Integrando LDAP e Samba

By admin on 8 de junho de 2012 in News

Dei uma olhada pela internet, e vi muitos tutoriais mostrando como fazer a integração desses dois ambientes. Mas a grande maioria, não funcionava, e quase todos se esqueciam de um detalhe primordial: Todos os usuários do samba devem ter um mapeamento, ou um usuário no Linux também.

Apesar de que os tutoriais ensinarem como se integra esses ambientes, sem a conta do Unix, o samba não permite a criação do usuário na base LDAP. Então neste caso devemos usar o próprio LDAP como sistema de autenticação pro Linux, dessa forma as exigências do samba são cumpridas.

O grande trabalho ai é integrar esses três ambientes: Linux (PAM) Samba e LDAP.

Neste Post mostrarei como fazer isso.

Usei neste exemplo um Debian Lenny, samba padrão da versão, OpenLDAP também padrão. Como repositório usei "**deb http://archive.debian.org/debian lenny main contrib non-free".** Basta adicionar ao sources.list e executar o apt-get update, para atualizar a base.

Então vamos botar a mão na massa.

Configurando o Linux e seus pacotes

Como disse anteriormente o Debian utilizado nessa versão é o Debian Lenny, pois a partir do Debian 6, o pacote LDAP quando instalado, já vêm com a versão nova do OpenLDAP, que muda totalmente a forma de configuração.

Quais pacotes serão instalados:

- apache2-suexec libapache2-mod-php5 php5 php5-cli php5-curl php5-gd php5-imap php5-ldap php5-mcrypt php5-mhash php5-sqlite php5-tidy php5-xmlrpc php-pear slapd mcrypt ldap-utils libgd-tools apache2-doc libpam-ldap libnss-ldap resolvconf samba swat smbclient smbfs smbldap-tools

Podemos ver que temos muitos pacotes ai, por exemplo, Apache, PHP, phpldapadmin, swat, que não são tão necessários. Deste acima os mais importantes são: **slapd, ldap-utils, libpam-ldap, libnss-ldap, samba, smbldap-tools.** Esses pacotes incluem o servidor LDAP, ferramentas LDAP, cliente PAM Ldap, samba e ferramentas samba LDAP.

Quando da instalação serão solicitados muitas informações, que podem ser preenchidas normalmente sem preocupação, pois durante a configuração dos serviços, estaremos alterando-os novamente.

Para esse exemplo, use o domínio **"dominiolinux.net" ,** onde o administrador é o **admin** e senha **debian.**

Como cliente usei um Windows XP, não tive tempo ainda de testar com o Windows 7, mas assim que o fizer , atualizarei esse mesmo Post.

Devemos também alterar o arquivo de configuração **/etc/fstab,** para suportar permissões especificas através dos módulos user_xattr, acl e relatime em nosso sistema de arquivos que será utilizado pelo samba para compartilhamento.

O valor **relatime**, vêm em contrapartida ao atime, para sistemas Linux ext2, ext3, etc. O sistema habilitado com essa opção causa acesso para atualização antes da própria modificação, ganhando em performance. Já é padrão no Ubuntu.

A opção **user_xattr,** permitirá que os arquivos e diretório, o uso de permissões Estendidas. Cada atributo estendido é um par como atributo/valor associado. Por exemplo, poderíamos criar um atributo chamado user, e dentro desse atributo adicionar vários valores, tais como dono, artigo, etc. A forma de adicionar isso é através do **setfattr** e para visualizar o comando **getfattr,** instale o pacote **attr.** Veja abaixo um exemplo:

```
setfattr -n user.dono -v "Andre" arquivo.txt

setfattr -n user.artigo.titulo -v "Permissoes" arquivo.txt

setfattr -n user.artigo.autor -v "Andre" arquivo.txt



getfattr -d arquivo.txt

# file: arquivo.txt

user.artigo.autor="Andre"

user.artigo.titulo="Permissoes"

user.dono="Andre"
```

Já a ACL (Access Control List) vai expandir o formato padrão de permissões que temos no Linux. O famoso UGO, para uma coisa mais próxima do NT. A Acl é composta por uma ou mais Entradas de Controle de Segurança (ACE), e quando do

acesso serão testados as permissões de negação e posteriormente as permissões de permissão. Deve-se também instalar o pacote **acl,** para pode manipular como cliente. Veja abaixo outro exemplo:

```
setfacl –m g:grupo:rw arquivo.txt



getfacl arquivo.txt

#file: arquivo.txt

#owner: root

#group: root

user: rw-

group: r–

group:adm:rw-

other: r–
```

Como podemos ver acima, agora temos dois grupos, com permissões distintas. Para habilitar isso ao sistema devemos editar o fstab, e adicionar o suporte a atributos e acl, na partição que será usado para compartilhamento, no caso abaixo, está sendo habilitada na raiz do Linux

```
/dev/sda1  /  ext3 relatime,user_xattr,acl,erros=remount-ro   0  1
```

Não é necessário reiniciar o servidor, basta executar o comando mount, com a opção de remount.

```
mount –o remount /
```

Podemos confirmar se já está habilitado , executando o comando mount sem opções:

```
mount

/dev/sda1 on / type ext3 (rw,relatime,user_xattr,acl,errors=remount-ro)
```

Agora , começaremos a configurar nossos serviços, o primeiro será o próprio LDAP.

Configurando slapd

O slapd é o servidor OpenLDAP. Quando instalamo-lo anteriormente foram solicitadas algumas informações, agora nesse momento iremos reconfigurar. O primeiro passo é apagar a base de dados existente, que foi configurado no momento da instalação, reiniciar o serviço e executar o configurador.

```
rm –rf /var/lib/ldap/*

invoke-rc.d slapd stop

dpkg-reconfigure slapd
```

Serão solicitadas algumas informações, essas devem ser preenchidas conforme abaixo :

- Omitir configuração do servidor OpenLDAP? **No**
- Nome de domínio DNS: **dominiolinux.net**
- Nome da organização: **dominiolinux.net**
- Senha do administrador: **debian**
- Banco de Dados Beckend: **HDB**
- Você deseja remover o pacote slapd? **No**
- Permite protocolo LDAPv2? **No**

Pronto já esta configurada, a primeira parte.

Iremos ainda já colocar as configurações do samba. Devemos adicionar o schema do samba, bem como as regras de acesso:

```
zcat /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz > /etc/ldap/schema/samba.schema
```

O comando acima descompacta o schema do samba na pasta /etc/ldap/schemas.

Depois deverá ser adicionado ao arquivo de configuração do slapd através da diretiva **include.**

Vamos gerar agora a senha para o administrador **admin ,** da base LDAP.

```
slappasswd –h MD5

{MD5}bpVSyb2OYcjyd8ISIBYCNA==
```

A senha exibida abaixo do comando , deverá ser colocada na diretiva **rootpw,** logo após da diretiva **rootdn.**

Devemos então configurar o slapd.conf localizado dentro de /etc/ldap. Veja abaixo os itens a serem alterados:

```
… <omitido>

include /etc/ldap/schema/samba.schema

…

rootdn          “cn=admin,dc=dominiolinux,dc=net”
rootpw          {MD5}Qhz9FD5FDD9YFKBJVAngcw==

…

…

index objectClass                       eq,pres
index ou,cn,sn,mail,givenname           eq,pres,sub
index uidNumber,gidNumber,memberUid     eq,pres
index loginShell                        eq,pres

…

…

index displayName                       pres,sub,eq
index nisMapName,nisMapEntry            eq,pres,sub
index sambaSID                          eq
index sambaPrimaryGroupSID              eq
index sambaDomainName                   eq
index default                           sub
index uniqueMember                      eq
index sambaGroupType                    eq
index sambaSIDList                      eq
```

```
…

attrs=userPassword,sambaNTPassword,sambaLMPassword,sambaPwdMustChange,sa
mbaPwdLastSet
by self write
by anonymous auth
by * none

access to attrs=shadowLastChange,shadowMax
by self write
by * read



access to *
by * read

…
```

Sugiro que copie o meu arquivo, fazendo download dele, logo abaixo e alterando somente o que for referente ao seu domínio, como os itens rootpw, rootdn, suffix, etc. Este itens ainda serão vistos em outros Post , tratando somente de LDAP e seu funcionamento**. Veja a série de Posts LDAP.**

Download do arquivo slapd.conf

Depois de feito download, altere as entradas que fazem referencia ao domínio, como **suffix, rootdn** e a entrada da senha **rootpw.**

Após esse processo basta a nós iniciar o serviço. Mas antes de iniciarmos devemos para o serviço nscd, que é responsável por fazer cache de resolução de nomes (NameService Caching Daemon), ele costuma fazer cache das consultar/resultados providos pelo NSS (/etc/nsswitch.conf). Abaixo todos os comandos executados:

```
rm –rf /var/lib/ldap/*

invoke-rc.d nscd stop

chown –Rf openldap.openldap /var/lib/ldap

slapindex

invoke-rc.d slapd start
```

Após iniciado o servidor, já é possível fazer consultas na base, basta executar o comando **slapcat .** Algo como abaixo deverá aparecer:

```
dn: dc=dominiolinux,dc=net

objectClass: top

objectClass: dcObject

objectClass: organization

o: dominiolinux.net

dc: dominiolinux

structuralObjectClass: organization

entryUUID: 63670f50-4429-1031-98e3-930c88793bcc

creatorsName:

createTimestamp: 20120606134348Z

entryCSN: 20120606134348.586655Z#000000#000#000000

modifiersName:

modifyTimestamp: 20120606134348Z


dn: cn=admin,dc=dominiolinux,dc=net

objectClass: simpleSecurityObject

objectClass: organizationalRole

cn: admin

description: LDAP administrator

userPassword:: e2NyeXB0fXlvWkQzTGtxNDMxY2s=

structuralObjectClass: organizationalRole
```

```
entryUUID: 6369cef2-4429-1031-98e4-930c88793bcc

creatorsName:

createTimestamp: 20120606134348Z

entryCSN: 20120606134348.604772Z#000000#000#000000

modifiersName:

modifyTimestamp: 20120606134348Z
```

Com isso nosso servidor LDAP já está pronto, devemos agora configurar o Samba,

Configurando o Samba

Na integração com o samba, precisamos configurar várias features para suportar desde a consulta a base LDAP, até criação de usuários, conta de máquinas, grupos ,etc.

O servidor samba está atuando com a Role DOMAIN_PDC. Neste tipo de configuração , será usado o padrão PDC de um domínio, mas o Beckend de usuários e senha será usado o do LDAP.

Vejamos as alterações necessárias:

```
…

workgroup = DOMINIOLINUX

realm = DOMINIOLINUX.NET

…

…

passdb backend = ldapsam:ldap://127.0.0.1/
pam password change = Yes
passwd program = /usr/sbin/smbldap-passwd -u %u
passwd chat = *New*password* %nn *Retype*new*password* %nn
*all*authentication*tokens*updated*
unix password sync = Yes
```

```
…

add user script = /usr/sbin/smbldap-useradd -m %u
delete user script = /usr/sbin/smbldap-userdel %u
add group script = /usr/sbin/smbldap-groupadd -p %g
delete group script = /usr/sbin/smbldap-groupdel %g
add user to group script = /usr/sbin/smbldap-groupmod -m %u %g
delete user from group script = /usr/sbin/smbldap-groupmod -x %u %g
set primary group script = /usr/sbin/smbldap-usermod -g %g %u
add machine script = /usr/sbin/smbldap-useradd -w %u

…

preferred master = Yes
domain master = Yes

domain logons = Yes
os level = 65

wins support = Yes

…
ldap admin dn = cn=admin,dc=dominiolinux,dc=net

ldap delete dn = Yes
ldap group suffix = ou=group
ldap idmap suffix = ou=idmap
ldap machine suffix = ou=computer
ldap suffix = dc=dominiolinux,dc=net
ldap ssl = no
ldap user suffix = ou=people

…
```

Vamos ver as configurações realizadas no servidor. No primeiro grupo temos as entradas **workgroup** e **realm,** onde deverão ser informados o nome do domínio, e o nome do domínio FQDN, ou seja, completo, neste segundo **realm.**

Na segunda parte temos as configurações de backend (senhas) do servidor, em **passdb beckend,** é informado o tipo de autenticação e onde, no caso em nosso próprio servidor LDAP, outras opções como **pam password change, passwd program, unix password sync,** fazem referencias como serão tratados trocas de senha, permitindo alteração via pam, o programa que será usado para troca de senha (smbldap-passwd) e se serão sincronizados com as contas Unix (que futuramente serão o próprio LDAP).

Já no terceiro grupo, informamos qual o caminha das ferramentas e opções pra criação de usuários , grupos, maquinas, bem como deleção, entre outras coisas. Todos os comandos usados são das ferramenta smbldap-tools.

Por fim, devemos passar as informações sobre nosso domínio, tal como quem é o administrador LDAP (**ldam admin),** quais serão as Unidades Organizacionais para Grupos, Computadores e Pessoas (**group,computer,people),** e por fim , não será usado autenticação com criptografia **ssl.**

Da mesma forma que o anterior, sugiro que seja copiado o meu arquivo smb.conf, e alterados os itens referentes ao domínio tanto do Samba como Ldap.

Download do arquivo smb.conf

Veja que não coloquei configurações sobre share (compartilhamento), mais isso pode ser definido por você mesmo. Mas preste atenção no meu arquivo que faz referencia para os compartilhamentos **netlogon, profile e public.** Para deixar de acordo deverão ser executados os comandos abaixo para criação e alteração de permissão.

```
mkdir -p /var/lib/samba/netlogon /var/lib/samba/profiles
chown -Rf root:root /var/lib/samba/netlogon /var/lib/samba/profiles
chmod 1777 /var/lib/samba/profiles
```

Ainda sim, no meu arquivo de configuração o esquema de salvar profile no servidor está desabilitado, me :

logon path =

Caso queira que seja feito perfil móvel, deverá colocar o caminho onde o usuário deverá gravar seu perfil no servidor. Segundo nosso smb.conf será em \%Nprofiles%U

então dessa forma a entrada do arquivos deverá ficar da seguinte forma:

logon path = \%Nprofiles%U

Antes de iniciarmos propriamente, devemos cadastrar senha do LDAP que será usado pelo samba:

smbpasswd –w debian

Só agora podemos então iniciar o serviço do samba:

invoke-rc.d samba restart

Configurando a Ferramenta smbldap-tools

O smbldap-tool têm vários scripts, que nos ajudarão na tarefa de criar usuários e grupos, e até popular nossa base LDAP com os Grupos Padrões e Unidades Organizacionais que iremos usar, tal como People, entre outras. Para configurarmos essa ferramentas deveremos editar dois arquivos o **smbldap.conf** e o **smbldap_bind.conf.**

Um deles terá informações sobre as configurações das nossa base Ldap, tal como diretório pessoal, shell padrão, o campo Gecos, Profile, Home Drive (Mapeamento), script logon, entre outras itens , todos relacionados à criação de usuário, neste caso o arquivo **smbldap.conf.** Já o segundo , trará informações sobre o administrador e senha do LDAP.

Vejamos o primeiro, smbldap.conf:

```
#Configurações Globais

SID=”S-1-5-21-3432973329-3508354683-4054472345″

sambaDomain=”DOMINIOLINUX”


#Configuração LDAP

slaveLDAP=”127.0.0.1″

slavePort=”389″

masterLDAP=”127.0.0.1″

masterPort=”389″

ldapTLS=”0″

verify=”require”

cafile=”/etc/smbldap-tools/ca.pem”

clientcert=”/etc/smbldap-tools/smbldap-tools.pem”

clientkey=”/etc/smbldap-tools/smbldap-tools.key”
```

suffix=”dc=dominiolinux,dc=net”

usersdn=”ou=people,${suffix}”

computersdn=”ou=computer,${suffix}”

groupsdn=”ou=group,${suffix}”

idmapdn=”ou=idmap,${suffix}”

sambaUnixIdPooldn=”sambaDomainName=${sambaDomain},${suffix}”

scope=”sub”

hash_encrypt=”MD5″

crypt_salt_format=”%s”

#Configurações de Conta Linux

userLoginShell=”/bin/bash”

userHome=”/home/%U”

userHomeDirectoryMode=”700″

userGecos=”System User”

defaultUserGid=”513″

defaultComputerGid=”515″

skeletonDir=”/etc/skel”

defaultMaxPasswordAge=”365″

#Configuração Samba

userSmbHome=””

userProfile=””

userHomeDrive=”U:”

userScript=”logon.bat”

```
mailDomain=”dominiolinux.net”

with_smbpasswd=”0″

smbpasswd=”/usr/bin/smbpasswd”

with_slappasswd=”0″

slappasswd=”/usr/sbin/slappasswd”
```

O primeiro item , um dos mais importante é o SID (Security Identifier), que é o número de identificação do nosso Samba , que fará a comunicação do samba com LDAP. Esse valor deverá ser obtido com o comando **net getlocalsid,** veja abaixo a saída:

```
net getlocalsid

SID for domain DEBIAN is: S-1-5-21-3432973329-3508354683-4054472345
```

Após rodar o comando a saída trará o valor do SID que deverá ser colocado no arquivo acima.

O arquivo acima, foi dividido em 4 partes para uma melhor compreensão. Temos Configurações Globais, Configurações LDAP, Configurações de Conta Linux e Configurações do Samba. Muito intuitivo né.

A primeiro parte configurações globais, é onde iremos colocar o valor do **SID** e o domínio configurado no Samba em **sambaDomain.**

Nas configurações LDAP, devemos informar o IP do servidor, tanto em **slaveLDAP ,** como **MasterLDAP,** já que estamos usando o mesmo servidor, ambos terão o mesmo endereço local no caso 127.0.0.1. Também informamos a porta do master e do slave, como anteriormente são os mesmos, e por isso a mesma porta. Um pouco mais a frente temos as configurações de criptografia e certificado (**ldapTLS, verify, cafile,cliencert, clientkey)** , que em nosso caso não serão usadas. Fica claro na diretiva **ldapTLS=0.** Ainda neste mesma seção temos o **suffix, usersdn, computersdn, groupsdn, idmapdn** que informa qual o nosso domínio LDAP. e qual serão os dn (distiguished name) para os usuários e grupos. Por exemplo quando cadastrarmos um usuário joao, seu dn será **dn: uid=joao,ou=people,dc=dominiolinux,dc=net,** que será atribuído automaticamente através destas diretivas.

Nas próximas diretivas **sambaUnixPooldn,** é informado onde serão armazenados os novos usuários e grupos. O escopo também é informado através da diretiva scope. E por

fim os dois últimos estão relacionados com a criptografia de senha, no caso usamos MD5, padrão Linux.

A terceira seção é tratada como se fosse o arquivo login.defs no Linux, onde informa as configurações para criação de usuário. Temos **userLoginShell,** para o shell padrão, **userHome,** para o diretório pessoal do usuário, e assim sucessivamente até tempo para expirar uma senha.

Por fim configurações do samba em **userSmbHome, userProfile , userHomeDrive, userScript,** entre outros. Onde serão utilizados para cada usuário, por exemplo, neste caso usaremos o script logon.bat, para mapeamento. Caso queira diferencia pode-se alterar essa configuração conforme necessidade.

Deixo abaixo o arquivo que usei para download:

Download do arquivo smbldap.conf

Obs.: É interessante que antes de gerar o SID, altera o nome da máquina (hostname) e cadastre-a no /etc/hosts, para corresponder ao valor informado no smb.conf em "netbios name" se usado. Os valores para userSmbHome, profile e script logon serão usados o próprio nome da máquina, que em caso chama-se debian.

Com esse arquivo pronto, vamos para o segundo, que é muito menor e bem mais fácil de fazer. Edite e altere o arquivo **smbconf_bind.conf** , como o arquivo abaixo, mudando para seu próprio domínio e senha.

```
slaveDN="cn=admin,dc=dominiolinux,dc=net"
slavePw="debian"
masterDN="cn=admin,dc=dominiolinux,dc=net"
masterPw="debian"
```

Devemos ajustar as permissões para evitar maiores problemas:

```
chmod 0644 /etc/smbldap-tools/smbldap.conf

chmod 0600 /etc/smbldap-tools/smbldap_bind.conf
```

Enfim, agora que tudo está OK (Certifique-se disso), podemos popular nossa base LDAP, usando uma das ferramentas smbldap-tools.

```
smbldap-populate
```

Será solicitado a senha do administrador ao final do script. Se todos os arquivos estiverem corretos, ao final desse processo serão criados vários grupos como Domain User, Domain Admins, Domain Guests, Domain Computers , Administrators, Account Operators , Print Operators , Backup Operators , Replicators e ainda as OU people, group, computers, etc.

Com um simples **slapcat,** já é possível ver todas essas entradas em nossa base. Alias é interessante fazer um backup , redirecionando para um arquivo:

```
slapcat >smbldap.ldif
```

Caso ache interessante, poderá instalar a ferramentas **PhpLdapAdmin,** facilitando a manutenção e visualização da base Samba/Ldap. Um simples apt-get resolve esse caso.

Bom à próxima parte deste Post, julgo ser uma das mais importantes, onde a maioria dos Admin, e dos Post que vi Net deixa de comentar. A integração do LDAP com o Linux. Já que para criar um usuário samba, deve existir um usuário Unix/Linux, então devemos integrar o Linux com nossa base LDAP, para que seja satisfeita as necessidades do samba, utilizando o script da ferramenta smbldap-tools para esse gerenciamento de usuários e grupos.

Integrando Linux e LDAP

Quando falamos da integração Linux com LDAP, devemos ter em mente como o Linux autentica.

Por padrão ele usa o PAM (Pluggable Authentication Modules for Linux) para autenticação, que por sua vez faz uso de vários mecanismos para buscar onde irá autenticar.

Um desses mecanismos é o arquivo /etc/nsswitch.conf, que é o nosso Name Service Switch. Outra tarefa também necessária é a configuração do nosso cliente LDAP.

De forma resumida, o Pam vai buscar no NSS como autenticar, que em nosso caso alteraremos para LDAP, e fará uso das bibliotecas pam_ldap e do cliente LDAP. Desta forma temos que configurar o **PAM, NSS e o Ldap cliente.**

O primeiro passo que iremos fazer é reconfigurar o Nss em relação ao LDAP, que está na forma de um pacote chamado **libnss-ldap.** Algumas informações já estarão corretas, enquanto outras deverão ser alteradas.

```
dpkg-reconfigure libnss-ldap
```

Serão solicitadas algumas informações, veja abaixo :

- Identificação do LDAP Server : **ldap://127.0.0.1**
- Distinguished Name da base: **dc=dominiolinux,dc=net**
- Ldap Versão para usar: **3**
- A base LDAP solicitará login: **No**
- Privilégios especiais para o root: **Sim**
- Fazer arquivo de configuração leitura/escrita para o dono: **Yes**
- Conta LDAP para o root: **cn=admin,dc=dominiolinux,dc=net**
- Senha para conta do root: **debian**

Agora devemos atualizar o nosso arquivo /etc/nsswitch.conf, para informar à forma que serão autenticados o usuários. Usaremos a base local Linux (shadow) e a base LDAP. Altere esse arquivo conforme abaixo:

```
passwd: files ldap
group: files ldap
shadow: files ldap
hosts: files mdns4_minimal [NOTFOUND=return] dns mdns4 ldap
```

Alteramos nossos arquivos para buscar usuários, grupos e senha tanto na base local no na base LDAP. Os hosts ficarão um pouco maior, onde a busca por resolução de hosts seguirá a ordem /etc/hosts (files), Multicast DNS (mdns4), caso o anterior não (files) não encontre o nome do host na busca o mdn4 será tratado como a resolução autoritária, posteriormente temos uma busca padrão de dns (dns) , e por fim o LDAP.

Próximo passo, configuração do LDAP cliente, para a máquina local. Devemos alterar o arquivo de configuração /etc/ldap/ldap.conf , conforme abaixo:

```
host localhost
base dc=dominiolinux,dc=net

binddn cn=admin,dc=dominiolinux,dc=net

bindpw debian

bind_policy soft
pam_password exop
timelimit 15

nss_base_passwd dc=dominiolinux,dc=net?sub
nss_base_shadow dc=dominiolinux,dc=net?sub
nss_base_group ou=group,dc=dominiolinux,dc=net?one
```

Altere conforme seu domínio, e grupo caso tenha usado um grupo diferente. Nesse arquivos iremos informar o host, a base local LDAP, o administrador, senha, e a busca por usuários, senha e grupos. Estes valores em sub e one, são relativos ao escopo (Scopo), que em momento apropriado (e futuro) , será abordado na serie Ldap.

E por fim ainda temos o últimos arquivo de configuração para o nss, que é o **/etc/libnns-ldap.conf**. Deverá ficar da seguinte maneira:

```
base dc=dominiolinux,dc=net

uri ldap://127.0.0.1

ldap_version 3

rootbinddn cn=admin,dc=dominiolinux,dc=net

bind_policy soft

pam_password md5

nss_base_passwd  dc=dominiolinux,dc=net?sub

nss_base_shadow  dc=dominiolinux,dc=net?sub

nss_base_group   ou=group,dc=dominiolinux,dc=net?one
```

É bem provável que já esteja quase pronto, afinal de contas usamos o dpkg-reconfigure para configura-lo. Verifique se as ultimas 5 linhas estão batendo com o exemplo acima, pois acredito que a base, uri, versão e rootdn, já estarão prontas.

Para fechar a configuração do nss, veja o conteúdo do arquivo **/etc/libnss-ldap.secrets.** O mesmo deverá conter a senha do LDAP e samba.

```
cat /etc/libnss-ldap.secrets

debian
```

Ufa!!!

Mas ainda não acabou …rsrs. Vamos configurar agora os arquivos do PAM. Ao todo, são 4 arquivos, **common-account, common-auth, common-password e common-session.**

Nesta sessão , peço que prestem muita atenção no que será digitado. Um erro , poderá ocasionar um problema de autenticação, e ninguém mais conseguir logar, tanto no LDAP, como no próprio Shadow local. Alias é interessante, fazer um backup antes, em caso de problemas, poderá volta-lo usando um CD live.

O primeiro passo novamente é usar o reconfigure para configura-los para nós.

```
dpkg-reconfigure libpam-ldap
```

Serão solicitadas algumas informações, veja abaixo :

- Identificação do LDAP Server : **ldap://127.0.0.1**
- Distinguished Name da base: **dc=dominiolinux,dc=net**
- Ldap Versão para usar: **3**
- Fazer o root admin: **Yes**
- A base LDAP solicitará login: **No**
- Conta LDAP para o root: **cn=admin,dc=dominiolinux,dc=net**
- Senha para conta do root: **debian**
    - Criptografia para ser usada: **MD5**

Novamente, a maioria deverá estar já pronta, mas veja com cuidado as que não estão, e principalmente a ultima , em relação à criptografia.

Termos agora que editar os arquivos do pam, para finalizar a configuração.

Todos os arquivos de configuração do pam ficam em **/etc/pam.d.** Vamos ao primeiro.

Arquivo /etc/pam.d/common-account:

```
account [success=2 new_authtok_reqd=done default=ignore]        pam_unix.so

account [success=1 default=ignore]      pam_ldap.so

account requisite                       pam_deny.so

account required                        pam_permit.so
```

Arquivo /etc/pam.d/common-auth:

```
auth    [success=2 default=ignore]      pam_unix.so nullok_secure

auth    [success=1 default=ignore]      pam_ldap.so use_first_pass

auth    requisite                       pam_deny.so

auth    required                        pam_permit.so
```

Arquivo /etc/pam.d/common-password:

```
password        [success=2 default=ignore]      pam_unix.so obscure md5

password        [success=1 user_unknown=ignore default=die]     pam_ldap.so
use_authtok try_first_pass

password        requisite                       pam_deny.so

password        required                        pam_permit.so
```

Arquivo /etc/pam.d/common-session:

```
session [default=1]                     pam_permit.so

session requisite                       pam_deny.so

session required                        pam_permit.so

session required                        pam_unix.so

session optional                        pam_ldap.so
```

Com isso é finalizado a configuração do PAM.

Para não passar em branco totalmente o que foi feito com os arquivos do PAM, passarei alguma informação. Mas de forma sucinta, já que esse assunto também é extenso, e no futuro farei um Post, somente para falar sobre ele (PAM).

O common-account, é responsável pela configuração de autorização para os serviços, em suma, verifica as acessos de permissão de acesso ou não, expiração de conta, restrições e regras de senha.

O common-auth, é utilizado para autenticar usuários e configurar ou cancelar credenciais.

O common-password, executa ações quando da alteração de senha.

E enfim o common-session, é responsável por iniciar e finalizar a sessão.

Todos eles têm seus papéis específicos dentro de toda estrutura de autenticação, e em nosso caso configuramos a biblioteca pam_ldap, para ser usada, para autenticar, controlar sessão, alteração entre outros.

Para finalizarmos e testar devemos reiniciar a distribuição, mas o Nss requer alguns grupos de sistema, que não existem por padrão, por isso antes de reiniciar, vamos cria-los.

```
addgroup –system nvram
addgroup –system rdma
addgroup –system fuse
addgroup –system kvm
adduser –system –group –shell /usr/sbin/nologin –home /var/lib/tpm tss
```

Agora sim!!!! Podemos então reiniciar.

Novamente, sugiro fortemente , para que seja visto as configurações à procura de possíveis erros, antes de reiniciarmos o sistema. Assim que tiver certeza reinicie o servidor.

Testando LDAP

Se tudo funcionou como previsto, você poderá autenticar-se normalmente com os usuários locais sem nenhum tipo de problema. O nosso teste vai consistir em criar um usuário samba-ldap, através da ferramenta smbldap-tools. Verificar se o Linux consegue visualizar a base de usuários tanto local como do Ldap , e por fim autenticar usando esse novo usuário.

Vamos lá!!

```
smbldap-useradd –a –m usuario1

smbldap-passwd usuario1
```

Podemos verificar se ele está já na base usando o comando **getent.** Esse comando ira trazes usuários, grupos em todas as bases que estão acessíveis.

```
#getent passwd

…

root:x:0:0:Netbios Domain Administrator:/home/root:/bin/false

nobody:x:65534:65534:nobody:/nonexistent:/bin/sh

usuario1:x:1001:513:System User:/home/usuario1r:/bin/bash

…
```

Podemos ver claramente que o usuário já consta na base Linux, sendo consultado no próprio servidor LDAP, já que o comando anterior , cria esse usuário na base LDAP.

Prova disso pode ser vista consultando o arquivo /etc/passwd, onde o mesmo não deverá existir.

```
#getent group

…

Domain Admins:*:512:root

Domain Users:*:513:

Domain Guests:*:514:

Domain Computers:*:515:
```

```
Administrators:*:544:

Account Operators:*:548:

Print Operators:*:550:

Backup Operators:*:551:

Replicators:*:552:

…
```

Podemos ver também que o usuário criado, tem como grupo prioritário o GID 513 que corresponde segundo a saída para grupos , ao grupo Domain Users.

Próximo passo , logar no sistema Linux com esse usuário e senha. Tudo deverá funcionar normalmente.

Talvez o Linux reclame por não conseguir mapear o usuário, mas não há problema quanto a isso, o importante é logar.

Enfim, depois de pronto poderá ingressar uma máquina Windows no domínio. Nos meus testes fiz apenas com XP. Irei testar com o Win 7, e tão logo finalize, volto a esse Post para passar informações.

Caso, no momento de ingressar ele reclamar da conta de usuário, é por que o usuário root, não está cadastrado na base LDAP, basta executar os mesmos comandos feitos para o usuario1, só que com o **root.**

A conta de máquina será criada automaticamente, veja abaixo a saída.

```
#getent group

…

usuario1:x:1001:513:System User:/home/usuario1r:/bin/bash

pcstato$:*:1002:515:Computer:/dev/null:/bin/false
```

Após ingressar no domínio, basta logar-se normalmente usando qualquer usuário cadastrado no samba, como o próprio usuario1. Tanto o perfil móvel, quanto o script logon funcionaram corretamente neste ambiente.

Conclusão

Acredito que este foi um dos maiores Posts que já escrevi para o Blog. Mas não tem como omitir determinadas tarefas, pois a omissão ocasionaria no não funcionamento correto deste ambiente. Principalmente no que diz respeitos as contas Unix (PAM/LDAP) e Samba.

Apesar de ser trabalhoso, o ambiente em si como estudo vale a pena, e obviamente como centralizador de contas também é interessante, já que é possível ter outros servidores samba usando a mesma base LDAP.

Em outros Posts veremos como integrar mais serviços ao LDAP.

Com certeza esse Post vai para o novo livro que estou escrevendo LPI 301 – Core (LDAP).lol!!!!

Espero que aproveitem mais um Post do Blog, e até o próximo.