

Manual para implementación y puesto en funcionamiento de un completo servidor de dominio PDC y BDC usando Samba4, Bind9 +dlz, NTP, DHCP.

Funcional tanto en debian 7, como ubuntu 12.04/14.04 LTS



Realizado por: Ing Arian Molina Aguilera
Administrador del Nodo Central ARTex S.A 2015
Email: arian@artex.cu
Jabber: linuxcuba@openmailbox.org
Phone: 7-2047874

Revisado y probado: Ing. Eduardo R. Barrera Pérez
Administrador Nodo CAP Pinar del Río
Email: ebarrera@gobpr.co.cu
Jabber: ebarrera@jabber.gobpr.co.cu
Phone: 0148-728131

<<< SERVIDOR PDC (pdc.tudominio.cu) 192.168.0.2>>>

Editamos el fichero /etc/hosts y lo dejamos de la siguiente manera:

```
# nano /etc/hosts

fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters

127.0.0.1    localhost.localdomain localhost
192.168.0.2  pdc.tudominio.cu pdc
# Auto-generated hostname. Please do not remove this comment.
::1 localhost
```

Editamos el fichero de configuración de la interfaz Ethernet.

```
# nano /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
allow-hotplug eth0
iface eth0 inet static
    address 192.168.0.2
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255
    gateway 192.168.0.1
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 192.168.0.2 192.168.0.3
    dns-search tudominio.cu
```

```
# nano /etc/resolv.conf

search tudominio.cu
nameserver 192.168.0.2
nameserver 192.168.0.3
```

Instalamos Midnight Commander, rcconf y otras herramientas

```
# aptitude install mc rcconf ccze mlocate
```

Instalamos otros paquetes:

```
# aptitude install iptraf nmap unzip unrar-free zip unpin telnet tcpdump
```

Instalamos dependencias para compilar samba4 y bind9.

```
# aptitude install build-essential libacl1-dev libattr1-dev libblkid-dev libgnutls-dev  
libreadline-dev python-dev python-dnspython gdb pkg-config libpopt-dev libldap2-dev  
dnsutils libbsd-dev attr docbook-xsl libcups2-dev acl chkconfig gcc autotools-dev bison  
geoip-bin hardening-wrapper html2text libcap-dev libdb-dev libgeoip-dev libtool  
libxml2-dev m4 krb5-user
```

Una vez instalados los paquetes no debe aparecer una ventana o asistente de configuración de Kerberos pidiéndonos los siguientes datos:

- Reino predeterminado para la versión 5 de kerberos:

en nuestro caso debemos especificar: TUDOMINIO # En Mayúsculas

A continuación nos pedirá:

Servidores Kerberos para su Reino:

Aquí especificamos nombre fqdn de nuestro servidor, en mi caso:

pd.c.tudominio.cu

A continuación nos pedirá:

Servidor administrativo para su reino kerberos: # Aquí también especificamos nombre fqdn de nuestro servidor, ya que el servidor Kerberos estará en el mismo PDC

pd.c.tudominio.cu

Ahora nos disponemos a instalar samba4 para ellos vamos a realizar la instalación desde los fuentes, por lo tanto nos llegamos al sitio oficial de samba [1] y descargamos los fuentes de la última versión estable de samba en este momento es la versión: 4.2.1 lo copiamos para nuestro servidor en /usr/local/src y nos cambiamos a esa ruta.

```
# cd /usr/local/src  
# wget -c https://download.samba.org/pub/samba/stable/samba-4.2.1.tar.gz
```

Descompactamos el paquete:

```
# tar -xvzf samba-4.2.1.tar.gz
```

Antes de instalar debemos asegurarnos de agregar las siguientes líneas al final del archivo `/etc/apparmor.d/local/usr.sbin.named` (crearlo si no existe):

```
# nano /etc/apparmor.d/local/usr.sbin.named
/usr/local/samba/lib/** rm,
/usr/local/samba/private/dns.keytab r,
/usr/local/samba/private/named.conf r,
/usr/local/samba/private/dns/** rwk,
```

Adicionalmente, algunas distribuciones, necesitan de esta otra línea es el caso de Ubuntu 14.04.2 LTS, BIND creará archivos en `/var/tmp/DNS_110`, por tanto se debes agregar esta otra línea:

```
/var/tmp/** rwmk,
```

Ahora debemos asegurarnos de que la partición donde estará instalado nuestro samba4 este montada con los siguientes atributos: `user_xattr,acl,barrier=1` en `/etc/fstab` como nuestro servidor samba4 estará en un servidor virtualizado sobre proxmox usando la variante de virtualización por openvz, ya estos atributos estan presente en la partición del storage local de proxmox, si no corres samba4 virtualizado en openvz (contenedor) y es una PC física o virtualizada con KVM, se debe agregar dichos atributos a la partición donde se instalará samba4.

Ahora nos cambiamos al directorio de samba4

```
# cd samba-4.2.1/
```

Antes de ejecutar el configure, deshabilitamos ipv6 en el sistema, para ello, editamos el fichero `/etc/sysctl.conf` y le añadimos las siguientes líneas:

```
# nano /etc/sysctl.conf

net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
net.ipv6.conf.eth0.disable_ipv6 = 1
```

Para que tengas efecto estos cambios debemos ejecutar el comando:

```
# sysctl -p
```

Oh! Reiniciar el sistema.

Si reiniciamos el sistema nos volvemos a ubicar en el directorio donde descomprimos samba4

```
# cd /usr/local/src/samba-4.2.1/
# ./configure --enable-debug --enable-selftest --disable-cups
```

(Este último si no queremos soporte para cups si solo va hacer el server de PDC)

Una vez que termine, si todo fue bien debe salirnos algo como esto:

'configure' finished successfully.

A continuación ejecutamos estos make y make install para instalar

```
# make
```

Como resultados del make si todo fue bien debemos obtener algo como esto:

'build' finished successfully

Y finalmente el make install

```
# make install
```

Con el que debemos obtener si todo fue bien, algo como esto:

'install' finished successfully

A continuación ejecutamos el siguiente comando para exportar las variables PATH donde se encuentra instalado nuestro samba4 y así poder ejecutar los comandos directamente.

```
# export PATH=$PATH:/usr/local/samba/sbin:/usr/local/samba/bin/
```

Para hacerlo de forma permanente editamos el fichero /etc/profile para añadir la ruta del path donde se instala el samba4

```
# nano /etc/profile

if [ "`id -u`" -eq 0 ]; then
    PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
else
    PATH="/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games"
fi
export PATH
```

Nota: Así deben aparecer las 1ras líneas de este fichero de configuración, después de modificado debe quedar de la siguiente manera:

```
if [ "`id -u`" -eq 0 ]; then

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/samba/sbin/
:/usr/local/samba/bin/"
else
    PATH="/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games"
fi
export PATH
```

Para ubuntu editamos donde lo dejamos de la siguiente manera:

```
#nano /etc/environment
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/samba/sbin:/
usr/local/samba/bin/"
```

Desplegamos samba4

```
# samba-tool domain provision --use-rfc2307 --interactive
```

Realm [tudominio.cu]: TUDOMINIO.CU
#Nombre del Reino Kerberos para Samba en Mayúsculas
Domain [DOMINIO]: DOMINIO
#Nombre que le pondremos al Dominio en Netbios que sería el nombre de nuestro grupo de trabajo
Server Role (dc, member, standalone) [dc]:
#Dejar por defecto el role que desempeñará nuestro servidor DC (Controlador de Dominio)
DNS backend (SAMBA_INTERNAL, BIND9_FLATFILE, BIND9_DLZ, NONE)
[SAMBA_INTERNAL]: BIND9_DLZ
#Tipo de backend DNS en nuestro caso seleccionamos bind9 con DLZ (Dinamic loader Zone)
Administrator password: #
Ponemos la contraseña que para el usuario administrator Que tiene que cumplir con las políticas de seguridad
Retype password:
#Rectificamos la contraseña anterior
#Y obtendremos la salida siguiente si todo fue bien.

Looking up IPv4 addresses
Looking up IPv6 addresses
No IPv6 address will be assigned
Setting up share.ldb
Setting up secrets.ldb
Setting up the registry
Setting up the privileges database
Setting up idmap db
Setting up SAM db
Setting up sam.ldb partitions and settings
Setting up sam.ldb rootDSE
Pre-loading the Samba 4 and AD schema
Adding DomainDN: DC=tudominio,DC=cu
Adding configuration container
Setting up sam.ldb schema
Setting up sam.ldb configuration data
Setting up display specifiers
Modifying display specifiers
Adding users container
Modifying users container
Adding computers container
Modifying computers container
Setting up sam.ldb data
Setting up well known security principals
Setting up sam.ldb users and groups
Setting up self join
Adding DNS accounts
Creating CN=MicrosoftDNS,CN=System,DC=tudominio,DC=cu
Creating DomainDnsZones and ForestDnsZones partitions
Populating DomainDnsZones and ForestDnsZones partitions
Unable to find group id for BIND,
set permissions to sam.ldb* files manually
See /usr/local/samba/private/named.conf for an example configuration include file for BIND
and /usr/local/samba/private/named.txt for further documentation required for secure DNS updates

Setting up sam.ldb rootDSE marking as synchronized

Fixing provision GUIDs

A Kerberos configuration suitable for Samba 4 has been generated at
/usr/local/samba/private/krb5.conf

Setting up fake yp server settings

Once the above files are installed, your Samba4 server will be ready to use

Server Role: active directory domain controller

Hostname: pdc

NetBIOS Domain: TUDOMINIO

DNS Domain: tudominio.cu

DOMAIN SID: S-1-5-21-3426060274-2789830405-1143475183

Ahora instalamos bind9 versión bind-9.10.2. Para configurar bind9 para que todo funcione OK, con nuestro PDC en samba4 hay que compilar el bind9 desde los fuentes ya que la versión del repositorio no viene compilado con soporte para DLZ, por ello nos llegamos al sitio [2] y nos descargamos la última versión de bind9 y la copiamos para nuestro servidor en /usr/local/src

```
# cd /usr/local/src
# wget -c ftp://204.152.184.110/isc/bind9/9.10.2/bind-9.10.2.tar.gz
```

Una vez terminada la descarga, procedemos a descompactar el paquete.

```
# tar -zxvf bind-9.10.2.tar.gz

# cd bind-9.10.2

# ./configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info
--sysconfdir=/etc/bind --localstatedir=/var --enable-threads --enable-largefile --with-libtool
--enable-shared --enable-fixed-rrset --enable-static --with-openssl=/usr --with-gssapi=/usr
--with-gnu-ld --with-dlz-bdb=yes --with-dlz-filesystem=yes --with-dlz-ldap=yes --with-dlz-
stub=yes --with-dlz-dlopen=yes --disable-isc-spnego --with-randomdev=/dev/urandom --with-
geoip=/usr CFLAGS=-fno-strict-aliasing
```

Y finalmente compilamos e instalamos.

```
# make && make install
```

Creamos usuario y grupo para ejecutar bind9.

```
# groupadd -g 23 bind
# useradd -g bind -u 23 -d /var/run/bind -M -s /sbin/nologin bind
```

Si el uid o gid ya está cogido por otro usuario del sistema, buscamos uno bajo que está libre.

Le damos permisos a las llaves de bind9 para kerberos.

```
# chgrp bind /usr/local/samba/private/dns.keytab
# chmod g+r /usr/local/samba/private/dns.keytab
```

Configuramos el bind9.

Creamos el fichero named.conf y le añadimos las siguientes líneas:

```
# nano /etc/bind/named.conf
controls {
    inet 127.0.0.1 port 953
    allow { 127.0.0.1; } keys { "rndc-key"; };
};
options {
    listen-on-v6 { none; };
    auth-nxdomain yes;
    allow-query { any; };
    allow-update { any; };
    dnssec-enable no;
    dnssec-validation no;
    empty-zones-enable no;
    notify yes;
    also-notify { 192.168.0.2; };
    allow-transfer { 192.168.0.3; }; // si va a ser transferencia de zona a otro servidor de dns
esclavo
    notify-source 192.168.0.2;
    transfer-source 192.168.0.2;
    recursion yes;
    allow-recursion { any; };
    forwarders { 200.55.128.3; 200.55.128.4; };
    tkey-gssapi-keytab "/usr/local/samba/private/dns.keytab";
    directory "/var/cache/bind";
};
include "/usr/local/samba/private/named.conf";
include "/etc/bind/rndc.key";
```

Generamos las llaves para el control del bind9

```
# rndc-confgen -a -r /dev/urandom
# chown bind /etc/bind/rndc.key
```

Creamos y le damos permisos al directorio de trabajo de named

```
# mkdir /var/cache/bind
# chown bind:bind /var/cache/bind
# chmod 775 /var/cache/bind
```

Editamos el fichero /usr/local/samba/private/named.conf

```
# nano /usr/local/samba/private/named.conf
y lo dejamos de la siguiente manera:

dlz "AD DNS Zone" {
    # For BIND 9.8.x
    # database "dlopen /usr/local/samba/lib/bind9/dlz_bind9.so";

    # For BIND 9.9.x
    # database "dlopen /usr/local/samba/lib/bind9/dlz_bind9_9.so";

    # For BIND 9.10.x
    database "dlopen /usr/local/samba/lib/bind9/dlz_bind9_10.so";
};
```

Creamos `/usr/local/samba/private/named.conf.update.static` y agregarle lo siguiente:

```
# nano /usr/local/samba/private/named.conf.update.static

grant *.tudominio.cu wildcard *.0.168.192.in-addr.arpa. PTR TXT;
grant local-ddns zonesub any;
```

Creamos enlace para el cliente kerberos use las llaves de samba4

```
# ln -s /usr/local/samba/private/dns.keytab /etc/krb5.keytab
```

Verificamos que el `/etc/resolv.conf` este apuntando al localhost oh! la ip de nuestro servidor samba, ya que anteriormente los teníamos apuntando a otros DNS para poder resolver los nombres para acceder a los repositorios de debian, además que en aquel momento el DNS en este equipo aún no estaba instalado. Como nuestro PDC está en un openvz haremos el cambio de los DNS desde la interfaz de administración de proxmox en lugar de hacerlo directamente en el `resolv.conf` ya que una vez que reiniciemos el openvz volver a tener los mismos valores que están ahora. :)

Arrancamos samba4 y bind9

```
# samba -i -M single &
```

Copyright Andrew Tridgell and the Samba Team 1992-2013

samba: using 'single' process model

Attempting to autogenerate TLS self-signed keys for https for hostname 'PDC.tudominio.cu'

TLS self-signed keys generated OK

Aquí debemos seguramente recibir errores como los siguientes ya que bind9 no está corriendo aún:

```
../source4/dsdb/dns/dns_update.c:294: Failed DNS update - NT_STATUS_IO_TIMEOUT
Matamos el proceso de samba lanzado anteriormente
```

```
# killall samba
```

Ejecutamos este comando varias veces hasta obtener

samba: no process found

Arrancamos el bind9

```
# named -u bind -4 -d 3 &
```

Revisamos los logs si todo funciona bien.

```
# tail -f -n 50 /var/log/syslog | ccze
```

Feb 20 17:03:59 pdc named[9905]: listening on IPv4 interface venet0:0, 192.168.0.57#53

Feb 20 17:03:59 pdc named[9905]: generating session key for dynamic DNS

Feb 20 17:03:59 pdc named[9905]: sizing zone task pool based on 0 zones

Feb 20 17:03:59 pdc named[9905]: Loading 'AD DNS Zone' using driver dlopen

Feb 20 17:03:59 pdc named[9905]: samba_dlz: Failed to connect to

/usr/local/samba/private/dns/sam.ldb

Feb 20 17:03:59 pdc named[9905]: dlz_dlopen of 'AD DNS Zone' failed

Feb 20 17:03:59 pdc named[9905]: SDLZ driver failed to load.

Feb 20 17:03:59 pdc named[9905]: DLZ driver failed to load.

Feb 20 17:03:59 pdc named[9905]: loading configuration: failure

Feb 20 17:03:59 pdc named[9905]: exiting (due to fatal error)

Como podemos ver hay algunos errores que debemos resolver!

Cambiamos el propietario y el grupo al directorio /usr/local/samba/private/dns y a /var/named/run

```
# chown -R bind:root /usr/local/samba/private/dns
# chown -R bind:bind /var/run/bind
# chmod 775 -R /var/run/bind
# chmod 775 -R /usr/local/samba/private/dns
```

Matamos el bind y lo ejecutamos de nuevo.

```
# killall named
```

Nuevamente varias veces en caso de ser necesario hasta que obtengamos: named: no process found

Iniciamos el bind9 nuevamente

```
# named -u bind -4 -d 3 &
```

Miramos el syslog nuevamente

```
# tail -f -n 50 /var/log/syslog | ccze
```

Feb 20 17:11:56 pdc named[10495]: samba_dlz: configured writeable zone '_msdcs.tudominio.cu'

Feb 20 17:11:56 pdc named[10495]: set up managed keys zone for view _default, file 'managed-keys.bind'

Feb 20 17:11:56 pdc named[10495]: zone 'version.bind' allows updates by IP address, which is insecure

Feb 20 17:11:56 pdc named[10495]: zone 'hostname.bind' allows updates by IP address, which is insecure

Feb 20 17:11:56 pdc named[10495]: zone 'authors.bind' allows updates by IP address, which is insecure

Feb 20 17:11:56 pdc named[10495]: zone 'id.server' allows updates by IP address, which is insecure

Feb 20 17:11:56 pdc named[10495]: /etc/named/named.conf:2: couldn't add command channel 127.0.0.1#953: address in use

Feb 20 17:11:56 pdc named[10495]: managed-keys-zone: loaded serial 0

Feb 20 17:11:56 pdc named[10495]: all zones loaded

Feb 20 17:11:56 pdc named[10495]: running

Como podemos ver, ya todo está OK

Ahora debemos crear el siguiente enlace simbólico

```
# mv /etc/krb5.conf /etc/krb5.conf.old
# ln -s /usr/local/samba/private/krb5.conf /etc/krb5.conf
```

Arrancamos Samba nuevamente

```
# samba -i -M single &
```

Si todo fue bien, solo debe mostrarnos!!

samba version 4.2.1 started.

Copyright Andrew Tridgell and the Samba Team 1992-2015
samba: using 'single' process model

Si todo fue bien, creamos la zona inversa en samba4 para que el dns esta completo.

Realizamos las pruebas para rectificar que todo este funcionando bien.

Probando Kerberos

```
# kinit administrator@TUDOMINIO.CU
```

Password for administrator@tudominio.cu:

Warning: Your password will expire in 41 days on Fri Apr 24 17:47:22 2015

```
# klist
```

Ticket cache: FILE:/tmp/krb5cc_0

Default principal: administrator@tudominio.cu

Valid starting	Expires	Service principal
20/02/15 17:39:24	21/02/15 03:39:24	krbtgt/tudominio.cu@tudominio.cu
renew until 21/02/15 17:39:11		

Esta salida nos indica que todo ha ido de maravillas!! :)

Si deseamos que la clave del usuario administrador del dominio no expire, podemos usar este comando:

```
# samba-tool user setexpiry administrator --noexpiry
```

Expiry for user 'administrator' disabled.

Creamos la zonas inversa del DNS y añadimos el registro PTR para nuestro PDC.

```
# samba-tool dns zonecreate tudominio.cu 0.168.192.in-addr.arpa
```

Password for [administrator@tudominio.cu]:

Zone 0.168.192.in-addr.arpa created successfully

```
# samba-tool dns add tudominio.cu 0.168.192.in-addr.arpa 2 PTR pdc.tudominio.cu
```

Password for [administrator@tudominio.cu]:

Record added successfully

Hasta aquí todo bien.

Probando Samba4

```
# smbclient -L localhost -U%
```

Domain=[TUDOMINIO] OS=[Unix] Server=[Samba 4.2.1]

Sharename	Type	Comment
-----	----	-----
netlogon	Disk	
sysvol	Disk	
IPC\$	IPC	IPC Service (Samba 4.2.1)

Domain=[TUDIMINIO] OS=[Unix] Server=[Samba 4.2.1]

Server	Comment
-----	-----

Workgroup	Master
-----	-----

```
# smbclient //localhost/netlogon -UAdministrator%'clave_del_user_administrator' -c 'ls'
```

Domain=[TUDIMINIO] OS=[Unix] Server=[Samba 4.2.1]

.	D	0	Fri Feb 20 16:47:16 2015
..	D	0	Fri Feb 20 16:47:23 2015

40960 blocks of size 1048576. 38636 blocks available

Podemos visualizar los usuarios y grupos:

```
# wbinfo -u
```

Administrator

Guest

krbtgt

dns-pdc

```
# wbinfo -g
```

Enterprise Read-Only Domain Controllers

Domain Admins

Domain Users

Domain Guests

Domain Computers

Domain Controllers

Schema Admins

Enterprise Admins

Group Policy Creator Owners

Read-Only Domain Controllers

DnsUpdateProxy

```
# samba-tool dns query 127.0.0.1 tudominio.cu @ ALL
```

Password for [administrator@tudominio.cu]:

Name=, Records=3, Children=0

SOA: serial=1, refresh=900, retry=600, expire=86400, minttl=3600, ns=pdc.tudominio.cu., email=hostmaster.tudominio.cu. (flags=600000f0, serial=1, ttl=3600)

NS: pdc.tudominio.cu. (flags=600000f0, serial=1, ttl=900)

A: 192.168.0.2 (flags=600000f0, serial=1, ttl=900)

Name=_msdcs, Records=0, Children=0

Name=_sites, Records=0, Children=1

Name=_tcp, Records=0, Children=4

Name=_udp, Records=0, Children=2

Name=DomainDnsZones, Records=0, Children=2

Name=ForestDnsZones, Records=0, Children=2

Name=pdc, Records=1, Children=0

A: 192.168.0.2 (flags=f0, serial=1, ttl=900)

Probando DNS (BIND)

```
# host -t SRV _ldap._tcp.tudominio.cu.
```

_ldap._tcp.tudominio.cu has SRV record 0 100 389 pdc.tudominio.cu.

```
# host -t SRV _kerberos._udp.tudominio.cu.
```

_kerberos._udp.tudominio.cu has SRV record 0 100 88 pdc.tudominio.cu

```
# host -t A pdc.tudominio.cu.
```

pdc.tudominio.cu has address 192.168.0.2

Configuramos el DNS Dinámico Editar el fichero smb.conf y agregar la siguiente línea en la sección [global]

```
# nano /usr/local/samba/etc/smb.conf
```

```
nsupdate command = /usr/bin/nsupdate -g
```

Salvamos y Salimos

Comprobamos que samba actualiza dinámicamente el DNS

```
# samba_dnssupdate --verbose --all-names
```

IPs: ['192.168.0.2']

Calling nsupdate for A pdc.tudominio.cu 192.168.0.2 (add)

Outgoing update query:

```
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0
;; flags: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
;; UPDATE SECTION:
pdc.tudominio.cu. 900 IN A 192.168.0.2
```

Calling nsupdate for A tudominio.cu 192.168.0.2 (add)

Outgoing update query:

```
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0
;; flags: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
;; UPDATE SECTION:
tudominio.cu. 900 IN A 192.168.0.2
```

Calling nsupdate for SRV _ldap._tcp.tudominio.cu pdc.tudominio.cu 389 (add)

Outgoing update query:

```
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0
;; flags: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
;; UPDATE SECTION:
_ldap._tcp.tudominio.cu. 900 IN SRV 0 100 389 pdc.tudominio.cu.
```

Calling nsupdate for SRV _ldap._tcp.dc._msdcs.tudominio.cu pdc.tudominio.cu 389 (add)

Outgoing update query:

```
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0
;; flags: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
;; UPDATE SECTION:
```

_ldap._tcp.dc._msdcs.tudominio.cu. 900 IN SRV 0 100 389 pdc.tudominio.cu.

Calling nsupdate for SRV _ldap._tcp.93e36c40-4a96-4bb6-ba2f-7e41fad6f7d5.domains._msdcs.tudominio.cu pdc.tudominio.cu 389 (add)

Outgoing update query:

;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0

;; flags;; ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0

;; UPDATE SECTION:

_ldap._tcp.93e36c40-4a96-4bb6-ba2f-7e41fad6f7d5.domains._msdcs.tudominio.cu. 900 IN SRV 0 100 389 pdc.tudominio.cu.

Calling nsupdate for SRV _kerberos._tcp.tudominio.cu pdc.tudominio.cu 88 (add)

Outgoing update query:

;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0

;; flags;; ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0

;; UPDATE SECTION:

_kerberos._tcp.tudominio.cu. 900 IN SRV 0 100 88 pdc.tudominio.cu.

Calling nsupdate for SRV _kerberos._udp.tudominio.cu pdc.tudominio.cu 88 (add)

Outgoing update query:

;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0

;; flags;; ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0

;; UPDATE SECTION:

_kerberos._udp.tudominio.cu. 900 IN SRV 0 100 88 pdc.tudominio.cu.

Calling nsupdate for SRV _kerberos._tcp.dc._msdcs.tudominio.cu pdc.tudominio.cu 88 (add)

Outgoing update query:

;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0

;; flags;; ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0

;; UPDATE SECTION:

_kerberos._tcp.dc._msdcs.tudominio.cu. 900 IN SRV 0 100 88 pdc.tudominio.cu.

Calling nsupdate for SRV _kpasswd._tcp.tudominio.cu pdc.tudominio.cu 464 (add)

Outgoing update query:

;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0

;; flags;; ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0

;; UPDATE SECTION:

_kpasswd._tcp.tudominio.cu. 900 IN SRV 0 100 464 pdc.tudominio.cu.

Calling nsupdate for SRV _kpasswd._udp.tudominio.cu pdc.tudominio.cu 464 (add)

Outgoing update query:

;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0

;; flags;; ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0

;; UPDATE SECTION:

_kpasswd._udp.tudominio.cu. 900 IN SRV 0 100 464 pdc.tudominio.cu.

Calling nsupdate for CNAME c113ebd7-24a4-4cf3-8f6e-03050e3fe4ce._msdcs.tudominio.cu

pdc.tudominio.cu (add)

Outgoing update query:

;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0

;; flags;; ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0

:: UPDATE SECTION:

c113ebd7-24a4-4cf3-8f6e-03050e3fe4ce._msdcs.tudominio.cu. 900 IN CNAME pdc.tudominio.cu.

Calling nsupdate for SRV _ldap._tcp.Default-First-Site-Name._sites.tudominio.cu pdc.tudominio.cu 389 (add)

Outgoing update query:

:: ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0

:: flags:: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0

:: UPDATE SECTION:

_ldap._tcp.Default-First-Site-Name._sites.tudominio.cu. 900 IN SRV 0 100 389

pdc.tudominio.cu.

Calling nsupdate for SRV _ldap._tcp.Default-First-Site-Name._sites.dc._msdcs.tudominio.cu pdc.tudominio.cu 389 (add)

Outgoing update query:

:: ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0

:: flags:: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0

:: UPDATE SECTION:

_ldap._tcp.Default-First-Site-Name._sites.dc._msdcs.tudominio.cu. 900 IN SRV 0 100 389

pdc.tudominio.cu.

Calling nsupdate for SRV _kerberos._tcp.Default-First-Site-Name._sites.tudominio.cu pdc.tudominio.cu 88 (add)

Outgoing update query:

:: ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0

:: flags:: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0

:: UPDATE SECTION:

_kerberos._tcp.Default-First-Site-Name._sites.tudominio.cu. 900 IN SRV 0 100 88

pdc.tudominio.cu.

Calling nsupdate for SRV _kerberos._tcp.Default-First-Site-Name._sites.dc._msdcs.tudominio.cu pdc.tudominio.cu 88 (add)

Outgoing update query:

:: ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0

:: flags:: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0

:: UPDATE SECTION:

_kerberos._tcp.Default-First-Site-Name._sites.dc._msdcs.tudominio.cu. 900 IN SRV 0 100 88

pdc.tudominio.cu.

Calling nsupdate for SRV _ldap._tcp.pdc._msdcs.tudominio.cu pdc.tudominio.cu 389 (add)

Outgoing update query:

:: ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0

:: flags:: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0

:: UPDATE SECTION:

_ldap._tcp.pdc._msdcs.tudominio.cu. 900 IN SRV 0 100 389 pdc.tudominio.cu.

Calling nsupdate for A gc._msdcs.tudominio.cu 192.168.0.2 (add)

Outgoing update query:

:: ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0

:: flags:: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0

```
:: UPDATE SECTION:
gc._msdcs.tudominio.cu. 900 IN      A      192.168.0.2
```

Calling nsupdate for SRV _gc._tcp.tudominio.cu pdc.tudominio.cu 3268 (add)

Outgoing update query:

```
:: ->>HEADER<<- opcode: UPDATE, status: NOERROR, id:    0
:: flags:: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
:: UPDATE SECTION:
_gc._tcp.tudominio.cu. 900  IN      SRV   0 100 3268 pdc.tudominio.cu.
```

Calling nsupdate for SRV _ldap._tcp.gc._msdcs.tudominio.cu pdc.tudominio.cu 3268 (add)

Outgoing update query:

```
:: ->>HEADER<<- opcode: UPDATE, status: NOERROR, id:    0
:: flags:: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
:: UPDATE SECTION:
_ldap._tcp.gc._msdcs.tudominio.cu. 900 IN SRV 0 100 3268 pdc.tudominio.cu.
```

Calling nsupdate for SRV _gc._tcp.Default-First-Site-Name._sites.tudominio.cu pdc.tudominio.cu 3268 (add)

Outgoing update query:

```
:: ->>HEADER<<- opcode: UPDATE, status: NOERROR, id:    0
:: flags:: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
:: UPDATE SECTION:
_gc._tcp.Default-First-Site-Name._sites.tudominio.cu. 900 IN SRV 0 100 3268 pdc.tudominio.cu.
```

Calling nsupdate for SRV _ldap._tcp.Default-First-Site-Name._sites.gc._msdcs.tudominio.cu pdc.tudominio.cu 3268 (add)

Outgoing update query:

```
:: ->>HEADER<<- opcode: UPDATE, status: NOERROR, id:    0
:: flags:: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
:: UPDATE SECTION:
_ldap._tcp.Default-First-Site-Name._sites.gc._msdcs.tudominio.cu. 900 IN SRV 0 100 3268
pdc.tudominio.cu.
```

Calling nsupdate for A DomainDnsZones.tudominio.cu 192.168.0.2 (add)

Outgoing update query:

```
:: ->>HEADER<<- opcode: UPDATE, status: NOERROR, id:    0
:: flags:: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
:: UPDATE SECTION:
DomainDnsZones.tudominio.cu. 900 IN A   192.168.0.2
```

Calling nsupdate for SRV _ldap._tcp.DomainDnsZones.tudominio.cu pdc.tudominio.cu 389 (add)

Outgoing update query:

```
:: ->>HEADER<<- opcode: UPDATE, status: NOERROR, id:    0
:: flags:: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
:: UPDATE SECTION:
_ldap._tcp.DomainDnsZones.tudominio.cu. 900 IN SRV   0 100 389 pdc.tudominio.cu.
```

Calling nsupdate for SRV _ldap._tcp.Default-First-Site-Name._sites.DomainDnsZones.tudominio.cu pdc.tudominio.cu 389 (add)

Outgoing update query:

```
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0
;; flags: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
;; UPDATE SECTION:
_ldap._tcp.Default-First-Site-Name._sites.DomainDnsZones.tudominio.cu. 900 IN SRV 0 100 389
pdc.tudominio.cu.
```

Calling nsupdate for A ForestDnsZones.tudominio.cu 192.168.0.2 (add)

Outgoing update query:

```
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0
;; flags: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
;; UPDATE SECTION:
ForestDnsZones.tudominio.cu. 900 IN A 192.168.0.2
```

Calling nsupdate for SRV _ldap._tcp.ForestDnsZones.tudominio.cu pdc.tudominio.cu 389 (add)

Outgoing update query:

```
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0
;; flags: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
;; UPDATE SECTION:
_ldap._tcp.ForestDnsZones.tudominio.cu. 900 IN SRV 0 100 389 pdc.tudominio.cu.
```

Calling nsupdate for SRV _ldap._tcp.Default-First-Site-Name._sites.ForestDnsZones.tudominio.cu pdc.tudominio.cu 389 (add)

Outgoing update query:

```
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id: 0
;; flags: ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
;; UPDATE SECTION:
_ldap._tcp.Default-First-Site-Name._sites.ForestDnsZones.tudominio.cu. 900 IN SRV 0 100 389
pdc.tudominio.cu.
```

En esta salida vemos que todo está OK.

Si todas las pruebas fueron satisfactorias, procedemos a instalar y configura NTP para dotar de todas las características a nuestro PDC.

Instalamos ntp

```
# aptitude install ntp
```

Configuramos la Zona Horaria

```
# dpkg-reconfigure tzdata
```

Y escogemos: America/Havana

Editamos el fichero de configuración de NTP /etc/ntp.conf

```
# nano /etc/ntp.conf
```

Añadimos las siguientes líneas:

Debajo de la línea:

```
driftfile /var/lib/ntp/ntp.drift
```

Añadimos esta:

```
logfile /var/log/ntpd.log
```

agregamos el contenido siguiente.

```
server 127.127.1.1
```

```
fudge 127.127.1.1 stratum 12
```

Comentamos las siguientes líneas:

```
server 0.debian.pool.ntp.org iburst
```

```
server 1.debian.pool.ntp.org iburst
```

```
server 2.debian.pool.ntp.org iburst
```

```
server 3.debian.pool.ntp.org iburst
```

quedando

```
# server 0.debian.pool.ntp.org iburst
```

```
# server 1.debian.pool.ntp.org iburst
```

```
# server 2.debian.pool.ntp.org iburst
```

```
# server 3.debian.pool.ntp.org iburst
```

Y añadimos las líneas:

```
server ntp.tudominio.cu iburst prefer #servidor de tiempo al que tengas acceso para que sincronice
```

```
ntpsigndsocket /usr/local/samba/var/lib/ntp_signd/
```

Comentamos y editamos las líneas:

```
restrict -4 default kod notrap nomodify nopeer noquery
```

```
restrict -6 default kod notrap nomodify nopeer noquery
```

Quedando de la siguiente manera:

```
restrict -4 default kod notrap nomodify nopeer noquery mssntp
```

```
#restrict -6 default kod notrap nomodify nopeer noquery
```

Y debajo ponemos estas líneas:

```
restrict default mssntp
```

```
restrict tiempo.tudominio.cu mask 255.255.255.255 nomodify notrap nopeer noquery
```

Salvamos y Salimos

Reiniciamos NTP

```
# invoke-rc.d ntp restart o service ntp restart en ubuntu
```

[ok] Stopping NTP server: ntpd.

[ok] Starting NTP server: ntpd.

Antes de comprobar si nuestro sistema corre es un contenedor openvz de proxmox, debemos agregar la funcionalidad system time a la máquina virtual, de lo contrario el servicio ntp no arrancará y nos dará un error. Si no es el caso todo sigue normal.

Comprobamos que estamos sincronizados

```
# ntpq -p
```

```
remote      refid      st t when poll reach  delay  offset jitter
=====
=====
*LOCAL(1)    .LOCL.      12 l 6 64  1  0.000  0.000  0.000
192.168.0.250 LOCAL(0)    5 u  1 64  1  0.262 1668.01 264.399
```

Le damos permiso al directorio ntp_signd de samba4

```
# chgrp ntp /usr/local/samba/var/lib/ntp_signd/
# chmod 750 /usr/local/samba/var/lib/ntp_signd/
```

Reiniciamos ntp nuevamente

```
# invoke-rc.d ntp restart o service ntp restart en ubuntu
```

Creamos los script para arrancar samba4 y bind9

```
# nano /etc/init.d/samba4
```

```
#!/bin/sh
```

```
#####
# Este Script ha sido realizado por el                               #
# Grupo de Software Libre de ARTex.SA                               #
#####
```

```
# Defaults
```

```
RUN_MODE="daemons"
```

```
[ -r /etc/default/samba ] && . /etc/default/samba
```

```
PIDDIR=/usr/local/samba/var/run
```

```
NMBDPID=$PIDDIR/nmbd.pid
```

```
SMBDPID=$PIDDIR/smbd.pid
```

```
unset TMPDIR
```

```
test -x /usr/local/samba/sbin/nmbd -a -x /usr/local/samba/sbin/samba || exit 0
```

```
. /lib/lsb/init-functions
```

```
case "$1" in
```

```
start)
```

```
    log_daemon_msg "Iniciando el Controlador de Dominio Samba"
```

```
    install -o root -g root -m 755 -d $PIDDIR
```

```

NMBD_DISABLED=`testparm -s --parameter-name='disable netbios' 2>/dev/null`
if [ "$NMBD_DISABLED" != 'Yes' ]; then
    log_progress_msg "nmbd"
    if ! start-stop-daemon --start --quiet --oknodo --exec /usr/local/samba/sbin/samba --
-D
        then
            log_end_msg 1
            exit 1
        fi
    fi

if [ "$RUN_MODE" != "inetd" ]; then
    log_progress_msg "smbd"
    if ! start-stop-daemon --start --quiet --oknodo --exec /usr/local/samba/sbin/samba --
-D; then
        log_end_msg 1
        exit 1
    fi
fi

log_end_msg 0
;;
stop)
    log_daemon_msg "Detenido el Controlador de Dominio Samba"
    log_progress_msg "nmbd"

    start-stop-daemon --stop --quiet --pidfile $NMBDPID
    sleep 1
    if [ -f $NMBDPID ] && ! ps h `cat $NMBDPID` > /dev/null
    then
        rm -f $NMBDPID
    fi

    if [ "$RUN_MODE" != "inetd" ]; then
        log_progress_msg "smbd"
        start-stop-daemon --stop --quiet --pidfile $SMBDPID
        sleep 1
        if [ -f $SMBDPID ] && ! ps h `cat $SMBDPID` > /dev/null
        then
            rm -f $SMBDPID
        fi
    fi

    log_end_msg 0

    ;;

reload)
    log_daemon_msg "Recargando configuraciones sencillas del Controlador de Dominio
Samba4"

```

```

start-stop-daemon --stop --signal HUP --pidfile $SMBDPID

log_end_msg 0
;;
restart|force-reload)
    $0 stop
    sleep 1
    $0 start
    ;;
status)
    status="0"
    NMBD_DISABLED=`testparm -s --parameter-name='disable netbios' 2>/dev/null`
    if [ "$RUN_MODE" != "inetd" ]; then
        status_of_proc -p $SMBDPID /usr/local/samba/sbin/samba Samba4 || status=$?
    fi
    if [ "$NMBD_DISABLED" = "Yes" -a "$RUN_MODE" = "inetd" ]; then
        status="4"
    fi
    exit $status
    ;;
*)
    echo "Usage: /etc/init.d/samba {start|stop|reload|restart|force-reload|status}"
    exit 1
    ;;
esac

exit 0

```

Salvamos y Salimos.

Le damos permiso de ejecución al scripts

```
# chmod +x /etc/init.d/samba4
```

Creamos el Scripts para Bind9

```
# nano /etc/init.d/bind9
```

```

#!/bin/sh -e
#####
# Este Script ha sido adaptado por el                                     #
# Grupo de Software Libre de ARTex.SA                                     #
#####
### BEGIN INIT INFO
# Provides:          bind9
# Required-Start:    $remote_fs
# Required-Stop:     $remote_fs
# Should-Start:      $network $syslog
# Should-Stop:       $network $syslog

```

```

# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Start and stop bind9
# Description: bind9 is a Domain Name Server (DNS)
# which translates ip addresses to and from internet names
### END INIT INFO

PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/bin:/usr/local/sbin

# for a chrooted server: "-u bind -t /var/lib/named"
# Don't modify this line, change or create /etc/default/bind9.
OPTIONS="-4 -u bind"
RESOLVCONF=no

#test -f /etc/default/bind9 && . /etc/default/bind9

test -x /usr/sbin/rndc || exit 0

. /lib/lsb/init-functions
PIDFILE=/var/run/bind/named.pid

check_network() {
    if [ -x /usr/bin/uname ] && [ "X$(/usr/bin/uname -o)" = XSolaris ]; then
        IFCONFIG_OPTS="-au"
    else
        IFCONFIG_OPTS=""
    fi
    if [ -z "$( /sbin/ifconfig $IFCONFIG_OPTS )" ]; then
        log_action_msg "No networks configured."
        return 1
    fi
    return 0
}

case "$1" in
    start)
        log_daemon_msg "Starting domain name service..." "bind9"

        modprobe capability >/dev/null 2>&1 || true

        # dirs under /var/run can go away on reboots.
        mkdir -p /var/run/bind
        chmod 775 /var/run/bind
        chown root:bind /var/run/bind >/dev/null 2>&1 || true

        if [ ! -x /usr/sbin/named ]; then
            log_action_msg "named binary missing - not starting"
            log_end_msg 1
        fi

```

```

if ! check_network; then
    log_action_msg "no networks configured"
    log_end_msg 1
fi

if start-stop-daemon --start --oknodo --quiet --exec /usr/sbin/named \
    --pidfile ${PIDFILE} -- $OPTIONS; then
    if [ "X$RESOLVCONF" != "Xno" ] && [ -x /sbin/resolvconf ] ; then
        echo "nameserver 127.0.0.1" | /sbin/resolvconf -a lo.named
    fi
    log_end_msg 0
else
    log_end_msg 1
fi
;;

stop)
log_daemon_msg "Stopping domain name service..." "bind9"
if ! check_network; then
    log_action_msg "no networks configured"
    log_end_msg 1
fi

if [ "X$RESOLVCONF" != "Xno" ] && [ -x /sbin/resolvconf ] ; then
    /sbin/resolvconf -d lo.named
fi
pid=$(/usr/sbin/rndc stop -p | awk '/^pid:/ {print $2}') || true
if [ -z "$pid" ]; then      # no pid found, so either not running, or error
    pid=$(pgrep -f ^/usr//sbin/named) || true
    start-stop-daemon --stop --oknodo --quiet --exec /usr/sbin/named \
        --pidfile ${PIDFILE} -- $OPTIONS
fi
if [ -n $pid ]; then
    sig=0
    n=1
    while kill -$sig $pid 2>/dev/null; do
        if [ $n -eq 1 ]; then
            echo "waiting for pid $pid to die"
        fi
        if [ $n -eq 11 ]; then
            echo "giving up on pid $pid with kill -0; trying -9"
            sig=9
        fi
        if [ $n -gt 20 ]; then
            echo "giving up on pid $pid"
            break
        fi
        n=$((n+1))
        sleep 1
    done

```

```

fi
log_end_msg 0
;;

reload|force-reload)
log_daemon_msg "Reloading domain name service..." "bind9"
if ! check_network; then
    log_action_msg "no networks configured"
    log_end_msg 1
fi

/usr/sbin/rndc reload >/dev/null && log_end_msg 0 || log_end_msg 1
;;

restart)
if ! check_network; then
    log_action_msg "no networks configured"
    exit 1
fi

$0 stop
$0 start
;;

status)
    ret=0
    status_of_proc -p ${PIDFILE} /usr/sbin/named bind9 2>/dev/null || ret=$?
    exit $ret
;;

*)
log_action_msg "Usage: /etc/init.d/bind9 {start|stop|reload|restart|force-reload|status}"
exit 1
;;
esac

exit 0

```

Salvamos y Salimos

Le damos permisos de ejecución

```
# chmod +x /etc/init.d/bind9
```

Agregamos los script para arranquen con el sistema.

```
# update-rc.d samba4 defaults && service samba4 start
# update-rc.d bind9 defaults && service bind9 start
```

A partir de aquí, ya podemos reiniciar el sistema y comprobar que todo inicie correctamente, posteriormente procedemos a agregar las pc al dominios, y desde una maquina en el dominio iniciar sesión con el usuario administrator, e instalar RSAT (Remote Server Administrator Tools) para gestionar y administrar nuestro controlador de dominio y dns, exactamente como se hace con un AD

de microsoft, creamos los usuarios y grupos necesario, agregamos nuestro usuario al grupo Domain Admins, para poder administrar desde nuestra pc y usuario del dominio nuestro PDC SAMBA4.

[1] <https://www.samba.org/>

[2] <https://www.isc.org/downloads/>

DHCP

Deberíamos conocer que usando esta alternativa de integración del servicio de dhcp el dns dinámico podría necesitar cambios desde los clientes windows, los cuales seguirán intentando actualizar dinámicamente el dns, por lo cual el servidor seguirá denegando dichas actualizaciones por el dueño del usuario del dhcp.

Necesitaremos crear una GPO para impedir esto, pero desafortunadamente, Samba 4 no cuenta con una utilidad de línea de comando para modificar GPOs, se necesita un PC con Windows y la herramienta RSAT instalada. Simplemente creamos una GPO dedicada con el Editor de políticas de grupo, la aplicamos solamente a la OUs que contiene a las estaciones de trabajo (Entonces estos servidores seguirán actualizándose de la forma clásica usando 'ipconfig /registerdns') y configurar siguiendo los siguientes pasos:

Computer Configuration

Policies

Administrative Templates

Network

DNS Client

Dynamic Update = Disabled

Register PTR Records = Disabled

Instalamos ISC DHCPd en el PDC:

```
# aptitude install isc-dhcp-server
```

Creamos un usuarios sin privilegios en el AD para poder realizar las actualizaciones dinámicas del dns vía dhcp script. Cuando pregunte la contraseña, usamos una contraseña segura. 63 random, mixed case, caracteres alpha-numeric es suficiente. Opcionalmente samba-tool también toma argumentos al azar (random):

```
#samba-tool user create dhcp --description="Usuario sin privilegios para actualizar DNS via DHCP server"
```

Agregamos dicho usuarios al grupo DnsAdmins administrador de DNS:

```
#samba-tool group addmembers DnsAdmins dhcp
```

Exportamos las credenciales del usuario al keytab privado:

```
#samba-tool domain exportkeytab --principal=dhcp@TUDOMINIO.CU dhcpd.keytab
```

Aseguramos los permisos para el archivo de llaves que solo pueda ser leído por root, si dhcpd corre con otro usuario cambiamos root por el mismo.

```
#install -vdm 755 /etc/dhcpd
#mv dhcpd.keytab /etc/dhcpd
#chown root:root /etc/dhcpd/dhcpd.keytab
#chmod 400 /etc/dhcpd/dhcpd.keytab
```

Ahora que el usuario esta creado, agregamos el siguiente script para realizar las actualizaciones dinámicas del DNS:

```
#cat > /usr/sbin/samba-dnsupdate.sh << "EOF"
#!/bin/bash
# Begin samba-dnsupdate.sh
# Author: DJ Lucas <dj_AT_linuxfromscratch_DOT_org>
# kerberos_creds() courtesy of Sergey Urushkin
# http://www.kuron-germany.de/michael/blog/wp-content/uploads/2012/03/dhcpdns-sergey2.txt

# DHCP server should be authoritative for its own records, sleep for 5 seconds
# to allow unconfigured Windows hosts to create their own DNS records
# In order to use this script you should disable dynamic updates by hosts that
# will receive addresses from this DHCP server. Instructions are found here:
# https://wiki.archlinux.org/index.php/Samba_4_Active_Directory_Domain_Controller#DHCP
sleep 5

checkvalues()
{
    [ -z "${2}" ] && echo "Error: argument '${1}' requires a parameter." && exit 1

    case ${2} in
        -*)
            echo "Error: Invalid parameter '${2}' passed to ${1}."
            exit 1
            ;;
        *)
            return 0
            ;;
    esac
}

showhelp()
{
    echo -e "\n" `basename ${0}` "uses samba-tool to update DNS records in Samba 4's DNS"
    echo "server when using INTERNAL DNS or BIND9 DLZ plugin."
    echo ""
    echo "  Command line options (and variables):"
    echo ""
    echo "  -a | --action    Action for this script to perform"
    echo "                   ACTION={add|delete}"
    echo "  -c | --krb5cc    Path of the krb5 credential cache (optional)"
    echo "                   Default: KRB5CC=/run/dhcpd.krb5cc"
    echo "  -d | --domain    The DNS domain/zone to be updated"
    echo "                   DOMAIN={domain.tld}"
    echo "  -h | --help      Show this help message and exit"
    echo "  -H | --hostname  Hostname of the record to be updated"
    echo "                   HNAME={hostname}"
}
```

```

echo "    -i | --ip      IP address of the host to be updated"
echo "                  IP={0.0.0.0}"
echo "    -k | --keytab   Krb5 keytab to be used for authorization (optional)"
echo "                  Default: KEYTAB=/etc/dhcp/dhcpd.keytab"
echo "    -m | --mitkrb5   Use MIT krb5 client utilities"
echo "                  MITKRB5={YES|NO}"
echo "    -n | --nameserver DNS server to be updated (must use FQDN, not IP)"
echo "                  NAMESERVER={server.internal.domain.tld}"
echo "    -p | --principal Principal used for DNS updates"
echo "                  PRINCIPAL={user@domain.tld}"
echo "    -r | --realm     Authentication realm"
echo "                  REALM={DOMAIN.TLD}"
echo "    -z | --zone      Then name of the zone to be updated in AD."
echo "                  ZONE={zonename}"
echo ""
echo "Example: $(basename $0) -d domain.tld -i 192.168.0.x -n 192.168.0.x \"\"
echo "          -r DOMAIN.TLD -p user@domain.tld -H HOSTNAME -m"
echo ""
}

```

Process arguments

```
[ -z "$1" ] && showhelp && exit 1
```

```
while [ -n "$1" ]; do
```

```
    case $1 in
```

```
        -a | --action)
```

```
            checkvalues ${1} ${2}
```

```
            ACTION=${2}
```

```
            shift 2
```

```
        ;;
```

```
        -c | --krb5cc)
```

```
            checkvalues ${1} ${2}
```

```
            KRB5CC=${2}
```

```
            shift 2
```

```
        ;;
```

```
        -d | --domain)
```

```
            checkvalues ${1} ${2}
```

```
            DOMAIN=${2}
```

```
            shift 2
```

```
        ;;
```

```
        -h | --help)
```

```
            showhelp
```

```
            exit 0
```

```
        ;;
```

```
        -H | --hostname)
```

```
            checkvalues ${1} ${2}
```

```

        HNAME=${2%%.*}
        shift 2
;;

-i | --ip)
    checkvalues ${1} ${2}
    IP=${2}
    shift 2
;;

-k | --keytab)
    checkvalues ${1} ${2}
    KEYTAB=${2}
    shift 2
;;

-m | --mitkrb5)
    KRB5MIT=YES
    shift 1
;;

-n | --nameserver)
    checkvalues ${1} ${2}
    NAMESERVER=${2}
    shift 2
;;

-p | --principal)
    checkvalues ${1} ${2}
    PRINCIPAL=${2}
    shift 2
;;

-r | --realm)
    checkvalues ${1} ${2}
    REALM=${2}
    shift 2
;;

-z | --zone)
    checkvalues ${1} ${2}
    ZONE=${2}
    shift 2
;;

*)
    echo "Error!!! Unknown command line option!"
    echo "Try" `basename $0` "--help."
    exit 1
;;

```

```

        esac
done

# Sanity checking
[ -z "$ACTION" ] && echo "Error: action not set." && exit 2
case "$ACTION" in
    add | Add | ADD)
        ACTION=ADD
        ;;
    del | delete | Delete | DEL | DELETE)
        ACTION=DEL
        ;;
    *)
        echo "Error: invalid action \"$ACTION\"." && exit 3
        ;;
esac
[ -z "$KRB5CC" ] && KRB5CC=/run/dhcpd.krb5cc
[ -z "$DOMAIN" ] && echo "Error: invalid domain." && exit 4
[ -z "$HNAME" ] && [ "$ACTION" == "ADD" ] && \
    echo "Error: hostname not set." && exit 5
[ -z "$IP" ] && echo "Error: IP address not set." && exit 6
[ -z "$KEYTAB" ] && KEYTAB=/etc/dhcp/dhcpd.keytab
[ -z "$NAMESERVER" ] && echo "Error: nameservers not set." && exit 7
[ -z "$PRINCIPAL" ] && echo "Error: principal not set." && exit 8
[ -z "$REALM" ] && echo "Error: realm not set." && exit 9
[ -z "$ZONE" ] && echo "Error: zone not set." && exit 10

# Disassemble IP for reverse lookups
OCT1=$(echo $IP | cut -d . -f 1)
OCT2=$(echo $IP | cut -d . -f 2)
OCT3=$(echo $IP | cut -d . -f 3)
OCT4=$(echo $IP | cut -d . -f 4)
RZONE="$OCT3.$OCT2.$OCT1.in-addr.arpa"

kerberos_creds() {
export KRB5_KTNAME="$KEYTAB"
export KRB5CCNAME="$KRB5CC"

if [ "$KRB5MIT" = "YES" ]; then
    KLISTARG="-s"
else
    KLISTARG="-t"
fi

klist $KLISTARG || kinit -k -t "$KEYTAB" -c "$KRB5CC" "$PRINCIPAL" || { logger -s -p
daemon.error -t dhcpd kinit for dynamic DNS failed; exit 11; }
}

add_host(){

```

```

    logger -s -p daemon.info -t dhcpd Adding A record for host $HNAME with IP $IP to zone $ZONE
on server $NAMESERVER
    samba-tool dns add $NAMESERVER $ZONE $HNAME A $IP -k yes
}

```

```

delete_host(){
    logger -s -p daemon.info -t dhcpd Removing A record for host $HNAME with IP $IP from zone
$ZONE on server $NAMESERVER
    samba-tool dns delete $NAMESERVER $ZONE $HNAME A $IP -k yes
}

```

```

update_host(){
    CURIP=$(host -t A $HNAME | cut -d " " -f 4)
    logger -s -p daemon.info -t dhcpd Removing A record for host $HNAME with IP $CURIP from
zone $ZONE on server $NAMESERVER
    samba-tool dns delete $NAMESERVER $ZONE $HNAME A $CURIP -k yes
    add_host
}

```

```

add_ptr(){
    logger -s -p daemon.info -t dhcpd Adding PTR record $OCT4 with hostname $HNAME to zone
$RZONE on server $NAMESERVER
    samba-tool dns add $NAMESERVER $RZONE $OCT4 PTR $HNAME.$DOMAIN -k yes
}

```

```

delete_ptr(){
    logger -s -p daemon.info -t dhcpd Removing PTR record $OCT4 with hostname $HNAME from
zone $RZONE on server $NAMESERVER
    samba-tool dns delete $NAMESERVER $RZONE $OCT4 PTR $HNAME.$DOMAIN -k yes
}

```

```

update_ptr(){
    CURHNAME=$(host -t PTR $OCT4 | cut -d " " -f 5)
    logger -s -p daemon.info -t dhcpd Removing PTR record $OCT4 with hostname $CURHNAME
from zone $RZONE on server $NAMESERVER
    samba-tool dns delete $NAMESERVER $RZONE $OCT4 PTR $CURHNAME -k yes
    add_ptr
}

```

```

case "$ACTION" in
    ADD)
        kerberos_creds
        host -t A $HNAME.$DOMAIN > /dev/null
        if [ "${?}" == 0 ]; then
            update_host

```

```

else
    add_host
fi

host -t PTR $IP > /dev/null
if [ "${?}" == 0 ]; then
    update_ptr
else
    add_ptr
fi
;;

DEL)
    kerberos_creds
    host -t A $HNAME.$DOMAIN > /dev/null
    if [ "${?}" == 0 ]; then
        delete_host
    fi

    host -t PTR $IP > /dev/null
    if [ "${?}" == 0 ]; then
        delete_ptr
    fi
;;

*)
    echo "Error: Invalid action '$ACTION'!" && exit 12
;;

```

esac

End samba-dnsupdate.sh
EOF

```
#chmod 750 /usr/sbin/samba-dnsupdate.sh
```

El servicios de dhcp necesitará un script rápido para la escritura evitando los bloqueos, demoras. Crearemos el script con el siguiente comando (sustituyendo los valores correctos para nuestra red, dominio y servidor, tudominio.cu, y TUDOMINIO.CU):

```

#cat > /etc/dhcpd/update.sh << "EOF"
#!/bin/bash
# Begin /etc/dhcpd/update.sh
# Variables
KRB5CC="/run/dhcpd4.krb5cc"
KEYTAB="/etc/dhcpd/dhcpd.keytab"
DOMAIN="tudominio.cu"
REALM="TUDOMINIO.CU"
PRINCIPAL="dhcp@${REALM}"
NAMESERVER="pdc.${DOMAIN}"
ZONE="${DOMAIN}"

```

```
ACTION=$1
IP=$2
HNAME=$3
```

```
export KRB5CC KEYTAB DOMAIN REALM PRINCIPAL NAMESERVER ZONE ACTION IP
HNAME
```

```
/usr/sbin/samba-dnssupdate.sh -m &
```

```
# End /etc/dhcpd/update.sh
EOF
```

Le damos permisos de ejecución al script

```
#chmod 750 /etc/dhcpd/update.sh
```

Configuramos servidor dhcpd siguiendo el artículo (<https://wiki.archlinux.org/index.php/Dhcpd>) y agregamos lo siguiente en todas las declaraciones de subnet en /etc/dhcpd.conf que provee el servicio de DHCP:

```
on commit {
    set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
    set ClientName = pick-first-value(option host-name, host-decl-name);
    execute("/etc/dhcpd/update.sh", "add", ClientIP, ClientName);
}
```

```
on release {
    set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
    set ClientName = pick-first-value(option host-name, host-decl-name);
    execute("/etc/dhcpd/update.sh", "delete", ClientIP, ClientName);
}
```

```
on expiry {
    set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
    set ClientName = pick-first-value(option host-name, host-decl-name);
    execute("/etc/dhcpd/update.sh", "delete", ClientIP, ClientName);
}
```

Aquí como nos debe quedar nuestro archivo de configuración /etc/dhcpd.conf para referencia:

```
# Begin /etc/dhcpd.conf
```

```
# Internal subnet
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.100 192.168.0.199;
    option subnet-mask 255.255.255.0;
    option routers 192.168.0.1;
    option domain-name "tudominio.cu";
    option domain-name-servers 192.168.0.2 192.168.0.3;
    option broadcast-address 192.168.0.255;
    default-lease-time 28800;
    max-lease-time 43200;
    authoritative;
```

```

on commit {
    set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
    set ClientName = pick-first-value(option host-name, host-decl-name);
    execute("/etc/dhcpd/update.sh", "add", ClientIP, ClientName);
}
on release {
    set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
    set ClientName = pick-first-value(option host-name, host-decl-name);
    execute("/etc/dhcpd/update.sh", "delete", ClientIP, ClientName);
}
on expiry {
    set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
    set ClientName = pick-first-value(option host-name, host-decl-name);
    execute("/etc/dhcpd/update.sh", "delete", ClientIP, ClientName);
}
}

# End /etc/dhcpd.conf

```

Finalmente, iniciamos o reiniciamos el servicio dhcpd4:

```
#service isc-dhcp-server restart
```

<<< SERVIDOR BDC (bdc.tudominio.cu) 192.168.0.3 >>>

Debemos realizar todos los pasos iguales que para el servidor PDC la única diferencia esta es que a la hora aprovisionar nuestro servidor samba4, le debemos especificar que este es miembro de un dominio que ya existe, de la siguiente manera.

El siguiente paso es el proceso de agregación de un controlador de dominio adicional o dcpromo. Añadir el server como controlador adicional:

```
# samba-tool domain join tudominio.cu DC -Uadministrator --realm=TUDOMINIO.CU --dns-backend=BIND9_DLZ
```

Si todo ha ido correctamente deberíamos ver en pantalla un mensaje tipo:

```

Finding a writeable DC for domain 'tudominio.cu'
Found DC pdc.tudominio.cu
Password for [WORKGROUP\administrator]:
workgroup is TUDOMINIO
realm is tudominio.cu
checking sAMAccountName
Adding CN=BDC,OU=Domain Controllers,DC=tudominio,DC=cu
Adding CN=BDC,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=tudominio, DC=cu
Adding CN=NTDS Settings,CN=BDC,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=tudominio, DC=cu
Adding SPNs to CN=BDC,OU=Domain Controllers,DC=tudominio, DC=cu
Setting account password for BDC$

```

Enabling account
 Adding DNS account CN=dns-BDC,CN=Users,DC=tudominio, DC=cu with dns/ SPN
 Setting account password for dns-BDC
 Calling bare provision
 No IPv6 address will be assigned
 Provision OK for domain DN DC=tudominio, DC=cu
 Starting replication
 Schema-DN[CN=Schema,CN=Configuration,DC=tudominio, DC=cu] objects[402/1550]
 linked_values[0/0]
 Schema-DN[CN=Schema,CN=Configuration,DC=tudominio, DC=cu] objects[804/1550]
 linked_values[0/0]
 Schema-DN[CN=Schema,CN=Configuration,DC=tudominio, DC=cu] objects[1206/1550]
 linked_values[0/0]
 Schema-DN[CN=Schema,CN=Configuration,DC=tudominio, DC=cu] objects[1550/1550]
 linked_values[0/0]
 Analyze and apply schema objects
 Partition[CN=Configuration,DC=tudominio, DC=cu] objects[402/1614] linked_values[0/0]
 Partition[CN=Configuration,DC=tudominio, DC=cu] objects[804/1614] linked_values[0/0]
 Partition[CN=Configuration,DC=tudominio, DC=cu] objects[1206/1614] linked_values[0/0]
 Partition[CN=Configuration,DC=tudominio, DC=cu] objects[1608/1614] linked_values[0/0]
 Partition[CN=Configuration,DC=tudominio, DC=cu] objects[1614/1614] linked_values[28/0]
 Replicating critical objects from the base DN of the domain
 Partition[DC=tudominio, DC=cu] objects[98/98] linked_values[28/0]
 Partition[DC=tudominio, DC=cu] objects[379/281] linked_values[31/0]
 Done with always replicated NC (base, config, schema)
 Replicating DC=DomainDnsZones,DC=tudominio, DC=cu
 Partition[DC=DomainDnsZones,DC=tudominio, DC=cu] objects[43/43] linked_values[0/0]
 Replicating DC=ForestDnsZones,DC=tudominio, DC=cu
 Partition[DC=ForestDnsZones,DC=tudominio, DC=cu] objects[18/18] linked_values[0/0]
 Partition[DC=ForestDnsZones,DC=tudominio, DC=cu] objects[36/18] linked_values[0/0]
 Committing SAM database
 Sending DsReplicateUpdateRefs for all the replicated partitions
 Setting isSynchronized and dsServiceName
 Setting up secrets database
 Unable to find group id for BIND,
 set permissions to sam.ldb* files manually
 Unable to find group id for BIND,
 set permissions to sam.ldb* files manually
 Joined domain TUDOMINIO (SID S-1-5-21-3426060274-2789830405-1143475183) as a DC

Luego continuamos los demás pasos, igual que hicimos en el servidor PDC

Ahora añadimos el registro PTR correspondiente a nuestro bdc a la zona inversa de nuestro DNS.

Nota: Este comando vamos a correrlo en la consola del pdc para que se replique al bdc

```
# kinit administrator@tudominio.cu

# samba-tool dns add tudominio.cu 0.168.192.in-addr.arpa 3 PTR bdc.tudominio.cu
```

Password for [administrator@tudominio.cu]:

Record added successfully

Cuando terminemos, arrancamos bind9 y samba4

Arrancamos el bind9

```
# named -u bind -4 -d 3 &
```

Revisamos los logs si todo funciona bien.

```
# tail -f -n 50 /var/log/syslog | ccze
```

Feb 24 21:49:45 bdc named[5183]: samba_dlz: configured writeable zone '_msdcs.tudominio.cu'

Feb 24 21:49:45 bdc named[5183]: set up managed keys zone for view _default, file 'managed-keys.bind'

Feb 24 21:49:45 bdc named[5183]: zone 'version.bind' allows updates by IP address, which is insecure

Feb 24 21:49:45 bdc named[5183]: zone 'hostname.bind' allows updates by IP address, which is insecure

Feb 24 21:49:45 bdc named[5183]: zone 'authors.bind' allows updates by IP address, which is insecure

Feb 24 21:49:45 bdc named[5183]: zone 'id.server' allows updates by IP address, which is insecure

Feb 24 21:49:45 bdc named[5183]: command channel listening on 127.0.0.1#953

Feb 24 21:49:45 bdc named[5183]: managed-keys-zone: loaded serial 0

Feb 24 21:49:45 bdc named[5183]: all zones loaded

Feb 24 21:49:45 bdc named[5183]: running

Debemos ver unas salidas como estas.

```
# samba -i -M single &
```

Copyright Andrew Tridgell and the Samba Team 1992-2015

samba: using 'single' process model

Attempting to autogenerate TLS self-signed keys for https for hostname 'BDC.tudominio.cu'

TLS self-signed keys generated OK

Copiamos los script de arranque de bind9 y samba9, los agregamos para los servicios arranquen con cuando arranque el sistema.

```
# update-rc.d samba4 defaults
# update-rc.d bind9 defaults
```

Realizamos todas y las mismas pruebas que hicimos para comprobar nuestro PDC

```
# host -t SRV _ldap._tcp.tudominio.cu.
```

_ldap._tcp.tudominio.cu has SRV record 0 100 389 bdc.tudominio.cu.

_ldap._tcp.tudominio.cu has SRV record 0 100 389 pdc.tudominio.cu.

```
# host -t SRV _kerberos._udp.tudominio.cu.
```

_kerberos._udp.tudominio.cu has SRV record 0 100 88 bdc.tudominio.cu.

_kerberos._udp.tudominio.cu has SRV record 0 100 88 pdc.tudominio.cu.

Notas:

Para una mejor interpretación de este manual aclaramos que:

Todas las líneas de comandos que se deben ejecutar en consola comienzan por el símbolo de número (#) y están en letra cursiva dentro de cuadros de texto, tener mucho cuidado y observación ya que hay otras líneas de comentarios dentro de las configuraciones que también comienzan con dicho símbolo.

Todos los archivos de configuraciones están en letra cursiva. Para una mejor diferenciación del texto y comentarios de este manual.

Espero que esta guía sirva para todos lo que tengan dudas y problemas para instalar un controlador de dominio usando Samba4.