LDAP, que garantem que a utilização do mesmo como uma base centralizada de usuários que vai continuar a crescer.

1.1 Motivação

A dificuldade para gerenciar diversas credenciais (nomes de usuários e senhas) e atributos (endereço, telefone, data de nascimento) de usuários em diversos serviços, somado com a variedade e a complexidade de controles de acesso pode ser trabalhoso. Essa dificuldade foi a motivação para a realização uma solução estratégica, empregando classes e atributos para sistemas independentes usando uma base LDAP comum, ou seja, uma solução que facilita e reduz a complexidade da infraestrutura de rede, ganhando tempo e consequentemente diminuindo os custos com a gestão das informações.

1.2 Objetivos geral e específicos

Estudos para integrar e criar uma estrutura de armazenamento organizada de forma hierárquica, também é apresentada em diferentes monografias, onde fundamentam a eficácia do OpenLDAP e do protocolo LDAP, realizando integrações com diferentes serviços.

RODRIGUES(2008) [29] em sua monografia, apresenta uma integração de serviços em redes de computadores Linux, utilizando a ferramenta OpenLDAP, abordando sobre a metodologia, autenticação de usuários e serviço de diretórios, integrando ferramentas como o Postfix e Samba. Por sua vez, MACHADO (et al. 2006) [6], descrevem o processo de

integração do serviço de diretório com os serviços de autenticação para redes Linux/Unix e Windows.

Neste trabalho, o objetivo geral é fazer a integração do serviço de diretório LDAP em um conjunto de aplicações diversas, tais como o serviço de acesso remoto - SSH, o Sistema de Gerenciamento de Conteúdo (do inglês *Content Management System* – CMS) WordPress, serviço de e-mail corporativo Zimbra e a plataforma WHMCS (*Web Hosting Automation*), que faz o gerenciamento financeiro dos servidores de hospedagem de sites. Portanto, as informações (credenciais e atributos) dos usuários serão centralizadas em um sistema de diretório único, facilitando o acesso aos dados e reduzindo custos de gerenciamento, manutenção e reduzindo a possibilidade de erros.

Os objetivos específicos são:

- Estudar o serviço de diretório LDAP, assim como a ferramenta OpenLDAP e phpLDAPadmin;
- Criar o ambiente de implantação com todas as aplicações necessárias para o projeto;
- Implementar a ferramenta OpenLDAP para integrar e gerenciar os serviços;
- Testar a autenticação nas aplicações utilizando o LDAP como fonte de credenciais;
- Implantar o estudo de caso e confirmar a eficiência da integração.

1.3 Organização do texto

Para uma melhor estruturação do projeto, o presente documento se apresenta dividido em caítulos. No CAPÍTULO 2, intitulado "Fundamentação Teórica", a discussão será feita em torno das teorias e tecnologias que vamos utilizar para configurar o controle centralizado dos usuários. Os conceitos do protocolo LDAP, juntamente com as diferenças entre banco de dados relacional e de diretórios, criando um conhecimento base para entender o decorrer do projeto.

No CAPÍTULO 3, intitulado "Implementação proposta", é discutido a motivação da realização do projeto e sua proposta, trazendo as vantagens desta arquitetura que será implementada. No CAPÍTULO 4, chamado "Desenvolvimento", é apresentada a implementação prática deste trabalho, onde as aplicações e o ambiente para desenvolvimento

serão criados e configurados, trazendo os benefícios deste novo conceito implementado junto com as considerações finais.

Por fim, no CAPÍTULO 5 será abordado os possíveis trabalhos futuros a serem implementados, seguindo da conclusão geral do projeto.

2 *Fundamentação teórica*

Este capítulo tem como objetivo a descrição do protocolo padrão de diretórios, LDAP, assim como as ferramentas que foram integradas utilizando LDAP. O conjunto de aplicações que se utilizará deste serviço de diretórios fazem parte de uma infraestrutura de rede geralmente utilizado por empresas de TI. Os recursos e o desenvolvimento de toda a implementação para o controle dos dados ao servidor LDAP será apresentado, juntamente com uma base teórica, e por fim uma análise técnica ressaltando as vantagens do uso de um único servidor para gerenciar e controlar a autenticação dos usuários junto às aplicações envolvidas.

2.1 LDAP - Histórico e padrões

Segundo MENEGUITE (2009, p.14) em sua monografia *LDAP - Autenticação Centralizada*, a utilização deste serviço surgiu da necessidade de se empregar um modelo de gerenciamento de diretórios que não fosse baseado em bases de dados relacionais. O autor cita a necessidade de se "desenvolver um protocolo que tivesse a capacidade de organizar entradas em um serviço de nomes de forma hierárquica, capaz de suportar grandes quantidades de dados e com uma enorme capacidade de procura de informações". Porém, o LDAP não foi desenvolvido sem uma base inicial, ele surgiu como uma alternativa "leve"para acesso ao serviço de diretório X.500, que será abordado na próxima seção.

2.1.1 X.500 e LDAP

X.500, definido pela RFC 1487 [13], é um protocolo padrão de serviços de diretório, que deu origem ao desenvolvimento do LDAP. O CCITT (Comite Consultivo Internacional de Telefone e Telegrafia) criou o padrão X.500 em 1988. X.500 organiza entradas de diretório em um espaço de nome hierárquico capaz de suportar grandes quantidades de informação.

Ele também define uma capacidade de pesquisa poderosas para fazer a recuperação de informações mais fácil, TUTTLE (et al., 2004, p.13) [5].

TRIGO(2007) [8] relata que:

> O protocolo de acesso a diretórios (DAP) fazia parte das especificações X.500, desenvolvidas pela ITU Telecommunication. No entanto, o DAP foi baseado no modelo de referência OSI (*Open Systems Interconnection*), um modelo de transmissão de dados pré-internet que, entre outros problemas, era extremamente difícil de ser implementado corretamente, resultando em aplicações complexas e lentas. Com o advento da internet, o protocolo TCP/IP ganhou força. Por causa disso, foi criado um protocolo de acesso a diretórios que se encaixasse melhor nos moldes do TCP/IP – estava plantada a semente do LDAP.

A evolução do protocolo pode ser observada na Figura 1.

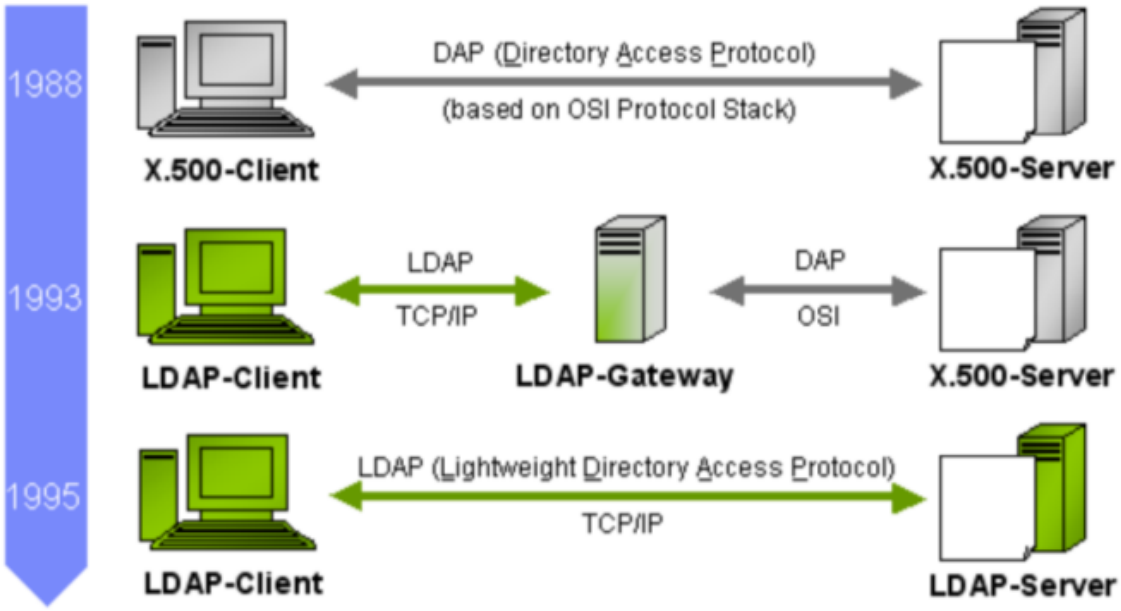


Figura 1: Evolução do DAP para LDAP - Fonte: CHAVES(2010) [9].

Portanto, a primeira ideia de servidores LDAP foi apresentada para capturar dados do X.500, mas isto evoluiu com o passar dos anos. Sobre as origens do protocolo LDAP e entendimento sobre diretórios, é primordial referenciar as definições da própria Universidade de Michigan nos Estados Unidos, onde foi desenvolvido e criado o protocolo, juntamente com as RFCs (Request for Comments, 1487 [13], 4511 [14], 2247 [15] e 2251 [16]) que fizeram parte das evoluções do LDAP. Segundo os guias da Universidade, o "LDAP foi originalmente desenvolvido como uma interface do X.500, o diretório de serviço da estru-

tura de rede de camada OSI. X.500 define o Protocolo de Acesso de Diretório (DAP) para o cliente usar quando conectar com servidores de diretório".

Conforme BUTCHER(2007) [4], a RFC 2251 [16], liberada em 1997, padronizou LDAPv3, fazendo grandes melhorias para o primeiro padrão do LDAP. O mercado de servidores LDAP foi amadurecido. Junto com o projeto OpenLDAP que foi inicializado em 1998, e pela visão de BUTCHER [4], a ferramenta OpenLDAP obteve sucesso, conseguindo avançar na área de Tecnologia, se tornando compatível para muitas distribuições Linux.

2.2 Protocolo LDAP e funcionamento

LDAP, é especificamente um serviço de diretórios baseado em X.500, rodando sobre arquitetetura TCP/IP, e baseado em um modelo cliente/servidor, como demostra a Figura 2, podendo receber uma variedade de consultas e requisições das aplicações.

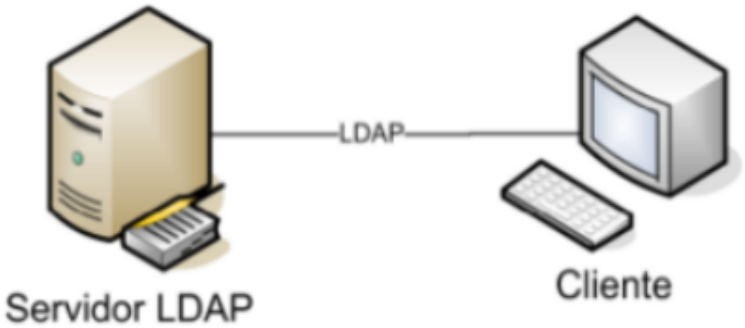


Figura 2: Modelo Cliente/Servidor. Fonte: Machado e Mori Junior (2006). [6]

A estrutura de uma árvore de diretórios LDAP, segundo GIL (2012) [32], busca organizar as informações em forma de diretório, ou seja, em forma de árvore. As partes que permitem essa formação são as especificações do protocolo, onde, baseando-se em campos, chamado de atributos, e em seus conjuntos chamados de schemas, é possível armazenar qualquer tipo de informação de forma estruturada.

Cada entrada de informação, estará retida a uma hierarquia de armazenamento dos dados na base LDAP. É necessário que se crie uma estrutura organizada na árvore, ou seja, para se habilitar acessos a determinada aplicação, foi criado um tributo *Common name* - CN (Nome comum) próprio para a aplicação, não utilizando atributos ou *Organizational unit* - OU (Unidade organizacional) previamente configuradas no Zimbra.

Segundo DONLEY (2003, pg 4) [2], LDAP é um padrão que computadores e dispositivos de rede podem usar para acessar informações sobre a internet. O protocolo LDAPv3 que é definido pela RFC 2251, se apresenta em muitas aplicações que são utilizadas por profissionais da área de tecnologia, embora integrado com a maioria delas, seu uso para o gerenciamento não é muito difundido. A ideia por trás do LDAP, segundo DONLEY (2003, p.27), é que não importa onde os dados finais estão armazenados, desde que tanto o cliente e o servidor possam usar LDAP para trocar informações de uma maneira que possam se entender pelos dois lados." A arquitetura de uma requisição ao servidor LDAP segue exemplificado pela Figura 3, onde são apresentadas basicamente 4 operações, ldap_open, ldap_bind, ldap_search e ldap_unbind.

TUTTLE (2009) [5], afirma que o LDAP é um padrão aberto capaz de facilitar, de forma flexível, o compartilhamento, a manutenção e o gerenciamento de grandes volumes de informações, definindo um método-padrão de acesso e atualização de informações dentro de um diretório.

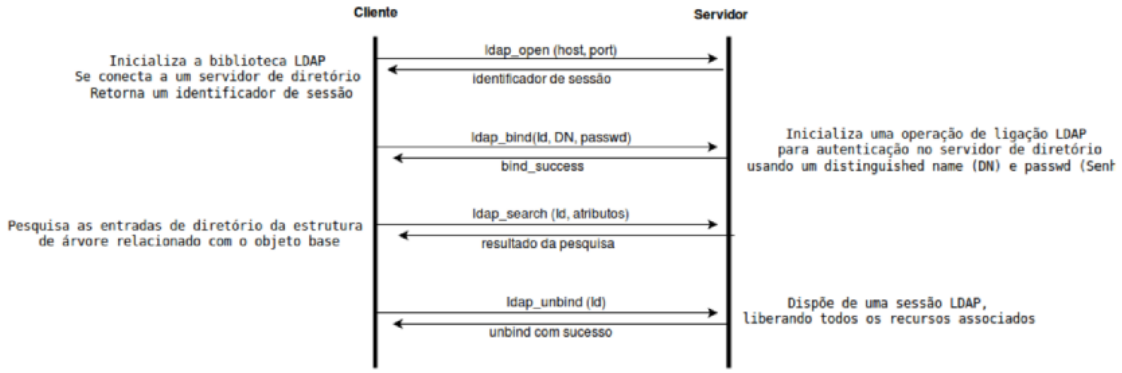


Figura 3: Funcionamento de uma pesquisa ao servidor LDAP - adaptada de WAN (et al., 2008). [3]

A operação *ldap_open*, abre uma conexão com o serviço de diretório e retorna uma sessão para uso futuro. O *ldap_bind* é responsável pela autenticação do cliente. A operação *Bind* permite o cliente se autenticar com o servidor de diretório através de um DN (*distinguished name*, ou nome distinto), e credenciais, como usuário e senha. Quando a operação bind é completada com sucesso, o servidor de diretório salva esta informação antes que um novo *bind* seja realizado ou quando a sessão é terminada pela chamada de um *ldap_unbind*. A identidade é usada pelo servidor para fazer decisões sobre qual tipo de mudanças podem ser feitas no diretório. O ldap_search é a operação que se inicializa a pesquisa LDAP através de um critério especificado que se combina com um filtro associado. Finalmente,

a sessão LDAP é fechada utilizando o *ldap_unbind*.

O processo para acessar as aplicações integradas, é realizado através das ações de autentiação e autorização. A autenticação é o processo no qual verifica a identidade digital do usuário para acessar determinado sistema no momento do *login*. Já a autorização, após o usuário ser autenticado, é a capacidade de determinar se o usuário, com o determinado atributo ou classe, tem a permissão de executar uma ação ou acessar um recurso.

É importante definir alguns níveis de segurança e configurá-los corretamente antes de permitir a realização de uma pesquisa. Como ilustração deste ponto, imagine uma base de dados de usuários e senhas utilizada para efetuar o *login* nos principais serviços da rede que serão integrados. Qual seria o tamanho do desastre caso um usuário qualquer, ao fazer uma pesquisa para saber quem são os funcionários da empresa, recebesse junto com os nomes as senhas de todos? Portanto, é preciso determinar regras de segurança para o melhor funcionamento da base, TRIGO (2007).

2.3 Serviço de Diretório Vs Banco de Dados relacional

Para entender melhor sobre o que é um diretório, segue uma definição da Universidade de Michigan [26], que desenvolveu o protocolo LDAP. A Universidade definiu que um diretório é uma base de dados, mas tende a conter mais informações descritivas e atributos. A informação dentro de um diretório é geralmente mais executada para leitura do que escrita e por consequência disto, usualmente não implementa transações complexas para fazer uploads de um elevado volume de dados. Portando, um serviço de diretório é um banco de dados que tem a finalidade de otimizar o acesso e a administração desta base.

Um banco de dados relacional, segundo PRICE (2009, p.30)[24]:

> "é uma coleção de informações relacionadas, organizadas em tabelas. Cada tabela armazena dados em linhas; os dados são organizados em colunas. As tabelas são armazenadas em esquemas de banco de dados, que são áreas onde os usuários podem armazenas suas próprias tabelas".

Considerando que as atualizações são mais constantes em banco de dados do que em um sistema de diretório, o protocolo LDAP se torna uma solução mais eficaz (Gil, 2016)[22].

O grande responsável pela flexibilidade do LDAP é a organização das informações de forma hierárquica. A árvore de informações possui um elemento-raiz, por onde começa a busca das informações. A partir daí, o sistema vai percorrendo os nós-filhos até encontrar o elemento desejado, TRIGO (2007).

A Figura 4 exemplifica o entendimento de um sistema de diretório.

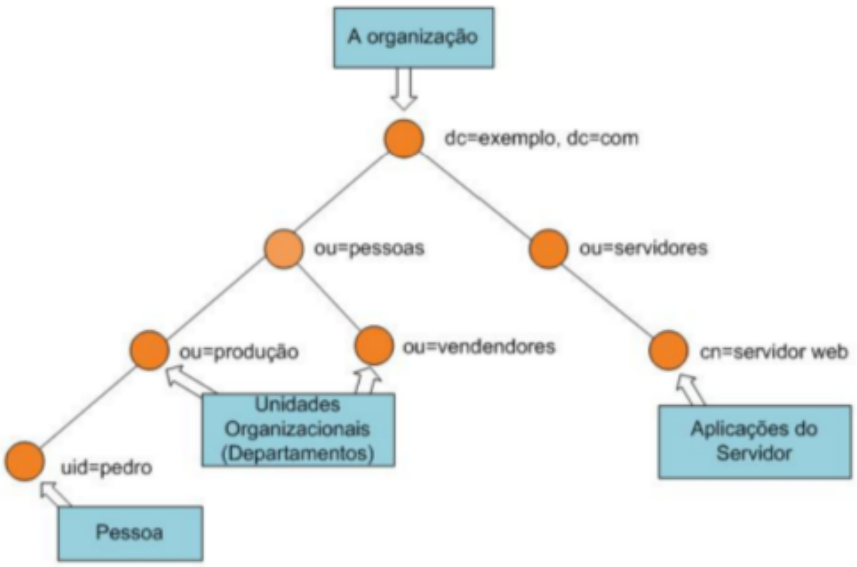


Figura 4: Sistema de diretórios. Fonte: EVARISTO(2008)

Por sua vez, BUTCHER (2007, p.16) [4] relata que a maioria dos sistemas de arquivos modernos representa dados de uma forma hierárquica. Por exemplo, no sistemas Unix, o diretório /home pode ter múltiplos subdiretórios: /home/mbutcher, /home/ikant, /home/dhume. Podemos dizer que /home possui três subordinados, mas cada um deles possui um superior (o diretório /home). Quando pensamos sobre diretórios em árvores LDAP, pode ajudar a compará-lo com o layout de um sistema de arquivos.

Seguindo mais uma definição, notamos o porquê da vantagem de se implantar um serviço de diretório ao invés de um banco de dados relacional, onde segundo TUTTLE (2004, p.7) [5], diretórios não se destinam a fornecer tantas funções de uso geral como as bases de dados relacionais, eles podem ser otimizados para economicamente fornecer mais aplicações com acesso rápido aos dados do diretório em grande ambientes distribuídos. Portanto, um serviço de diretórios fornece as direções para se chegar as informações requisitadas, retornando ao usuário o que foi requisitado, otimizado para leitura e com um sistema de transações simples quando comparado a um banco de dados relacional.

2.4 Vantagens e desvantagens do LDAP

Segundo CHAVES (2010), as principais vantagens do LDAP são:

- É um padrão aberto;
- É otimizado para realizar pesquisas e leitura;
- Centraliza toda a informação proporcionando enormes benefícios tais como: um único ponto de administração, menos informação duplicada, maior transparência das informações;
- Possui um mecanismo de replicação da base incluído;
- Suporta mecanismos de segurança para autenticação (SASL) e para troca de dados (TLS);
- Muitas aplicações e serviços possuem suporte ao LDAP.

As principais desvantagens do LDAP são:

- Em alguns casos não substitui as bases de dados relacionais;
- Pouco eficiente para operações de escrita e atualização;
- Integração com outros serviços e aplicações torna a implantação complexa.

2.5 OpenLDAP e Active Directory

Com a popularização dos sistemas de diretórios e sua grande utilidade, muitas aplicações do mesmo gênero surgiram posteriormente. Como exemplos, podemos citar o OpenLDAP e o Microsoft Active Directory. OpenLdap se destina à utilização do protocolo LDAP, para manipular o serviço de diretórios, e sua implementação é utilizada em plataformas Linux, por isso foi escolhido para ser implementado neste projeto. Segundo os desenvolvedores da OpenLDAP Foundation [17], o *software* OpenLDAP é uma ferramenta de código aberto que implementa o protocolo LDAP. Sendo este projeto gerido através de colaboradores e desenvolvedores, que conforme o site, utilizam-se a internet para se comunicar, planejar e desenvolver o projeto OpenLDAP e a documentação relacionada. O Active Directory, possui as mesmas funcionalidades do OpenLDAP, porém realiza a implementação do serviço de diretório nos sistemas operacionais da Microsoft.

2.6 Problema da pesquisa

Manter diversas soluções de autenticação pode ser oneroso, pois deve-se manter diversas bases de dados e ainda replicar estes dados em caso de mudança de informações do usuário. As redes heterogêneas formam um cenário típico para implantação de um servidor LDAP, onde se consegue integrar os dados a partir de redes formadas por sistemas operacionais diferentes, rodando aplicações distintas.

A Figura 5 apresenta uma estrutura de rede heterogênea (diversos hardwares e softwares), muito comum nos ambientes corporativos, onde protocolo LDAP tem sua aplicação justificada. Como o protocolo serve tal como um repositório de informações, aplicações com suporte ao LDAP podem se comunicar com a base de dados para coleta e acesso as informações. Segundo BUTCHER (2007, p.6) [4], deve-se usar um servidor de diretórios quando você precisar gerenciar os dados de forma centralizada, armazenados e acessíveis através de métodos padrões.

Figura 5: Exemplo de redes Heterogêneas - Fonte: OpenLDAP Foundations [17].

2.7 Aplicações que serão integradas ao servidor LDAP

O LDAP, por se tratar de um repositório centralizado de informações, permite que qualquer aplicação que possua suporte ao protocolo possa se conectar a base e obter as informações necessárias. Algumas aplicações foram escolhidas para fazer esta integração com o protocolo. Baseando-se em uma infraestrutura de rede de um ambiente corporativo

que provê serviços em computação em nuvem, do inglês *Cloud Computing*, e-mail e hospedagem de sites, ferramentas que administram e gerenciam as contas dos usuários, como por exemplo WHMCS, foram escolhidas para integração, assim como o serviço SSH. A plataforma de e-mail corporativo Zimbra e o CMS WordPress também serão integrados.

Todas as aplicações citadas serão abordadas nas próximas seções, e suas respectivas configurações no CAPÍTULO 4.

2.7.1 SSH - Secure Shell

Segundo BARRETT(2005, p.1) [18], o SSH é uma popular e poderosa ferramenta baseada em *software* para a segurança da rede. Qualquer arquivo é enviado de um computador através da rede, o SSH automaticamente encripta estes dados. Quando os dados chegam no receptor, o SSH automaticamente decripta esta mensagem. Possui uma arquitetura cliente/servidor e é tipicamente instalado pelo administrador do sistema.

O crescimento de uma empresa, em paralelo com um aumento do número de colaboradores, ocasiona um maior número de usuários para realizar *login* de autenticação em servidores e aplicações, necessitando uma melhor administração da estrutura. O protocolo SSH (*Secure Shell*) permite o acesso remoto aos computadores via console, assim como a transferência de arquivos de uma maneira segura. Este processo funciona através do módulo PAM, no qual será integrado ao sistema de diretórios LDAP para gerenciar os usuários que terão acesso SSH aos servidores.

2.7.2 WHMCS

Esta é uma plataforma para empresas que realizam a hospedagem de sites.
Segundo WHMCS(2016) [19], a plataforma é líder mundial em automatização de cobrança, onde provê uma solução completa para os serviços relacionados com hospedagens na web. Esta ferramenta, faz com que todas as cobranças, o gerenciamento dos sites e usuários administrativos, sejam integrados com outras plataformas úteis quando se tratam da hospedagem de sites.

A utilização do protocolo LDAP atuando como backend na integração do sistema de pagamento (WHMCS), irá facilitar a administração das contas e credenciais de toda uma base de usuários. Esta integração com o LDAP, faz com que a entrada ou saída dos colaboradores da empresa ServerDo.in (Empresa de Hospedagem que será realizada a

integração com o LDAP) possa ser gerenciada mais facilmente, garantindo uma política de segurança melhor, evitando falhas ou cruzamento de dados.

2.7.3 Zimbra

Zimbra é um software de código aberto para mensagens e colaboração, email, calendário, contatos e gerenciamento de documentos web. O servidor Zimbra está disponível para Linux e Mac OS, e plataformas de virtualização (RESNICK, 2007) [21].

Sendo assim, é uma solução de e-mail corporativo muito utilizada, é uma solução de código aberto líder neste mercado (ZIMBRA, 2016) [20]. A plataforma Zimbra portanto, por ser líder nesta área, possui um grande potencial para continuar ganhado espaço e se consolidando como uma ferramenta robusta e essencial para empresas de tecnologia.

A integração do servidor LDAP com os serviços apresentados, proporciona para uma empresa de hospedagem uma administração simples e uma adequada estrutura da rede. Vale lembrar que estes serviços já possuem um módulo para configuração com o serviço de diretório LDAP, tornando possível esta integração.

2.7.4 WordPress

O WordPress "é uma plataforma semântica de vanguarda para publicação pessoal, com foco na estética, nos padrões *web* e na usabilidade. Ao mesmo tempo é um software livre, gratuito", WORDPRESS (2016) [25].

Cerca de 90% (dados coletados na empresa) dos sites hospedados pela ServerDo.in utiliza a plataforma em WordPress, o que totaliza uma média de 200 sites/clientes. O WordPress atrai as demandas nesta área de desenvolvimento de sites, pois tem como vantagens sua vasta gama de plugins e temas para serem vinculados à plataforma, sendo uma ferramenta poderosa também no quesito manutenção.

A facilidade para desenvolver e usar essa plataforma é uma de suas maiores vantagens, assim como a comunitadade ativa que pode ajudá-lo quando encontrar problemas, e o fato de que trata-se do que há de melhor em código aberto, HEDENGREN (2012) [27].

3 *Implementação proposta*

3.1 Estudo de caso

Envolvendo a área de hospedagens de websites e computação em nuvem, foi identificado uma fraqueza quanto a administração, segurança, e o gerenciamento dos dados e usuários. Em questão de segurança na área de serviços de diretórios, DONLEY (2003, p.254) [2], explica que ela tende a implicar que as informações e funcionalidades estão disponíveis para aqueles que deve ser capaz de acessá-los e indisponível para aqueles que não deveriam ter acesso.

Uma boa solução de gerenciamento é essencial, onde além de facilitar a sincronização das informações, cria-se uma política de segurança forte, organiza o acesso aos diretórios (criando uma proteção com senha, por exemplo), diminui a dificuldade de organização e até mesmo o custo no suporte para resolução de cenários deste tipo.

Ações rápidas são necessárias para resolver as invasões provenientes de usuários ou arquivos indesejados. A troca de e-mails para receber uma credencial de acesso ao *admin* do site, muitas vezes se torna lenta entre o administrador da hospedagem e o cliente. A integração com os sites desenvolvidos em WordPress facilitará este processo.

Segundo TRIGO (2007),

> Pode existir uma série de aplicativos na rede trabalhando com a mesma base, tornando, assim, muito mais fácil a vida do administrador de rede. Quando um usuário alterar a senha, automaticamente todas as aplicações utilizarão a nova senha alterada e assim não será necessário alterá-la em uma série de aplicações. O administrador da rede pode colocar uma política de senha forte, e esta política valerá para todos os aplicativos. Quando um usuário deixa o quadro de funcionários, basta desabilitá-lo da base LDAP e automaticamente esse usuário não terá acesso à rede.

As decisões do administrador da rede em relação a política de segurança das senhas da

organização, determina o quão segura ficará a infraestrutura. A política de senhas define um controle de segurança a partir de um conjunto de regras projetadas, incentivando os usuários a empregar senhas fortes e usá-las adequadamente. Com as aplicações integradas, pode-se implantar um padrão de senha melhor definido aos usuários, incentivando uma conscientização de segurança.

Portanto, em uma infraestrutura de rede, contando com um grande número de servidores, serviços e aplicações, se faz necessário realizar uma comunicação centralizada. Após feita uma análise sobre este setor da tecnologia, decidiu-se integrar as ferramentas e aplicações mais utilizadas que foram descritas anteriormente. A Figura 6 mostra, de forma gráfica, como ficará a rede com o servidor LDAP. O usuário para acessar as aplicações, fará uma requisição transparente ao servidor LDAP, e assim ter acesso as aplicações.

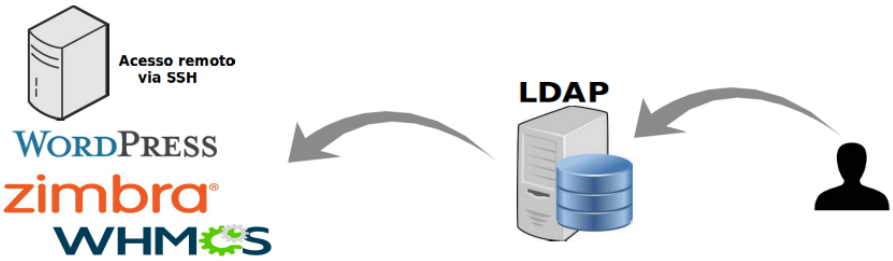


Figura 6: Utilização da base LDAP.

Escolhemos as aplicações e vamos implantá-las na empresa ServeDo.in, utilizando a infraestrutura da empresa como estudo de caso para os problemas enfrentados pela mesma. Vale lembrar que os serviços a serem integrados são utilizados por seus colaboradores.

4 Desenvolvimento

Este capítulo será dividido em duas seções, são eles: Criação do ambiente e Configurações do projeto.

4.1 Criação do ambiente

Para se criar o ambiente de implementação utilizado neste projeto, é necessário instalar a ferramenta de e-mail colaborativa Zimbra, a ferramenta administrativa WHMCS, assim como o phpLDAPadmin e o WordPress. Para configuração do SSH é utilizado servidores virtualizados. Estas aplicações são executadas em diferentes servidores, como demonstra a Figura 7. As instalações padrões de cada ferramenta serão apresentadas, e posteriormente como estas ferramentas foram configuradas para se criar uma autenticação integrada junto ao serviço de diretório LDAP. Todos os servidores utilizados no projeto utilizam a versão Linux Ubuntu 14.04.

Os serviços estão operando, conforme Figura 7, em servidores localizados na Amazon Web Server - AWS, que oferece serviços de computação em nuvem e possui uma grande rede de recursos nesta área. Os servidores clientes utilizados como testes para acesso SSH via *backend* LDAP, são máquinas virtuais, VPS (*Virtual Private Server*), localizados em diferentes *datacenters*.

4.1.1 Zimbra

A versão do Zimbra utilizada é Zimbra 8.0.6_GA_5922. Para se instalar a ferramenta Zimbra, é necessário executar os seguintes comandos:

```
# wget https://files.zimbra.com/downloads/8.6.0_GA/zcs-8.6.0_GA_1153.UBUNTU14_64.20141215151116.tgz
# tar -xvf zcs-8.6.0_GA_1153.UBUNTU14_64.20141215151116.tgz
```

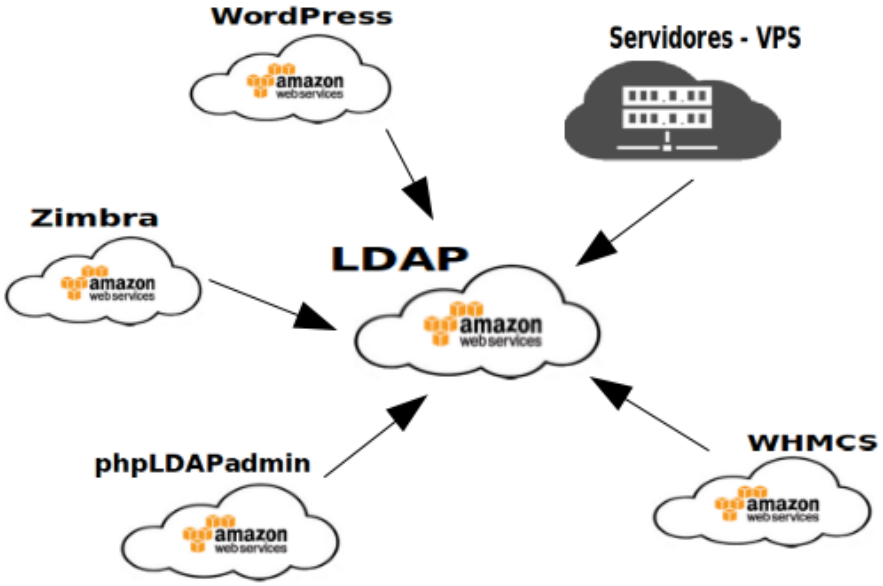


Figura 7: Distribuição dos serviços.

```
# cd zcs-8.6.0_GA_1153.UBUNTU14_64.20141215151116

# sudo ./install.sh
```

Neste momento é necessário seguir com as configurações do assistente de instalação. Como este serviço de e-mail já é utilizado na empresa, a instalação desta ferramenta em outro servidor não foi necessária, e como o Zimbra já possui o serviço de diretórios através do OpenLDAP, também não é necessário fazer instalação do mesmo, pois utilizaremos esta base LDAP para as integrações.

4.1.2 WHMCS

Os seguintes passos são necessários para instalar a ferramenta WHMCS: Primeiro é necessário fazer o *download* da versão desejada em `http://whmcs.com` e fazer o *upload* no servidor. Posteriormente renomear o arquivo *configuration.php.new* para *configuration.php*. Visitar o script de instalação em `http://serverdo.in/whmcs/install/install.php` e seguir com os processos de instalação. Lembrando que este procedimento envolve uma licença para uso.

Como trata-se de uma ferramenta administrativa, não foi necessário realizar este procedimento de instalação, e sim adquirir o conhecimento para a instalação e configuração do plugin compatível com o WHMCS.

4.1.3 WordPress

Para se instalar o Sistema de Gerenciamento de Conteúdo WordPress, foi realizado o *download* da versão desejada em `http://br.wordpress.org`. Após *download*, é necessário descompactar os arquivos, criar um banco de dados mysql e vincular ao procedimento de configuração. Como a instalação foi executada em um ambiente em nuvem, deve-se seguir com os passos de instalação em `http://serverdo.in/wp-admin/install.php` .

Tendo o site `http://serverdo.in` desenvolvido sobre a plataforma WordPress, novamente não foi necessário realizar o procedimento de instalação, pois será usado como testes o próprio site da empresa, para uma posterior implementação da base LDAP na estrutura dos clientes.

4.1.4 phpLDAPadmin

Para auxiliar nas configurações do serviço de diretórios e ter uma visão via *browser* de toda base LDAP, como citado no 4.1.1, foi configurado no servidor uma aplicação em PHP chamada phpLDAPadmin, um *frontend* do LDAP, onde traz toda a raiz de diretório, assim como todos os usuários, grupos e atributos adicionados para cada usuário independente.

Para se obter esta interface gráfica e gerenciar a base de dados, é executado o seguinte comando:

```
# sudo apt-get install phpldapadmin
```

Com este comando, é instalado o phpLDAPadmin e também o PHP e o servidor *web* Apache. Maiores detalhes sobre a configuração é descrita na subseção 4.2.2

4.2 Configurações do projeto

A seguir será explanado a configuração de cada um dos serviços previamente escolhidos para integração. Note que, os erros durante os processos também serão documentados, mostrando as dificuldades do projeto.

4.2.1 Zimbra

Já com o ambiente criado, um dos primeiros detalhes do trabalho, foi identificar que a plataforma Zimbra já possui uma base LDAP integrada (*hardcoded*), ou seja, o Zimbra já utiliza as credencias dos usuários (autenticação e autorização), utilizando o protocolo LDAP (OpenLDAP). Como o projeto é aplicado dentro da empresa ServerDo.in, se tornou interessante utilizar a mesma plataforma que já vem sendo utilizada, ou seja, utilizar o mesmo servidor e a base de usuários LDAP já existente no serviço Zimbra, para realizar a integração com as aplicações desejadas.

A utilização do *backend* LDAP da plataforma Zimbra, realizando integrações com outras aplicações não é muito difundida na internet. Realizar novas configurações em um ambiente de produção, instalando novos processos e/ou alterando arquivos de configuração e permissões, podem ocasionar em paralisações nos serviços, o que resulta em perdas se tratatando de um ambiente corporativo. Portanto, um estudo mais aprofundado sobre o Zimbra se fez necessário, para identificar a capacidade da ferramenta como um serviço de diretórios e conhecer a estrutura deste serviço.

4.2.1.1 Criando um usuário no servidor de e-mail Zimbra

Ao se criar um usuário, procurou-se verificar quais eram os atributos e *objectClass* (classe do objeto) adicionados na conta, procurando-se entender e diferenciar as permissões que são criadas junto ao protocolo LDAP. Dentre os vários esquemas que são utilizados dentro desta base, são adicionados os seguintes padrões de *objetClass* para conta do usuário:

inetOrgPerson, zimbraAccount e amavisAccount

É importante detalhar que um esquema é formado por um conjunto de objectClass, e este é formado por uma coleção de atributos. Portanto, vemos um LDAP hierárquico, onde o objectClass é definido dentro dos esquemas, e os atributos definidos em um ou mais objectClass.

A Figura 8 traz as informações de atributos do *objectClass* **inetOrgPerson**, que por sua vez herdou informações de *organizationalPerson*. Aqui identificamos os atributos que são requeridos, ou seja, que precisam ter os campos preenchidos, e os atributos opcionais. Segundo a RFC2798 - *Internet Organizational Person*, a *inetOrgPerson objectClass* é uma classe de propósitos gerais que possui atributos sobre pessoas.

Required Attributes

- cn
  (Inherited from person)
- sn
  (Inherited from person)

(a)

Optional Attributes

- audio
- businessCategory
- carLicense
- departmentNumber
- displayName
- employeeNumber
- employeeType
- givenName
- homePhone
- homePostalAddress
- initials
- jpegPhoto
- labeledURI
- mail
- manager
- mobile
- o
- pager
- photo
- preferredLanguage
- roomNumber
- secretary
- uid
- userCertificate
- userPKCS12
- userSMIMECertificate
- x500uniqueIdentifier
- destinationIndicator
  (Inherited from organizationalPerson)
- facsimileTelephoneNumber
  (Inherited from organizationalPerson)
- internationaliSDNNumber
  (Inherited from organizationalPerson)

(b)

Figura 8: Atributos obrigatórios e opcionais.

A classe de objeto **zimbraAccount**, segundo o site do desenvolvedor, "utiliza classes de objeto auxiliares para adicionar atributos específicos para uma conta". Portanto, é com esta classe que ele diferencia quem pode ou não enviar/receber e-mails, é definido se o e-mail está em alguma lista de distribuição, informações do servidor, entre outros.

Sobre o **amavisAccount**, ele é também mais uma classe auxiliar que vai representar a conta de e-mail. Nesta classe de objeto ficam os atributos referentes ao anti-virus, listas de quarentena, listas brancas de e-mail, tags de spam, entre outros.

O próximo passo foi descobrir qual base e senha de admin do LDAP Zimbra. Executando os comandos abaixo, conseguimos resgatar estes dados e assim configurar as aplicações e os plugins utilizados. Primeiramente devemos realizar o *login* com o usuário 'zimbra' e executar os comandos abaixo para descobrir qual é o usuário gerenciador do LDAP do Zimbra e sua credencial.

```
# zmlocalconfig zimbra_ldap_userdn
```

Temos como resposta:

$uid = zimbra, cn = admins, cn = zimbra$

```
# zmlocalconfig -s zimbra_ldap_password
```

Temos como resposta:

$zimbra\_ldap\_password = * * * * * * *$

Ao ser adicionado um usuário na plataforma Zimbra, junto às classes de objeto criadas, são criadas as credenciais de acesso, os atributos, permissões e outras informações desejadas de cadastro.

A tela para se adicionar um novo usuário é demonstrada na Figura 9.

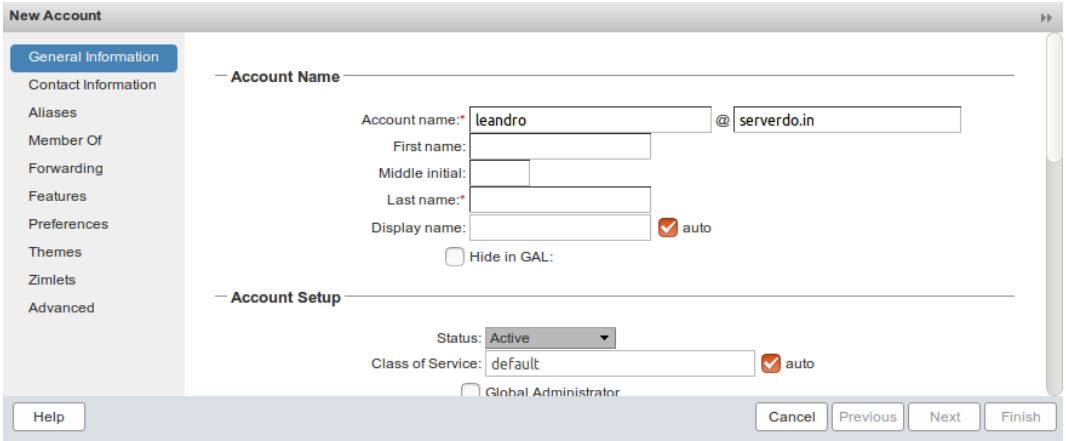


Figura 9: Adicionando novo usuário, Fonte: Própria.

A hierarquia do usuário quanto ao domínio, chamado no termo técnico como DC(*Domain Control*), segue:

$ou = people, dc = serverdo, dc = in$

Ou seja, para o usuário de testes criado (leandro), as informações do DN ficariam:

$uid = leandro, ou = people, dc = serverdo, dc = in$

UID (Identificador de usuário), **OU** (*Organizational Unit*), organização única simples , que é chamado de *people* neste exemplo que é padrão do LDAP; **DC**, ou controle de

domínio, que é o domínio da empresa e consequentemente o endereço de e-mail.

As prioridades são importantes para identificar as necessesidades a serem configuradas, a sequência de prioridade padrão segue:

País, organização, unidade organizacional e pessoa.

O servidor ldap é nomeado *slapd*, e seus arquivos de configuração geralmente se encontram em */etc/openldap/slapd.conf*, mas como estamos usando o LDAP do Zimbra, os arquivos se apresentam em diferentes diretórios: */opt/zimbra/openldap* .

Os clientes LDAP são configurados utilizando o arquivo que se encontra em:

$/opt/zimbra/openldap/etc/openldap/ldap.conf.$

Aqui podemos verificar se a base e o servidor estão configurados corretamente:

$BASE\ cn = admins, cn = zimbra$
$URI\ ldap : //zimbra.serverdo.in$

Outro arquivo importante é o *slapd.conf*, que traz informações sobre os esquemas que são utilizados e também informações de importantes para o funcionamento do serviço:

$include\ /opt/zimbra/openldap{-}2.4.38.2z/etc/openldap/schema/core.schema$
$include\ /opt/zimbra/openldap - 2.4.38.2z/etc/openldap/schema/nis.schema$
$database\ bdb$
$suffix\ \text{“}cn = admins, cn = zimbra\text{”}$
$rootdn\ \text{“}uid = zimbra, cn = admins, cn = zimbra\text{”}$
$rootpw\ secret$

Com o serviço rodando e as configurações corretas, vamos avançar nas configurações. Para checar o status do serviço, com o usuário 'zimbra', é executado:

```
# /opt/zimbra/bin/ldap  status
```

Tendo como saída:

$slapd\ running\ pid :\ 5435$

Ou seja, o serviço está rodando, com a identidade do processo (*process ID*) número 5435.

4.2.2 phpLDAPadmin

Os arquivos de configuração desta aplicação ficam localizados em */etc/phpldapadmin/*. Nesta pasta, o arquivo *config.php* segue configurado:

$$servers->setValue('server','name','Serverdo.inLDAPServer');$$
$$servers->setValue('server','host','ldap://br14.serverdo.in:389');$$
$$servers->setValue('server','base',array('dc=serverdo,dc=in'));$$
$$servers->setValue('login','bind_id','uid=zimbra,cn=admins,cn=zimbra');$$

A ferramenta phpLDAPadmin foi configurada no mesmo servidor do LDAP, nesta etapa ocorreram alguns problemas. A instalação do phpLDAPadmin inclui por padrão uma instalação servidor http Apache, e como a plataforma Zimbra já utiliza o servidor *web* Apache, alguns arquivos de configuração acabaram sendo sobreescritos, impossibilitando o acesso ao e-mail através da url zimbra.serverdo.in (url configurada para abrir o serviço de e-mail), pois a instalação do phpLDAPadmin forçou o uso da porta 80, porta na qual a aplicação Zimbra já estava em execução. Foi necessário configurar a aplicação phpLDAPadmin na porta 81, não prejudicando a aplicação no servidor de produção.

Portanto, os seguintes arquivos precisaram ser alterados para que as aplicações Zimbra e o phpLDAPadmin pudessem ser executadas em paralelo.

/etc/apache2/sites-available/ldap

/etc/apache2/sites-available/default

As portas no *firewall*, tanto do serviço phpLDAPadmin, bem como do LDAP (porta padrão 389), tiveram que ser liberadas. Para garantir a segurança, a empresa ServerDo.in configura um *firewall* utilizando o *Iptables*, liberando apenas as portas necessárias para as aplicações existentes no servidor. Por esse motivo as portas 81 e 389, porta do phpLDAPadmin e do LDAP respectivamente, precisaram ser liberadas. O arquivo *rules.v4* foi alterado, deixando esta configuração como padrão nas regras do *Iptables*, não prejudicando os acessos à interface *web* e a utilização do protocolo:

/etc/iptables/rules.v4

$$-AINPUT - i\ eth0 - p\ udp - -dport\ 389 - j\ ACCEPT$$
$$-AINPUT - i\ eth0 - p\ tcp - -dport\ 389 - j\ ACCEPT$$
$$-AINPUT - i\ eth0 - p\ tcp - -dport\ 81 - j\ ACCEPT$$
$$-AINPUT - i\ eth0 - p\ udp - -dport\ 81 - j\ ACCEPT$$

Após as modificações, é preciso reiniciar o serviço:

```
# sudo service iptables-persistent restart
```

Para confirmar as configurações:

```
# sudo iptables -L
```

Tendo como saída:

$$ACCEPT\ udp\ - -anywhere\ anywhere\ udp\ dpt : ldap$$
$$ACCEPT\ tcp\ - -anywhere\ anywhere\ tcp\ dpt : ldap$$
$$ACCEPT\ tcp\ - -anywhere\ anywhere\ tcp\ dpt : 81$$
$$ACCEPT\ udp\ - -anywhere\ anywhere\ udp\ dpt : 81$$

O acesso a aplicação se dá através da url `zimbra.serverdo.in:81/phpldapadmin`, após o *login* como usuário gerenciador da base, a Figura 10 é apresentada.

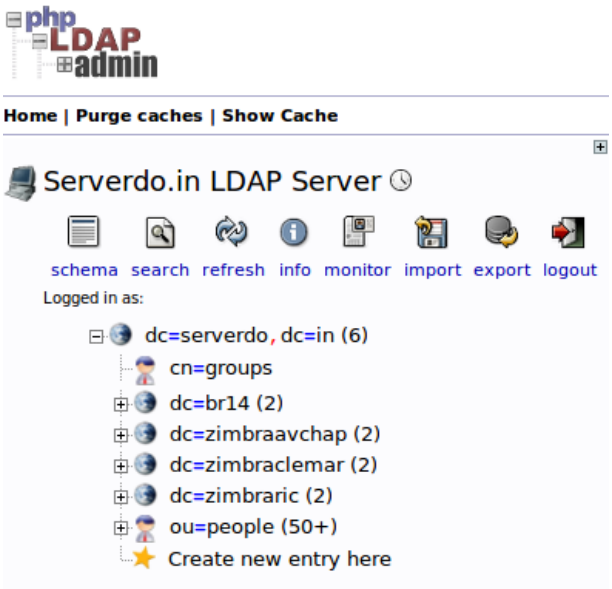


Figura 10: Tela principal após o *login* no phpLDAPadmin.

Observamos na Figura 10, que através da ferramenta phpLDAPadmin podemos realizar consultas, verificar esquemas, importar/exportar configurações, monitorar o serviços e ter um controle de todos os usuários que estão populados na base de dados (diretórios). As principais vantagens observadas através da ferramenta foram:

- Ter uma visão através do browser de todos os usuários e da árvore de diretórios;
- Adicionar/excluir atributos e unidades organizacionais aos usuários;
- Criar uma nova base de diretório;
- Alterar as senhas;
- Vincular conta como administrador;
- Preencher e/ou alterar campos de cadastro.

Grande parte destas ações citadas podem ser também realizadas através do CLI (*Command Line Interface*), via linha de comandos.

O arquivo do phpLDAPadmin *apache.conf* localizado em /etc/phpldapadmin/, possui uma variável de entrada chamada *php_value memory_limit*, configurada por padrão com um valor baixo, fazendo com que a aplicação não listasse os usuários da base LDAP. Foi aumentado esta variável para *php_value memory_limit 1024M*, solucionando os problemsa da falta de memória para listar as entradas.

A ferramenta se mostra de muita importância para o administrador da rede, que possui total controle a partir de uma interface *web*, gerenciando todo o sistema.

4.2.3 WordPress

Como muitos dos sites hospedados pela empresa ServerDo.in são desenvolvidos no CMS WordPress, a integração com o admin do site para realização do *login* via protocolo LDAP tornou-se fundamental. Sites com o mesmo propósito do WordpPress, possuem uma interface para os usuários, sejam eles editores ou administradores, realizarem os *uploads* e postagens. O usuário quando criado, é adicionado a uma base de dados relacional, portanto, durante um procedimento de *login*, a aplicação busca nesta base o usuário e faz a checagem com a senha cadastrada.

Uma empresa de hospedagem, possui uma larga escala de sites em sua infraestrutura, onde o *webmaster* (administrador da hospedagem) presta suporte às demandas dos clientes, mantém a estabilidade do site, mantendo o serviço *online* e livre de *malwares* (Software malicioso) ou outras ameaças encontradas no mundo virtual. Torna-se complexo o gerenciamento de todos os acessos, pois é necessário criar um usuário administrador em cada site hospedado e recordar-se das credenciais.

É recorrente a troca de e-mails entre empresa e cliente, visando a busca das credenciais de administrador, para assim analisar algum problema diagnosticado por ambos. Visando resolver esta demanda, otimizando este processo, a autenticação *backend* pelo protocolo será configurada.

Com o auxílio de um *plugin* de código aberto chamado Simple LDAP Login, desenvolvido por Clif Griffin, foi possivel instalar e configurar o protocolo LDAP dentro do ambiente de teste (http://serverdo.in), e através do Simple LDAP Login, foi minimizado os transtornos de acesso ao *admin* do site, permitindo que tanto os usuários cadastrados previamente no banco de dados relacional realizem os acessos, assim como os usuários da base LDAP. Ver Figura 11.

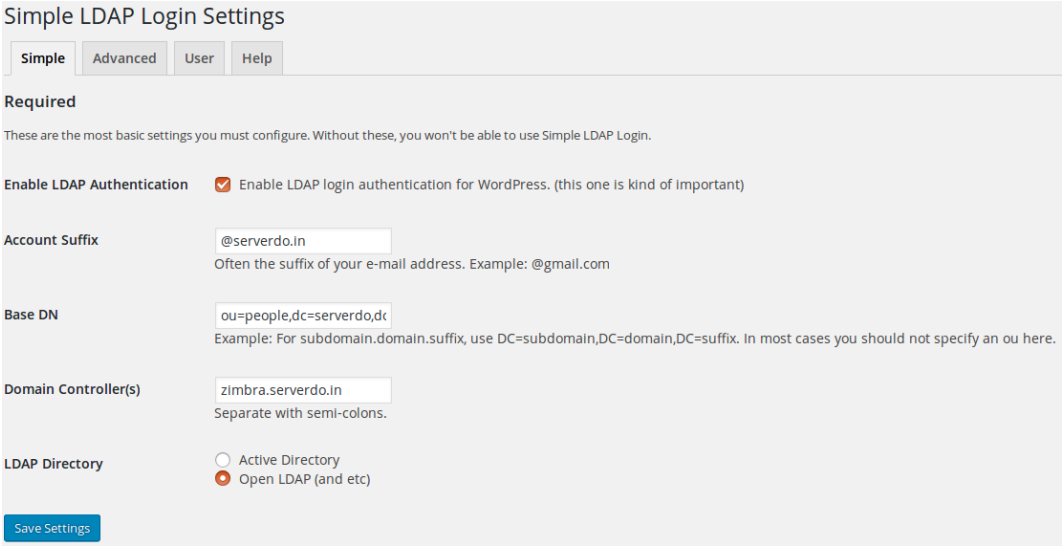


Figura 11: Plugin Simple LDAP Login.

O plugin conseguiu identificar a existência de uma base de usuários LDAP, porém se faz necessário distinguir quem poderá acessar ou não esta interface administrativa dos sites em WordPress, uma vez que a base LDAP possui todos os usuários da empresa. Entretando, o plugin Simple LDAP Login, em sua versão 1.6.0, não traz uma opção para filtro de usuários.

Foi necessário alterar o código fonte do plugin para realizar um filtro de pesquisa, as linhas alteradas no arquivo *Simple-LDAP-Login.php* seguiram da 394 até 410, conforme Figura 12, fazendo com que o *bind* (requisição) da pesquisa analizasse previamente um CN (*common name*) chamado wordpress (CN=wordpress) adicionada aos usuários que terão os acessos administrativos. Da mesma forma, foi fundamental alterar o *schema* para incluir este CN aos usuários que terão permissão de acesso, pois por padrão ele não existe no LDAP.

```
// Se o bind está correto, login e senha estão Ok.
if ($result == true)
{
// Fazemos a busca pelo cn=wordpress
$inGroup = ldap_search($this->ldap, $this->get_setting('base_dn'), 'cn=wordpress', array('cn'));
$table = ldap_get_entries($this->ldap, $inGroup);

// Verificando se o login que se autenticou esta no cn do wordpress
$newResult = false;
for ( $i = 0; $i < $table['count']; $i++ )
{
if (strpos($table[$i]['dn'], $filter) !== false )
{
$newResult = true;
}
}
}
```

Figura 12: Plugin Simple LDAP Login, com alterações no código em PHP.

A arquitetura para realizar o *login* é dada como segue na Figura 13

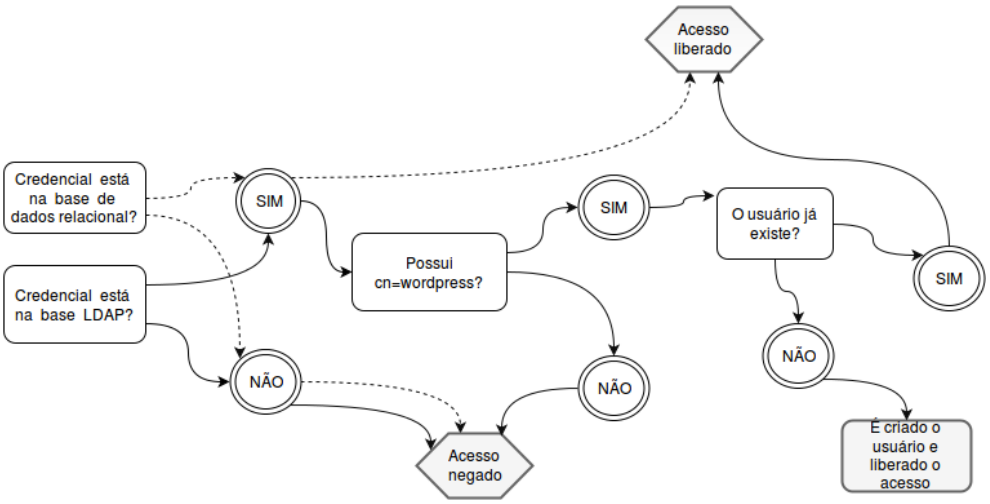


Figura 13: Arquitetura de *login* após modificações no plugin.

Temos como exemplo o usuário *leandro*, ele foi criado para ser o usuário de testes durante o projeto e avaliar as integrações. Ao ser criada uma nova conta de e-mail para *leandro*, sua árvore de diretórios padrão ficará como: *uid=leandro,ou=people,dc=serverdo,dc=in*. Adicionamos o parâmetro *cn=wordpress* ao *uid=leandro*, logo a busca ldap(ldapsearch) através do plugin irá liberar o acesso.

Segue o domínio do usuário após ser adicionado o novo parâmetro:

$$cn = wordpress, uid = leandro, ou = people, dc = serverdo, dc = in$$

Estudar o código fonte e adaptá-lo para necessidades foi uma tarefa difícil. Sendo escrito em PHP, precisou-se compreender a linguagem utilizada, usar funções (*var_dump*)

para receber as saídas das variáveis, e assim implementar as mudanças. Encontrar um plugin de código aberto e sem licença para uso, que atingisse os objetivos do projeto e da empresa ServerDo.in.

Após as configurações do plugin, ainda é preciso confirmar a inexistência de uma abertura de segurança, na qual o usuário criado na plataforma WordPress, após excluido da base LDAP, não conseguirá realizar o *login*. Esta inexistência de uma possível falha se confirmou nos testes realizados.

4.2.4 WHMCS

A ferramenta administrativa WHMCS, por ser uma ferramenta que requer uma licença para uso, poucas soluções de código livre são disponibilizadas. Assim sendo, foi adquirido um plugin para realizar a integração com a base LDAP. Foi realizado a instalação e configuração, posteriormente habilitado para testes, e por fim colocado em produção. A interface do plugin pode ser vista na Figura 14.

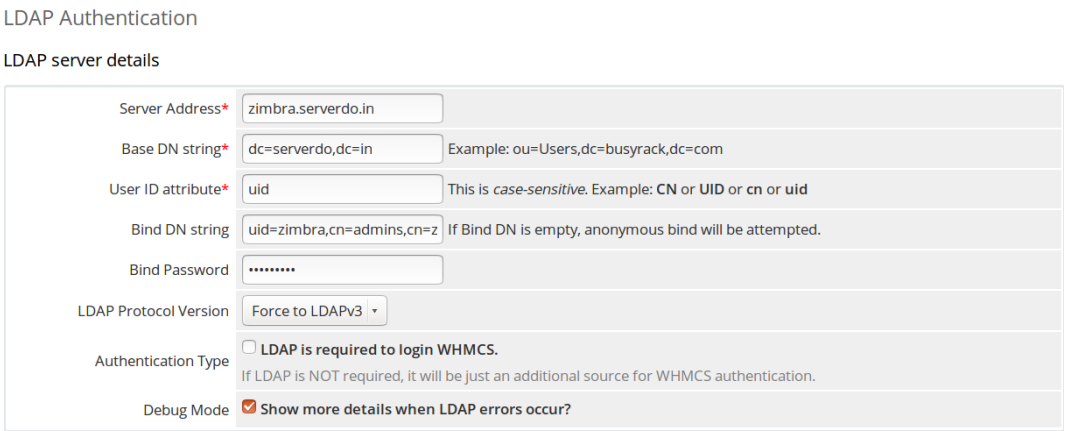


Figura 14: Plugin WHMCS - Desenvolvido por `busyrack.com`

Sendo uma ferramenta administrativa, somente usuários pré configurados como Administrador completo (*Full Administrator*) na ferramenta WHMCS conseguem logar usando o *backend* LDAP. Não sendo um plugin de código aberto, as modificações no mesmo não foram possíveis para diferenciar os acessos através de atributos, ou seja, a aplicação vai definir qual usuário LDAP terá acesso.

4.2.5 SSH - *Secure Shell*

Como previsto no projeto, a empresa ServerDo.in possui muitos servidores que são acessados somente através do serviço SSH. Embora cadastrar um usuário no servidor seja uma ação rápida, ela se torna dispendiosa quando o colaborador da empresa é desligado, sendo necessário retirar as permissões de acesso em cada servidor individualmente, desta forma, esta proposta para autentição via servidor LDAP tornou-se essencial para o escopo da empresa. Hoje, as consultas fazem as requisições de acesso backend ao servidor LDAP, tornando a ação de desligamento do colaborador muito prática.

Para que isso seja possível, é necessário instalar um cliente LDAP nas máquinas que se deseja conectar via *backend* LDAP. Os passos para instalação do cliente LDAP, assim como as configurações necessárias são descritas como seguem:

Para instalar o cliente LDAP:

```
# sudo apt-get install auth-client-config nscd

# apt-get install libnss-ldap (apontar para o servidor ldap)

# dpkg-reconfigure auth-client-config (caso necessite mudar configurações)
```

A seguir, segue nas Figuras 15 e 16 uma das configurações do assistente de instalação do cliente LDAP, realizadas no servidor chamado *us76*, nosso servidor de testes para o cliente.

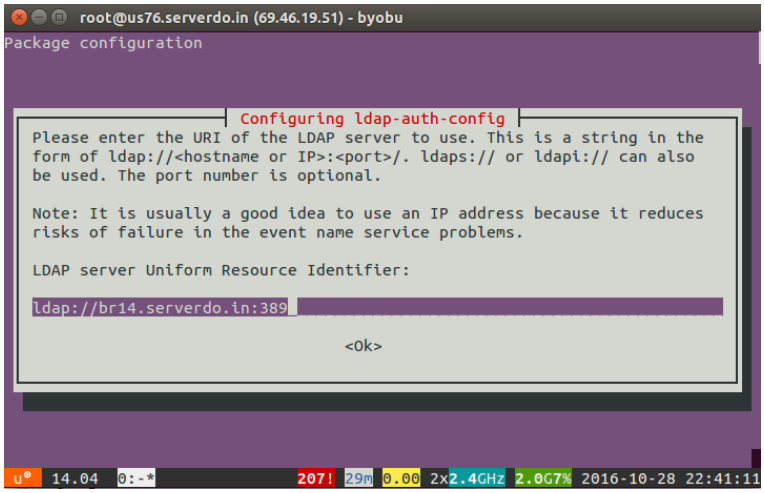


Figura 15: Configurando onde está localizado o servidor LDAP. Fonte: Própria.

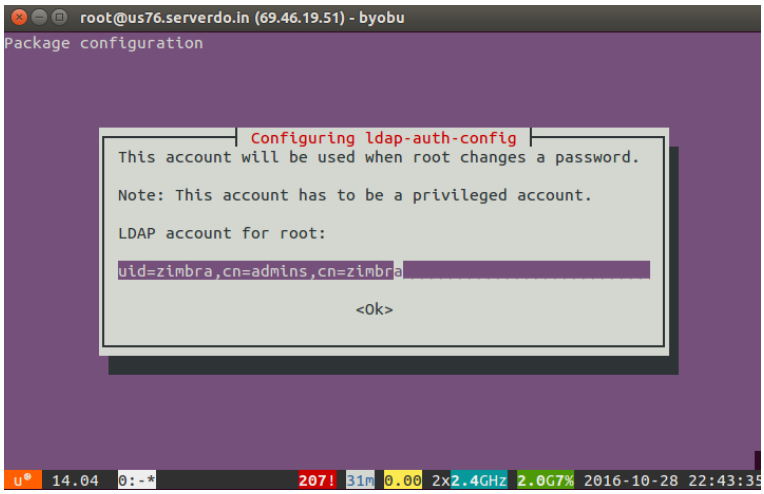


Figura 16: Configurando a conta *root* do servidor LDAP. Fonte: Própria.

As configurações do cliente se dividem em alguns arquivos, são eles:

/etc/ldap.conf

/etc/nsswitch.conf

/etc/pam.d/common-auth

/etc/pam.d/common-session

/etc/pam.d/common-account

/etc/pam.d/common-password

4.2.5.1 PAM - Módulos Anexáveis de Autenticação

O PAM, do inglês *Pluggable Authentication Module*, é uma API - Interface de Programação de Aplicativos (do inglês *Application Program Interface*) que cuida da autenticação de um usuário para um serviço. As configurações do PAM são geralmente implementadas no arquivo de configuração que residem em */etc/pam.d/*. Nos arquivos de configuração, foi necessário acrescentar o *pam_ldap.so*, que é um módulo PAM que usa um servidor LDAP para verificar direitos de acesso e credenciais do usuário.

4.2.5.2 NSS – Troca de nomes do serviço

NSS, do inglês *Name Server Switch*, segundo Linux Programmer's Manual [12], *nsswitch.conf* determina a ordem das buscas realizadas quando uma certa informação é requisitada (exemplo: senha e usuário), sendo tratada pela implementação do *Name Service Switch*. O arquivo de configuração segue:

$/etc/nsswitch.conf$

Onde foram alterados os seguintes campos:

$passwd : files\ ldap$
$group : files\ ldap$
$shadow : files\ ldap$

Estas configurações, informam ao sistema para procurar por usuário e grupo pela primeira vez nos arquivos padrões do Linux, seguido pelo ldap. Portanto, é especificado o nome do provedor de serviço, em nosso caso: LDAP.

Após configurações, foram realizados os testes de conexão. Para acompanhar os erros de autenticação, segue o arquivo de log:

$/var/log/auth.log$

Ao realizar o *login*, acompanhando os logs, vemos que:

$pam\_unix(sshd : auth) : authentication\ failure;\ logname = uid = 0euid = 0tty = sshruser = rhost = 189.4.75.167\ user = leandro$

$Accepted\ password\ for\ leandro\ from\ 189.4.75.167\ port\ 6178\ ssh2$

$pam\_unix(sshd : session) : session\ opened\ for\ user\ leandro\ by\ (uid = 0)$

É identificado através do *pam_unix*, que fornece autenticação de senha tradicional do */etc/passwd*, que nosso usuário não consegue realizar o *login*, porém, as configurações através do módulo pam_ldap permite que o usuário seja autenticado no servidor. Isto se deve ao fato das configurações estarem priorizando o *login* através do *pam_unix*, sendo possível ao administrador de rede alterar a ordem de autenticação, priorizando o *pam_ldap*.

Segue na Figura 17 , uma comparação com a autenticação *login* realizada pelo pam_unix

Todas configurações referentes ao SSH devem ser executadas com cuidado, pois de-

```
$ Accepted password for novousuario from 189.34.32.120 port 45766 ssh2
$ pam_unix(sshd:session): session opened for user novousuario by (uid=0)
$ New session 1 of user novousuario.
```

Figura 17: Acesso SSH na máquina cliente sem estar na base LDAP.

pendendo das alterações feitas nestes arquivos, pode ocorrer um bloqueio no servidor, perdendo o acesso ao servidor via SSH.

Foi alterado o arquivo:

/etc/pam.d/sshd

Adicionado a seguinte linha:

session required pam_mkhomedir.so skel = /etc/skel/ umask = 000

Aqui declaramos que ao abrir uma sessão de usuário, será criada um diretório */home/usuario* com todas as permissões, leitura, escrita e execução.

Dificuldades encontradas: No início, como já havia sido aplicada uma estrutura entre cliente e servidor LDAP, aplicado em máquinas virtuais, seguiu-se os mesmos conceitos para configuração no ambiente em produção, porém, como especificado anteriormente, o usuário criado na base Zimbra, não possui todas as classes necessárias para acesso via terminal. Os *logs* apontavam erros de autenticação, credenciais inválidas, não especificando exatamente a raiz que desencadeava este erro no processo. Testes de conexão foram executados para mitigar os problemas. E, por fim, identificado que a classe *posixAccount*, que define atributos como diretório pessoal, *shell*, número de usuário e grupo, não fazia parte do usuário. Assim sendo, o *objectClass posixAccount* torna-se essencial para o funcionamento desta aplicação, e será necessário ser adicionado em todos os usuários que terão permissões de acesso via SSH aos servidores.

4.2.5.3 Classe de objeto posixAccount

Segundo a RFC 2307[11], o sistema operacional UNIX e seus derivados necessitam de um meio para pesquisar entidades, combinando-as com critérios de pesquisa ou por enumeração. Essas entidades incluem usuários, grupos, serviços IP (que mapeiam nomes para portas IP e protocolos, e vice-versa), protocolos IP (que mapeiam nomes para números de protocolo IP e vice-versa), NIS netgroups, informações de inicialização (parâmetros de

inicialização e mapeamentos de endereço MAC), montagens de sistema de arquivos, hosts IP e redes.

A clase **posixAccount** é um diminutivo para uma conta com atributos Posix. Posix (*Portable Operating System interface*) por sua vez, segundo IEEE - *Standard for Information Technology* [10], define uma interface e um ambiente de sistema operacional padrão, incluindo um intérprete de comandos (ou "shell") e programas de utilidade comuns para suportar a portabilidade de aplicativos no nível do código-fonte.

Segue um exemplo de uma entrada com a classe **posixAccount**:

```
dn: uid=leandro,dc=serverdo, dc=in
objectClass: top
objectClass: account
objectClass: posixAccount
uid: leandro
cn: Leandro Usuário Teste
userPassword: cryptK0AyUubDrfOgO4s
loginShell: /bin/bash
uidNumber: 10
gidNumber: 10
homeDirectory: /home/leandro
```

Temos como atributos requeridos desta classe:

```
cn
gidNumber
homeDirectory
uid
uidNumber
```

E os atributos opcionais:

```
description
gecos
loginShell
userPassword
```

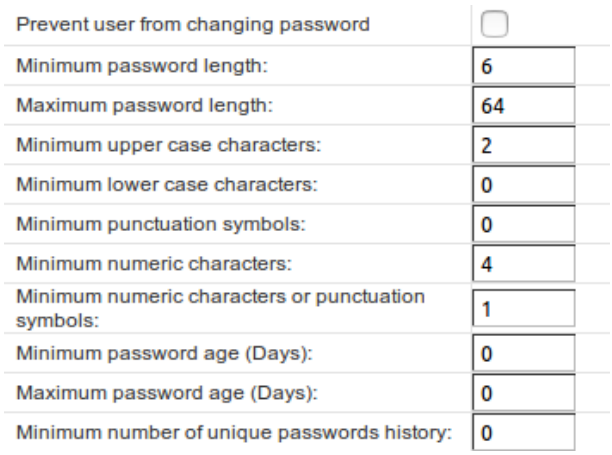
Todos os usuários LDAP com a classe **posixAccount** terão acesso via bash ao servidores da rede que utilizam LDAP para autenticação.

4.2.6 Políticas de senhas

Segundo Nakamura (2007, p.204) [23], "a provisão de senhas pelos administradores de sistema e a utilização de senhas pelos usuários é uma parte específica da política de segurança, de grande importância para as organizações.". Com o atributo *pwdPolicySubentry*, segundo Gil [22], "é possivel criar grupos de políticas de senha diferentes para atender as exigências diferentes. Este atributo permite que se defina qual a política de senha que será utilizada neste objeto."

Como a empresa ServerDo.in não viu uma necessidade neste tipo de implementação, foram configurados uma única política de senha para todas as contas. Portanto, como discutido no subcapítulo 3.1, tendo o *backend* LDAP do Zimbra como plataforma das credenciais, segue as políticas de senhas definidas: Entrada mínima de números, obrigatório o uso de caracteres especiais e tamanho mínimo e máximo da senha.

A alterações foram feitas através do próprio Zimbra, como segue na Figura 18, mas que poderiam ser configuradas através da ativação do módulo chamado *ppolicy.* Este módulo ou *schema* por sua vez, poderia ser configurado da mesma maneira através de suas classes de objeto e atributos.



Prevent user from changing password	☐
Minimum password length:	6
Maximum password length:	64
Minimum upper case characters:	2
Minimum lower case characters:	0
Minimum punctuation symbols:	0
Minimum numeric characters:	4
Minimum numeric characters or punctuation symbols:	1
Minimum password age (Days):	0
Maximum password age (Days):	0
Minimum number of unique passwords history:	0

Figura 18: Alterações nas políticas de senha.

Conforme o ZIMBRA (2016) [28], pode-se exigir que os usuários criem senhas fortes para proteger a estrutura. Os usuários podem ser bloqueados de suas contas se eles não conseguirem entrar após um número máximo de tentativas configuradas.

5 *Conclusão*

O OpenLDAP juntamente com o protocolo LDAP, desperta interesse para administradores de rede, onde é comtemplado e solucionado muitos problemas na administração de sua infraestrutura. Como citado e implementado neste trabalho, a ferramenta consegue atender todas às aplicações que possuem um módulo para integração, visando uma centralização do sistema de autenticação, tornando-se uma solução interessante na área de Tecnologia da Informação.

A base central adaptada do Zimbra, com o uso de uma política de esquemas, classes e atributos, atendeu os serviços heterogêneos sem interferências entre si, tornando a infraestrutura mais adequada para o gerenciamento, facitando ações de controle e processos dentro da empresa.

É imprescindível que o administrador da infraestrutura compreenda os esquemas e atributos para dar acesso de *login* ou negar este acesso em cada aplicação. Como por exemplo: É necessário bloquear o usuário *leandro* para fazer *login* nos servidores, portanto, precisa-se retirar o *objectClass posixAccount*. Se o usuário *leandro* não pode acessar o WordPress, é retirado o CN=wordpress.

Na empresa ServerDo.in, foi acelerado os processos que, em sua grande parte, tomavam tempo dos administradores da infraestrutura. A utilização do Zimbra e sua base LDAP, permitiram que testes fossem executados em um ambiente real de produção, aproveitando um serviço já disponível da empresa.

5.1 Trabalhos futuros

Depois da integração, foi possível perceber que algumas das modificações precisaram ser feitas de uma maneira manual, o que acaba por também dispender tempo. Dessa forma, foi pontuado alguns trabalhos futuros, de modo a automatizar estas operações:

1. Modificar a arquitetura do Zimbra, de forma a alterar o *schema* automático, incluindo os atributos ou classes necessárias para acessar as aplicações, cn=wordpress e *posixAccount* para usuários que terão acesso ao *admin* WordPress e SSH respectivamente.

2. Fazer com que cada novo servidor criado, durante a instalação base do mesmo, já execute a instalação do cliente LDAP, poupando tempo do administrador de rede, através da criação de *templates* e modelos de VMs, de forma que estas já estejam com as bibliotecas pré-configuradas para acesso através do LDAP, dispensando uma configuração manual.

3. Aumentar a segurança ao acessar o phpLDAPadmin através do *browser*, forçando um *login* de confirmação antes de liberar acesso à interface. Integrar também o phpLDAPadmin ao Apache já existente, removendo a necessidade de ser abrir uma nova porta para acesso.

Referências

[1] MENEGUITE, R.L. *LDAP – Autenticação Centralizada.* Dissertação - Faculdades Unificadas Doctum de Cataguases, MG - Brasil, 2009.

[2] DONLEY, C. *LDAP Programming, Management and Integration.* 74. ed. Greenwich - Ct: Manning, 2003. 352 p.

[3] WAN, Xin; SCHULZRINNE, Henning; KANDLUR, Dilip; Verma, D. *Measurement and Analysis of LDAP Performance.* [s. L.]: Ieee/acm Transactions On Networking, 2008. 12 p.

[4] BUTCHER, Matt. *Mastering OpenLDAP: Configuring, Securing, and Integrating Directory Services.* Birmingham - Mumbai: Packt, 2007. 482 p.

[5] TUTTLE, Steven; EHLENBERGER, Ami; GORTHI, Ramakrishna; LEISERSON, Jay; MACBETH, Richard; OWEN, Nathan; RANAHANDOLA, Sunil; STORRS, Michael; YANG, Chunhui. *Understanding LDAP: Design and Implementation.* 2. ed. [s. L.]: Ibm Tivoli Directory Server, 2004. 774 p.

[6] MACHADO, Erich Soares; MORI JUNIOR, Flavio da Silva. *Autenticação Integrada Baseada em Serviço de Diretório LDAP.* 2006. 84 f. Monografia - Curso de Ciência da Computação, Ime, Universidade de São Paulo, São Paulo, 2006. Disponível em: `https://www.linux.ime.usp.br/~cef/mac499-06/monografias/erich/monografia.pdf`. Acesso em: 06 Dezembro 2016.

[7] EVARISTO, Lincon Ruam Angioletti. *Integrando a base de usuários LDAP com serviços de e-mail, proxy web e ssh.* 2002. 51 f. Tese (Doutorado) - Curso de Sistemas de Informação, Instituto Superior Tupy, Joinville, 2008.

[8] TRIGO, Clodonil Honório. *OpenLdap: Uma Abordagem Integrada.* São Paulo: Novatec, 2007. 240 p.

[9] CHAVES, Tiago Rodrigues. *Autenticação IEEE 802.1X baseada no protocolo RADIUS e serviço de diretório LDAP aplicado a rede GIGAUFOPNET.* 2010. 110 f. Monografia - Curso de Ciência da Computação, Instituto de Ciências Exatas, Universidade Federal de Ouro Preto, Ouro Preto, 2010.

[10] 1003.1-2008 - IEEE Standard for Information Technology - Portable Operating System Interface (POSIX(R)). Disponível em: `https://standards.ieee.org/findstds/standard/1003.1-2008.html` Acesso em: 26 Novembro 2016.

[11] L, Howard. *An Approach for Using LDAP as a Network Information Service.* 1988. Disponível em: `https://www.ietf.org/rfc/rfc2307.txt` Acesso em: 26 Novembro 2016.

[12] Kerrisk, Michael. *Linux Programmer's Manual.* 2016. Disponível em: `http://man7.org/linux/man-pages/man5/nsswitch.conf.5.html`. Acesso em: 28 Novembro 2016.

[13] T, Howes; W, Yeong; S, Kille. *X.500 Lightweight Directory Access Protocol.* 1993. Disponível em: `https://tools.ietf.org/html/rfc1487` Acesso em: 15 Julho 2016.

[14] J. Sermersheim. *Lightweight Directory Access Protocol (LDAP): The Protocol.* 2016. Disponível em: `https://tools.ietf.org/html/rfc4511` Acesso em: 10 Janeiro 2006.

[15] S. Kille; M. Wahl; A. Grimstad; R. Huber; S. Sataluri. *Using Domains in LDAP/X.500 Distinguished Names.* 1998. Disponível em: `https://tools.ietf.org/html/rfc2247` Acesso em: 10 Janeiro 2017.

[16] M, Wahl; T, Howes; S, Kille. *Lightweight Directory Access Protocol (v3)* 1997. Disponível em: `https://www.ietf.org/rfc/rfc2251.txt`. Acesso em: 15 Julho 2016.

[17] OpenLDAP. Disponível em: `http://www.openldap.org`. Acesso em: 29 Novembro 2016.

[18] BARRETT, J.Daniel; SILVERMAN, E.Richard; BYRNES, G.Robert. *SSH, the Secure Shell: The Definitive Guide.* 2. ed. 2005. Published By O'Reilly Media, 645 p.

[19] WHMCS. Disponível em: `http://whmcs.com`. Acesso em: 29 Novembro 2016.

[20] ZIMBRA. Disponível em: `http://www.zimbra.com`. Acesso em: 29 Novembro 2016.

[21] RESNICK, Marty; TOUITOU, David. *Zimbra - Implement, Administer and Manage.* 2007. Publicado por: Packt Publishing.

[22] Gil, A.P. *OpenLDAP Extreme.* 2012, p.90. Editora Brasport Livros e Multimídia Ltda. Acesso em: 30 Novembro 2016.

[23] Nakamura, E.T; Geus, P.L. *Segurança de Redes em Ambientes Cooperativos.* 2007, p204. Editora Novatec. Acesso em 30 Novembro 2016.

[24] Price, Jason. *SQL: Domine SQL e PL/SQL no banco de dados Oracle.* 2009, Editora Artmed. Acesso em: 30 Novembro 2016.

[25] WORDPRESS. Disponível em: `https://br.wordpress.org/`. Acesso em: 1 Dezembro 2016.

[26] MIchigan University. Disponível em: `https://www.umich.edu/`. Acesso em 15 Abril 2016.

[27] Hedengren, D. Thor. *Smashing WordPress - Beyond the Blog.* 2012. Editora Bookman.

[28] Zimbra Collaboration. *Administrator Guide.* 2016. Disponível em: `https://www.zimbra.com/documentation/`. Acesso em: 04 Dezembro 2016.

[29] RODRIGUES, V.V.Carlos. *Migração e integração de serviços de e-mail, autenticação de usuários e servidor de diretórios.* Monografia - Departamento de Ciência da Computação da Universidade Federal de Lavras - MG - Brasil, 2008. Disponível em: `http://repositorio.ufla.br/bitstream/1/5606/1/MONOGRAFIA_Migra\%C3\%A7\%C3\%A3o\%20e\%20integra\%C3\%A7\%C3\%A3o\%20de\%20servi\%C3\%A7os\%20de\%20e-mail,\%20autentica\%C3\%A7\%C3\%A3o\%20de\%20usu\%C3\%A1rios\%20e\%20servidores\%20de\%20diret\%C3\%B3rios.pdf`. Acesso em: 06 Dezembro 2016.

[30] CARTER, Gerald. *LDAP System Administration.* Publicado por: O'Reilly Media. 2003. First Edition

[31] HELMKE, Matthew. Hudson, Andrew. Hudson, Paul. *Ubuntu Unleashed.* Publicado por: Pearson Education. 2015. 875p.

[32] GIL, de Paula, Anahuac. *OpenLDAP Extreme.* Rio de Janeiro: Brasport Livros e Multimidia, 2012. 255 p.

APÊNDICE A

A.0.1 Criando usuários pelo terminal.

Para adicionar os usuários na árvore de diretórios LDAP, existe um arquivo padrão de diretórios com a extensão .ldif, que é um arquivo de dados/entradas associado ao LDAP. Este *arquivo.ldif* é adicionando ao slapd.conf através do comando *slapadd*. Para administrar as contas do usuário através do Zimbra, a ferramenta *'zmprov'* executa todas as tarefas de aprovisionamento no Zimbra LDAP, segue um exemplo adicionando um usuário chamado ifsc:

```
# zmprov ca ifsc@serverdo.in *senha* displayName InstitutoFederal
```

uid=ifsc	
dn	uid=ifsc,ou=people,dc=serverdo,dc=in
cn	InstitutoFederal
displayName	InstitutoFederal
Email	ifsc@serverdo.in
objectClass	inetOrgPerson zimbraAccount amavisAccount
sn	ifsc
User Name	ifsc
Password	**********
zimbraAccountStatus	active
zimbraCreateTimestamp	20161123221138Z
zimbraId	2e96b0a7-63b8-4c92-94f0-94a9686ecec8
zimbraMailDeliveryAddress	ifsc@serverdo.in
zimbraMailHost	br14.serverdo.in
zimbraMailStatus	enabled
zimbraMailTransport	lmtp:br14.serverdo.in:7025
zimbraPasswordModifiedTime	20161123221138Z

Figura 19: Atributos do usuário uid=ifsc, vistas pelo phpLDAPadmin.

Ao criar a conta do usuário, é identificado pela Figura 19, que muitos atributos são adicionados ao gerar uma única conta, pois esta nova conta é vinculada há um *schema* pré definido pelo Zimbra.

$zimbraId$,

Identificador de sistema único do Zimbra

$zimbraCreateTimestamp$

O dia em que o objeto foi criado.

$zimbraAccountStatus$

Apresenta o status da conta do usuário, ou seja, se esta conta está ativa ou bloqueada.

Pelo terminal, por sua vez, pode ser executado o seguinte comando:

```
# zmprov ga ifsc@serverdo.in
```

Trazendo todas as informações do usuário (objetos de classe e atributos) pelo terminal (ga = getAccount).

Para se excluir uma conta da base de dados:

```
# zmprov da ifsc@serverdo.in
```

Para mais detalhes e comandos, segue a documentação do Zimbra, que se encontra em: `https://wiki.zimbra.com/wiki/Zmprov`

A.0.2 Comandos úteis do LDAP utilizados nas máquinas clientes - SSH

Nos servidores clientes, que se pretende acessar via SSH, para identificar que às configurações do PAM e NSS foram corretamente configuradas, os comandos abaixo podem ser executados:

```
# ldapsearch -x -b dc=serverdo,dc=in -h zimbra.serverdo.in -W -D uid=zimbra,cn=admins,cn=zimbra
```

Listando todas os usuários da conta de administrador principal do Zimbra. Com isso, temos como saída as contas vinculadas à base dc=serverdo,dc=in, provando que o servidor que se quer acessar via SSH, está se comunicando com a base LDAP localizada em `zimbra.serverdo.in`.

O comando do Unix *getent passwd*, que armazena informações dos usuários, também se torna útil, pois trará como saída aqueles que possuem acesso ao servidor, ou seja, após as configurações da seção 4.2.5, todos os usuários com a classe *posixAccount* devem aparecer listados, como mostra a Figura 20:

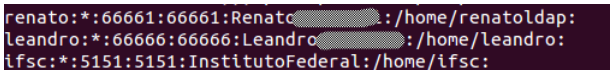


Figura 20: Saída do comando *getent passwd*