 Si estamos instalando, configurando o explotando el servicio LDAP (OpenLDAP) y aparecen problemas ¿qué hacemos?

TÉCNICAS BÁSICAS

Necesitamos herramientas que nos permitan a preguntas como las siguientes:

1. ¿cómo puedo monitorizar exhaustivamente los **recursos en la máquina local** (ficheros, sockets TCP/IP, sockets IPC, ficheros mapeados a memoria, ...) de los que hace uso el servicio LDAP donde está corriendo y ver si falla algo? Por ejemplo:
   - slapd intenta acceder a un fichero pero no puede (permisos erróneos, nombre mal, bloqueado, ...)
   - slapd intenta abrir un socket TCP/IP pero no puede por alguna razón
   - slapd intenta abrir un socket IPC local pero no puede por alguna razón
   - slapd detecta un fallo en una base de datos de un backend
   - ...
2. ¿cómo puedo monitorizar exhaustivamente las **"transacciones" por la red TCP/IP** (o locales entre servicio LDAP y otros locales) a nivel de capa de aplicación (protocolo LDAP)? Por ejemplo:
   - consultas que realizan los clientes LDAP al servidor y respuestas que éste da.
   - problemas de permisos LDAP al acceder a ciertos objetos del árbol LDAP.
   - proceso de autenticación del cliente LDAP: con que identidad LDAP es autenticado (anónimo, usuario autenticado, ...)
   - contraseña que recibe realmente slapd para autenticar al usuario,....

Algunas de las técnicas básicas serán las siguientes. Pero hace falta algo de práctica y tiempo para familiarizarse con los logs de slapd y strace .....

Opciones de depuración del propio daemon slapd

Prácticamente todos los daemons en Gnu/Linux disponen de abundantes opciones de depuración (named, dhcpd, slapd, ...). La idea es la siguiente: arrancar slapd a mano, decirle que no suelte la consola y que vaya dando toda la información de depuración que necesitemos. Además, podemos guardar en un fichero el volcado para su análisis más detenido.

Por ejemplo:

```
#slapd -h "ldap:/// ldapi:///" -g openldap -u openldap -F /etc/ldap/slapd.d -d -1 \
  | tee depurar-ldap.txt
```

- recordemos que \ (sin nada después más que INTRO) es el carácter del shell de Linux para indicar que la introducción del comando continúa en la línea siguiente.
- la opción **`-d`** es la que indica al demonio que no pase a *background* y se quede pegado a la consola.
- el código numérico indica el tipo o nivel de detalles de depuración que se desea.
- debemos asegurarnos antes de ejecutarlo de que el slapd no está ya corriendo; por ejemplo:
  - **`service slapd stop`**
  - **`ps xa | grep slapd`** (si está corriendo **`kill -TERM <PID>`**)
- (si no recordamos toda sentencia anterior: si ya está corriendo slapd ejecutamos **`pstree -a`** y ya sólo hay que añidirle **`-d -1`** para tener toda la sentencia.)

Debugging Levels

(se pueden sumar para obtener información combinada: 256 + 128 = 384 = log connections + ACLS)

Level	Keyword	Description
-1	any	enable all debugging
0		no debugging
1	(0x1 trace)	trace function calls
2	(0x2 packets)	debug packet handling
4	(0x4 args)	heavy trace debugging
8	(0x8 conns)	connection management
16	(0x10 BER)	print out packets sent and received
32	(0x20 filter)	search filter processing
64	(0x40 config)	configuration processing
128	(0x80 ACL)	access control list processing
256	(0x100 stats)	stats log connections/operations/results
512	(0x200 stats2)	stats log entries sent
1024	(0x400 shell)	print communication with shell backends
2048	(0x800 parse)	print entry parsing debugging
16384	(0x4000 sync)	syncrepl consumer processing
32768	(0x8000 none)	only messages that get logged whatever log level is set

Por la consola (y en el archivo **`depurar-ldap.txt`**) van apareciendo todos los sucesos requeridos:

1. los mensajes de arranque de slapd
2. los mensajes de funcionamiento de slapd
3. los mensajes de parada de slapd

Ahora queda con paciencia y práctica ir aprendiendo a interpretar los mensajes, discernir posibles problemas y posibles soluciones. Al cabo de un poco se le pilla el truco...

Consejo: se aprende mucho si uno provoca un problema y luego mira en la información de depuración cómo se refleja el problema. Por ejemplo:

- cambiar permisos de algún fichero y ver que dice slapd al arrancar en modo depuración.
- modificar una ACL de ldap, ...

Depuración utilizando `strace`

Los pasos básicos serían:

1. Creamos un directorio donde meter los datos de depuración recogidos:
   `mkdir DEPURAR`
   `cd DEPURAR`

2. Ejecutamos slapd en modo depuración y con strace (podemos omitir -e read= all -e write=all si queremos...)

```
#strace -ff -s 655365 -v -y -o strace01.txt \
slapd -h "ldap:/// ldapi:///" -g openldap -u openldap -F /etc/ldap/slapd.d/ -d -1 2>&1 \
| tee slapd01.txt
```

   - **`-ff`**: strace también de los procesos hijo de slapd (sino perdemos información de depuración)
   - **`-s 65535`**: tamaño máximo de los volcados hexadecimales, cadenas de texto, etc.
   - podemos añadir, para obtener todavía más información de volcado: **`-e read=all -e write=all`**
   - **`2>&1:`** redirigir salida stderr a stdout (para que los mensajes de error salgan también en el fichero **`slapd01.txt`**)

3. Trabajamos contra LDAP.

4. Al acabar, con CTRL+C paramos el strace (o, si queremos ver como responde slapd a distintas señales del sistema con kill -TERM <pid-slapd>, kill -SIGTERM <pid-sldap>, etc. --> útil para ver como slapd hace el "shutdown")

Después con **`grep`**, o en nano (CTRL+W) buscaremos información sobre "cacert.pem", "server.key", "server.crt", "PERMISSION DENIED", etc. Ejemplos:

- open("/etc/ssl/certs/cacert.pem", O_RDONLY) = -1 ENOENT (No such file or directory)
- open("/etc/ssl/private/server.key", O_RDONLY) = -1 EACCES (Permission denied)
-

Modificar cómodamente la configuración de slapd (directivas olc: olcTLSCACertificate, olcTLS

Este método no es el más adecuado para un servidor slapd en producción, pero con propósitos didácticos y aprender a trabajar con slapd vale.

Ejemplo 1

Por ejemplo, supongamos que estamos configurando slapd para que utilice LDAPS. Hemos ejecutado:

```
ldapmodify -Y EXTERNAL -H ldapi://
Pegaremos os seguintes datos (é como cargalos dende un ficheiro). Imos modificar a rama cn=config:
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/server.crt
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/server.key (CTRL-D)
```

pero, algo falla y sólo nos carga la primera directiva **`olcTLSCACertificateFile`** (por ejemplo, porque teníamos mal el fichero de **`server.key`** o los permisos mal ...)
Podemos ver que sólo se cargó esa directiva olc con:

```
#ldapsearch -Y EXTERNAL -H ldapi:// -b 'cn=config'  |  grep "olcTLS"
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
```

Después parece que no hay manera de corregir el problema.

Podemos hacer lo siguiente:

1) Paramos el servicio slapd: **`#service slapd stop`**
2) Modificamos el fichero LDIF de configuración, borrando la línea de configuración de la directiva olcTLSCACertificate
`#nano /etc/ldap/slapd.d/cn\=config.ldif`
Borramos la línea marcada en fondo rojo:

```
# AUTO-GENERATED FILE - DO NOT EDIT!! Use ldapmodify.
# CRC32 938df8c4
dn: cn=config
objectClass: olcGlobal
cn: config
olcArgsFile: /var/run/slapd/slapd.args
olcLogLevel: none
olcPidFile: /var/run/slapd/slapd.pid
olcToolThreads: 1
structuralObjectClass: olcGlobal
entryUUID: 1800ab2c-945a-1035-9bae-4d6a83ec15df
creatorsName: cn=config
createTimestamp: 20160411175351Z
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
entryCSN: 20160419105503.063859Z#000000#000#000000
modifiersName: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
modifyTimestamp: 20160419105503Z
```

3) **`#service slapd start`**
(en los logs/depuración podremos ver que slapd detecta un error de CRC en el fichero, pero tira para adelante igual...)

Comprobamos que ya no está la directiva:

```
#ldapsearch -Y EXTERNAL -H ldapi:// -b 'cn=config'  |  grep "olcTLS"
```

(vacío)

Ejemplo 2

Por ejemplo, supongamos que queremos cargar las directivas olcTLS pero sin utilizar **`ldapmodify`** (recordemos que estamos ahora en un contexto de aprendizaje y experimentación).

1) Paramos el servicio slapd:

```
service slapd stop
```

2) Modificamos el fichero LDIF de configuración añadiendo las líneas con fondo amarillo:

```
#nano /etc/ldap/slapd.d/cn\=config.ldif
```

```
# AUTO-GENERATED FILE - DO NOT EDIT!! Use ldapmodify.
# CRC32 938df8c4
dn: cn=config
objectClass: olcGlobal
cn: config
olcArgsFile: /var/run/slapd/slapd.args
olcLogLevel: none
olcPidFile: /var/run/slapd/slapd.pid
olcToolThreads: 1
structuralObjectClass: olcGlobal
entryUUID: 1800ab2c-945a-1035-9bae-4d6a83ec15df
creatorsName: cn=config
createTimestamp: 20160411175351Z
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
olcTLSCertificateFile: /etc/ssl/certs/server.crt
olcTLSCertificateKeyFile: /etc/ssl/private/server.key
entryCSN: 20160419105503.063859Z#000000#000#000000
modifiersName: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
modifyTimestamp: 20160419105503Z
```

3) Arrancamos el servicio slapd (en los logs podremos ver que slapd detecta un fallo de CRC en el fichero, pero tira para adelante igual...)

```
service slapd start
```

Comprobamos que las directivas están cargadas:

```
ldapsearch -Y EXTERNAL -H ldapi:// -b 'cn=config'  |  grep "olcTLS"
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
olcTLSCertificateFile: /etc/ssl/certs/server.crt
olcTLSCertificateKeyFile: /etc/ssl/private/server.key
```

EJEMPLO 1: PROBLEMAS CONFIGURANDO ACCESO CON SEGURIDAD TLS

PRUEBA	ESCENARIO	¿Funciona?	PISTAS Y SOLUCIÓN
1	Todo configurado correctamente	SÍ	cat /etc/ldap/slapd.d/cn=config contiene las tres directivas olcTLS.
2	Ficheros de certificados no están en su directorio (falta uno o más archivos)	NO	**Error ldapmodify**: `ldap_modify: Other (e.g., implementation specific) error (80)` Buscar en captura **depuración strace** patrón parecido a: open("/etc/ssl/certs/cacert.pem", O_RDONLY) = -1 ENOENT (No such file or directory)
3	Permisos de archivos de certificados cacert.pem y server.crt: no importan permisos (slapd accede como root) server.key: sí importan permisos	NO	**Error ldapmodify**: `ldap_modify: Other (e.g., implementation specific) error (80)` Buscar en captura **depuración strace** patrón parecido a: open("/etc/ssl/private/server.key", O_RDONLY) = -1 EACCES (Permission denied)
4	Certificados no válidos: la server.key no se corresponde con el server.crt	NO	**Error ldapmodify**: `ldap_modify: Other (e.g., implementation specific) error (80)` No fui capaz de encontrar ningún patrón ni en depuración strace ni en depuración slapd ([FALTA])
5	Certificados no válidos: certificado caducado	SÍ	Funciona bien (falta ver si se puede configurar el cliente LDAP para que verifique los certificados: CA RAIZ, marcas de tiempo, etc.).
6	Certificados no válidos: el certificado de la CA no se corresponde con el server.crt	NO	**Error ldapmodify**: `ldap_modify: Undefined attribute type (17)` `additional info: olcCACertificateFile: attribute type undefined` **Depuración slapd:** **57166b6b conn=1001 op=1 MOD dn="cn=config"** **57166b6b conn=1001 op=1 MOD attr=olcCACertificateFile olcTLSCertificateFile olcTLSCertificateKeyFile** **57166b6b send_ldap_result: conn=1001 op=1 p=3** **57166b6b send_ldap_result: err=17 matched="" text="olcCACertificateFile: attribute type undefined"** **57166b6b send_ldap_response: msgid=2 tag=103 err=17**
7	Cambio del orden de carga de directivas de configuración OLC TLS: 1. olcTLSCertificateKeyFile, 2. olcTLSCertificateFile 3. olcTLSCACertificateFile	SÍ	*Observación: las directivas aparecen en /etc/ldap/slapd.d/cn=config.ldif en el mismo orden en que fueron cargadas.*
8	Cambio del orden de carga de directivas de configuración OLC TLS: 1. olcTLSCertificateKeyFile, 2. olcTLSCACertificateFile 3. olcTLSCertificateFile	SÍ	*Observación: las directivas aparecen en /etc/ldap/slapd.d/cn=config.ldif en el mismo orden en que fueron cargadas.*
9	Cambio del orden de carga de directivas de configuración OLC TLS: 1. olcTLSCertificateFile, 2. olcTLSCertificateKeyFile 3. olcTLSCACertificateFile	SÍ	*Observación: las directivas aparecen en /etc/ldap/slapd.d/cn=config.ldif en el mismo orden en que fueron cargadas.*
10	Omitir la directiva olcTLSCACertificateFile	SÍ	*Observación: slapd arrancó perfectamente sin cargar directiva olcTLSCACertificateFile y el cliente uclient01 se conectó por LDAPS sin problema.*
11	Caracteres extraños en LDIF de carga de directivas de configuración OLC.	----	No admite `--` en vez de `-` para separar la carga de directivas. Admite cualquier número de espacios entre los : y el valor de la directiva. Admite cualquier número de tabulaciones entre los : y el valor de la directiva.

EJEMPLO 2: OTROS PROBLEMAS

	ESCENARIO	PISTAS Y SOLUCIÓN
1		
2		
3		