AD Rettung

Die Übernahme eines bestehenden ActiveDiretories beispielsweise beim Upgrade von Sernet-Samba Paketen hin zu unseren eigenen, kann durchaus auch schief gehen. Uns ist das inzwischen mehrfach passiert. Beschädigt war in aller Regel das ID-Mapping zwischen den Windows-SIDs und den UNIX UID/GIDs. Im Samba-Log haben sich dann immer nachfolgende Einträge zu Hauf gezeigt:

```
[2017/09/11 12:45:36.109347,  0]
../source4/auth/unix_token.c:107(security_token _to_unix_token)
   Unable to convert SID (S-1-1-0) at index 3 in user token to a GID.
Conversion was returned as type 0, full token:
[2017/09/11 12:45:36.109486,  0]
../libcli/security/security_token.c:63(security_token_debug)
   Security token SIDs (8):
     SID[  0]: S-1-5-21-477667662-2079173432-720341635-1110
     SID[  1]: S-1-5-21-477667662-2079173432-720341635-513
     SID[  2]: S-1-5-21-477667662-2079173432-720341635-1108
     SID[  3]: S-1-1-0
     SID[  4]: S-1-5-2
     SID[  5]: S-1-5-11
     SID[  6]: S-1-5-32-545
     SID[  7]: S-1-5-32-554
    Privileges (0x          800000):
     Privilege[  0]: SeChangeNotifyPrivilege
    Rights (0x             400):
     Right[  0]: SeRemoteInteractiveLogonRight
```

Eine Möglichkeit dies zu reparieren haben wir noch nicht gefunden, dafür aber einen anderen sehr eleganten Ausweg.

Solange die Datei **`sam.ldb`** im Verzeichnis

```
/var/lib/samba/private
```

nicht beschädigt ist, können alle darin enthaltenen Informationen wie Benutzerkonten, Maschinenkonten usw. in ein neues AD übernommen werden. Sichern Sie sich diese Datei evtl. auf dem neuen System in ein Arbeitsverzeichnis.

Voraussetzung für die Übernahme der Informationen ist, dass das neue AD den gleichen **Domain Secure Identifier** (Domain SID) trägt wie das zu rettende.

Das kann bereits beim Domain-Provisioning geschehen oder nachträglich erledigt werden.

Domain SID

Lesen Sie zunächst die SID des beschädigten ADs aus. Solange es noch aktiv ist, ist das einfach:

```
invis:~ # net getdomainsid
SID for domain INVIS-NET is: S-1-5-21-768460775-4665083570-3002063577
```

Ist das AD nicht mehr aktiv, müssen Sie die SID aus der Datei „sam.ldb“ extrahiert werden:

```
invis:~ # ldbsearch -H /var/lib/samba/private/sam.ldb -b "DC=INVIS-
NET,DC=LOC" objectClass=domainDNS |grep objectSid |cut -d " " -f2
S-1-5-21-768460775-4665083570-3002063577
```

Möchten Sie die Domain-SID bereits beim Provisioning setzen, müssen Sie die folgende Option in den Befehlsaufruf einbauen:

```
--domain-sid=S-1-5-21-768460775-4665083570-3002063577
```

Um das während des invis-Setups vorzunehmen müssen Sie im Script

```
/usr/share/sine/scripts/samba_ad
```

die Zeile 111 entsprechend erweitern:

```
...
samba-tool domain provision --realm="$ad_realm" --domain-
sid=S-1-5-21-768460775-4665083570-3002063577 --domain="$ad_domain" --host-
ip="$ad_ip"
--adminpass="$ad_adminpass" --server-role="$ad_server_role" --use-rfc2307 --
dns-backend="BIND9_DLZ" 2>&1 | tee -a $LOGFILE | pgbox
provresult=${PIPESTATUS[0]}
...
```

Alternativ, können Sie die SID auch nachträglich mit:

```
invis:~ # net setdomainsid S-1-5-21-768460775-4665083570-3002063577
```

ändern.

Daten aus altem AD extrahieren und anpassen

Aus dem alten ActiveDirectory müssen Informationen extrahiert werden, die sich an unterschiedlichen Stellen im LDAP-Verzeichnisbaum befinden. Folgende Informationen werden benötigt:

1. **CN=additionalUserInformation,CN=invis-server,DC=invis-net,dc=loc** - Hier befinden sich die Zugangsdaten und Mailrouting Informationen der Email-Konten aller Benutzer.
2. **CN=DHCP Config,CN=DHCP-Server,CN=invis-Server,DC=doc-net,DC=loc** - Hier befinden sich die Informationen des DHCP-Servers
3. **CN=Computers,DC=invis-net,DC=loc** - Hier werden die Maschinenkonten gespeichert.
4. **CN=Users,DC=invis-net,DC=loc** - Hier befinden sich die Benutzer und Gruppenkonten.

Hinweis: *Achten Sie darauf, dass Sie die Basis (DC=invis-net,DC=loc) des LDAP-Verzeichnisses entsprechend Ihrer Domain anpassen.*

Zum Extrahieren der Informationen kommt das Tool ***ldbsearch*** zum Einsatz. Im Anschluss an die Extraktion der Daten müssen diese Teilweise manuell angepasst werden, bevor Sie in das neue AD importiert werden können. Ergebnis der Extraktion ist jeweils eine LDIF-Datei.

Bei der Extraktion ist weiterhin zu beachten, dass alle extrahierten Objekte ein Attribut namens „objectGUID“ enthält. Dieses Attribut darf beim Import nicht mehr erhalten sein, da es automatisch beim erzeugen neuer Einträge im LDAP-Verzeichnis des AD erzeugt wird.

Um es von vorne herein zu eliminieren, lässt es sich mit „grep -v objectGUID“ schon bei der Extraktion herausfiltern. Alle nachfolgenden Extraktionsschritte werden am besten direkt im Arbeitsverzeichnis in dem Ihre alte **`sam.ldb`** liegt ausgeführt.

Extraktion der Mailkonten-Informationen

Aus dem oben genannten LDAP-Knoten Nr. 1 werden die notwendigen Informationen in drei Schritten extrahiert. Jeder Benutzer, der über ein externes Mailkonto verfügt, hat unterhalb des Knotens 1 einen eigenen Untercontainer. Dieser Untercontainer enthält dann wiederum die eigentlichen Mailkonten- und Mailrouting-Informationen.

Im ersten Schritt extrahieren wir die Unterknoten der Benutzer:

```
invis:~/arbeitsverzeichnis # ldbsearch -H sam.ldb -b
CN=additionalUserInformation,CN=invis-server,DC=invis-net,dc=loc
objectClass=container | grep -v objectGUID > 01_userinfocontainer.ldif
```

Die gewonnene LDIF-Datei enthält jetzt noch einen Hauptknoten, der händisch gelöscht werden muss, da er bereits im neuen AD enthalten ist. Bei meinen Tests war dies immer der letzte Eintrag in der Datei. Öffnen Sie die LDIF-Datei in einem Editor und **entfernen** Sie den entsprechenden Block:

```
# record 6
dn: CN=AdditionalUserInformation,CN=invis-Server,DC=invis-net,DC=loc
objectClass: top
objectClass: container
cn: AdditionalUserInformation
description: Basis fuer ergaenzende Benutzerinrfomationen - cornaz
instanceType: 4
whenCreated: 20180329133814.0Z
whenChanged: 20180329133814.0Z
uSNCreated: 5011
uSNChanged: 5011
showInAdvancedViewOnly: TRUE
name: AdditionalUserInformation
objectCategory: CN=Container,CN=Schema,CN=Configuration,DC=invis-net,DC=loc
distinguishedName: CN=AdditionalUserInformation,CN=invis-Server,DC=invis-
net,DC=
 loc
```

Im zweiten Schritt werden die Mailkonten-Informationen extrahiert:

```
invis:~/arbeitsverzeichnis # ldbsearch -H sam.ldb -b
CN=additionalUserInformation,CN=invis-server,DC=invis-net,dc=loc
objectClass=fspFetchMailAccount | grep -v objectGUID >
02_usermailkonten.ldif
```

Im dritten und letzten Schritt werden noch die ergänzenden Mailrouting-Informationen extrahiert:

```
invis:~/arbeitsverzeichnis # ldbsearch -H sam.ldb -b
CN=additionalUserInformation,CN=invis-server,DC=invis-net,dc=loc
objectClass=fspLocalMailRecipient | grep -v objectGUID >
03_usermailrouting.ldif
```

Damit sind alle Informationen rund um die Mailkonten der Benutzer komplett.

DHCP- und DNS-Informationen extrahieren

Bei den Informationen für DHCP- und DNS-Server liegt der Fall ein wenig anders. Extrahieren, wie oben gezeigt, lassen sich lediglich die Daten des DHCP-Servers. Die des DNS-Servers liegen in einer Form vor, die sich so nicht in übertragbare Form extrahieren lassen.

Für die Übertragung genügen aber Hostname, IP-Adresse, MAC-Adresse sowie ggf. der Standort des IP-Gerätes. Diese werden in eine Liste übernommen, die dann wieder mit dem Script **`hostadd2ad`** ins neue AD übernommen wird.

Im ersten Schritt erfolgt aber zunächst Extraktion der Informationen mittels **`ldbsearch`**:

```
invis:~/arbeitsverzeichnis # ldbsearch -H sam.ldb -b "CN=DHCP
Config,CN=DHCP-Server,CN=invis-Server,DC=doc-net,DC=loc"
objectClass=iscDhcpHost > hosts.ldif
```

Der nächste Schritt ist Handarbeit. Sie müssen die erzeugte Datei zum Lesen öffnen. Auch hier sind die einzelnen Objekte leicht zu erkennen. Aus jedem Objekt sind jetzt die oben genannten Informationen auszulesen und in eine einfache Textdatei zu übernehmen.

Hier ein Beispielobjekt:

```
# record 1
dn: CN=verwaltung-2014,CN=DHCP Config,CN=DHCP-Server,CN=invis-
Server,DC=invis-net,DC=loc
objectClass: top
objectClass: iscDhcpHost
cn: verwaltung-2014
instanceType: 4
whenCreated: 20180329170133.0Z
whenChanged: 20180329170133.0Z
uSNCreated: 5227
```

```
uSNChanged: 5227
name: verwaltung-2014
objectGUID: c5697b81-7d30-47c8-b947-f0f1efc9720d
objectCategory: CN=iscDhcpHost,CN=Schema,CN=Configuration,DC=invis-
net,DC=loc
iscDhcpStatements: fixed-address 192.168.230.123
iscDhcpHWAddress: ethernet 74:d4:35:5a:28:e4
iscDhcpComments: Anmeldung links
distinguishedName: CN=verwaltung-2014,CN=DHCP Config,CN=DHCP-
Server,CN=invis-S
 erver,DC=invis-net,DC=loc
```

Übernehmen Sie die Informationen in eine Datei folgender Form:

```
74:d4:35:5a:28:e4,192.168.230.123,anmeldung-2014,Anmeldung links
...
```

Wenn es sich nicht um hunderte Hosts handelt, lässt sich dies zu Fuß erledigen. Irgendwann kommt aber sicherlich der Zeitpunkt, wo evtl. ein selbst geschriebenes Script diesen Job vereinfachen könnte.

Damit sind auch hier alle Informationen für die Übernahme ins neue AD zusammengetragen.

Maschinen-Konten extrahieren

Das extrahieren der Maschinenkonten ist ebenfalls ein bisschen komplizierter als die Informationen rund um die Mailkonten der Benutzer.

Neben der **objectGUID** müssen mit den Attributen **isCriticalSystemObject** und **primaryGroup** weitere Attribute entfernt und mit **pwdLastSet** ein Attribut manuell geändert werden.

Zunächst die angepasste Extraktion:

```
invis:~/arbeitsverzeichnis # ldbsearch -H sam.ldb -b "CN=Computers,DC=invis-
net,DC=loc" objectClass=Computer |grep -v objectGUID |grep -v primaryGroup
|grep -v isCriticalSystemObject > 04_machines.ldif
```

Öffnen Sie jetzt die soeben erzeugte Datei in einem Editor und setzen Sie für alle enthaltenen Objekte den Wert des Attributes „pwdLastSet“ auf **-1** oder **0**. Die vorhandenen realen Werte sind für den Import nicht zulässig. Dabei bedeutet der Wert **0**, dass das Passwort des Clients abgelaufen ist und der Wert **-1** bedeutet, dass das es Maschinen-Kontos niemals abläuft. Einen Unterschied macht das leider nicht:

Hintergrundwissen

Beim Extrahieren der Konto-Informationen ist das Passwort-Attribut nicht Teil des Ergebnisses. Es fehlt schlicht. Somit nutzt es auch nichts einen realen Zeitstempel für die letzte Passwort-Änderung anzugeben.

Gespeichert werden die Passwörter (das gilt auch für Benutzerpasswörter) Bas64-Codiert im Attribut **`unicodePwd`**. Dieses lässt sich zwar gesondert auslesen:

```
invis:~/arbeitsverzeichnis # ldbsearch -H sam.ldb -b CN=Computers,DC=invis-net,DC=loc unicodePwd
....
# record 4
dn: CN=balrog,CN=Computers,DC=invis-net,DC=loc
unicodePwd:: A7pFWs7Kbh010J/E3kFAoQ==
....
```

...aber fügt man dieses Attribut einfach in die Ergebnis-LDIF-Datei ein und versucht einen Import, endet dies mit folgender Fehlermeldung:

```
Unwilling to perform : "setup_io: it's not allowed to set the NT hash password directly'"
```

Da also die Übernahme des Passworts, nicht möglich ist geht die Vertrauensstellung zwischen Computer und Domäne in jedem Fall verloren und muss durch einen erneuten Domänenbeitritt wiederhergestellt werden. (Es sei angemerkt, dass das originale Passwort vom Computer zufällig generiert wurde, ist es schlicht unbekannt und kann also auch nicht nachträglich gesetzt werden!)

Wem hier eine bessere Lösung einfällt, der darf sich gerne bei uns melden!

Evtl. enthalten die einzelnen Maschinenkonten auch Unterknoten mit Informationen zu Druckerwarteschlangen freigegebener Drucker. Diese können gesondert extrahiert werden:

```
invis:~/arbeitsverzeichnis # ldbsearch -H sam.ldb -b "CN=Computers,DC=invis-net,DC=loc" objectClass=connectionPoint| grep -v objectGUID > 05_connectionPoints.ldif
```

Diese Datei muss nicht weiter angepasst werden.

Extraktion der Benutzerkonten und Gruppen

Beim extrahieren der Benutzerkonten ist ein bisschen mehr Kreativität gefragt, schließlich verfügt das neue AD ja bereits über einige Benutzerkonten wie etwa die Konten Administrator und Guest. Diese müssen nicht aus dem alten AD extrahiert und übernommen werden.

Es empfiehlt sich hier unerwünschte Konten mittels Filtern von vorne herein unerwünschte Konten auszuschließen.

Hinweis:*In Sachen Filter verhält sich **ldbsearch** wie sein Pendant **ldapsearch**. Wer also nachlesen möchte wie ein Suchfilter aufgebaut wird, sollte gleich nach Dokumentationen von **ldapsearch** suchen, **ldbsearch** ist diesbezüglich weit weniger gut dokumentiert.*

Im nachfolgenden Beispiel werden die Benutzerkonten „Administrator“, „dns-*hostname*“ (hier ist *hostname* durch den Hostnamen des AD-Servers zu ersetzen), „krbtgt“, „Guest“, „junk“ und „ldap.admin“ bei der Suche herausgefiltert:

```
invis:~/arbeitsverzeichnis # ldbsearch -H sam.ldb -b "CN=Users,DC=doc-net,DC=loc" '(&(objectClass=user) (!(samAccountName=junk)) (!(samAccountName=Administrator)) (!(samAccountName=dns-mars)) (!(samAccountName=krbtgt)) (!(samAccountName=ldap.admin)))'  |grep -v objectGUID |grep -v primaryGroupID > 06_users.ldif
```

Auch hier werden die Attribute „objectGUID“ und „primaryGroupID“ aus der Extraktion von vorne herein eliminiert.

Gehen sie die Ergebnisdatei in einem Editor manuell durch. Entfernen Sie alle Konten, die Sie nicht im neuen AD brauchen. Setzen Sie bei allen verbleibenden Konten das Attribut „pwdLastSet“ wie zuvor bei den Maschinenkonten auf den Wert **-1** oder **0**.

Da hier prinzipiell das gleiche gilt wie oben unter „Hintergrundwissen“ im Abschnitt „Extraktion der Maschinenkonten“ beschrieben, spielt es auch hier keine Rolle, für welchen Wert Sie sich entscheiden. Sie müssen die Passwörter in jedem Falle neu setzen. Im Falle der Benutzer ist das allerdings kein ganz so großes Problem. Sie können mit einem einfachen Script einfach ein Standard-Passwort setzen und dies den Usern, mit der Bitte es unmittelbar neu zu setzen, mitteilen. Im Moment in dem Sie ein Passwort setzen, ändert sich auch unmittelbar der Wert des Attributs **pwdLastSet**. Ändern Sie ihn nachträglich für alle User wieder auf **0** werden sie bei der Erstanmeldung gezwungen ihr Passwort neu zu setzen.

Damit sind auch die Benutzerkonten erfolgreich exportiert.

Bei den Gruppen wird auf gleiche Weise verfahren. Hier sollten sie lediglich die Gruppen in der Exportdatei haben, die sie selbst auf dem alten Server nach der Installation angelegt haben. Hier ist die Anzahl der herauszufilternden Gruppen allerdings so groß, dass der Filterausdruck recht lang wird. Alternativ können Sie auch ohne Filter arbeiten und die Ergebnisdatei vollständig von Hand bereinigen. Ein guter Ansatz ist es nur nach Gruppen zu suchen, die das Attribut **msSFU30Name** enthalten. Das ist bei allen Gruppen, die über das invis-Portal angelegt wurden der Fall:

```
invis:~/arbeitsverzeichnis # ldbsearch -H sam.ldb -b "CN=Users,DC=doc-net,DC=loc" '(&(objectClass=group)(msSFU30Name=*))' |grep -v objectGUID > 07_gruppen.ldif
```

Auch hier sollten Sie die Ergebnisdatei manuell kontrollieren oder nachbearbeiten.

Damit sind alle Informationen für die Übernahme ins neue AD komplett.

Daten in neues AD importieren

Das importieren der Daten ist weit weniger aufwendig als die Extraktion, wichtig ist hier vor allem die Reihenfolge in der Die vorbereiteten Dateien ins AD eingepflegt werden. Wenig überraschend, dass ich die LDIF-Dateien in der Reihenfolge nummeriert habe in der sie importiert werden müssen.

Es kann beim Importieren nicht schaden, wenn Sie nach jedem Importschritt eine Sicherung des ADs

anlegen. invis-Server haben dafür das Script ***adbackup*** im Gepäck. Das Tool packt das gesamte Verzeichnis

```
/var/lib/samba/
```

in ein „tar.gz“ Archiv und speichert dieses nach:

```
/srv/shares/archiv/sicherungen/vollsicherungen/ad
```

Hinweis: *Ab invis-Version 13.1 schreibt* **adbackup** *neben dem Datum auch die Uhrzeit der Sicherung in den Namen des Archivs, bei älteren Versionen lediglich das Datum. Sie können also mit der aktuellen Version des Scripts beliebig viele Sicherungen hintereinander vornehmen ohne Gefahr zu laufen eine bestehende Sicherung zu überschreiben.*

Das Script ***adbackup*** wird einfach ohne weitere Optionen auf der Kommandozeile aufgerufen. Erstellen Sie die erste Sicherung vor dem ersten Import-Schritt. Mit den Sicherungen können Sie im Falle eines Problems immer wieder auf den vorigen Zustand zurück springen.

Importiert wird mit dem Tool ***ldbadd*** und ist absolut simpel:

```
invis:~/arbeitsverzeichnis # ldbadd -v -H /var/lib/samba/private/sam.ldb
01_userinfocontainer.ldif
```

Die Reihenfolge hier noch einmal vollständig:

1. Benutzerknoten für Mailkonten- und Mailrouting-Informationen (01_userinfocontainer.ldif)
2. Mailkonten-Informationen (02_usermailkonten.ldif)
3. Mailrouting-Informationen (03_usermailrouting.ldif)
4. DHCP- und DNS-Informationen
5. Maschinenkonten (04_machines.ldif)
6. Ggf.: vorhandene Druckerfreigaben (05_connectionPoints.ldif)
7. Benutzerkonten (06_users.ldif)
8. Gruppen (07_gruppen.ldif)

Punkt 4 wird wie bereits erwähnt mit unserem eigenen Tool ***hostadd2ad*** erledigt, aber auch das ist denkbar einfach:

```
invis:~/arbeitsverzeichnis # hostadd2ad hostliste.txt
```

Kontrollieren Sie das AD nach jedem Schritt. Über die Administrationsseite des invis-Portals können Sie über den Link „Verzeichnisdienst“ die Appklikation **phpldapadmin** nutzen um sich Ihr AD anzuschauen.

Damit sind alle erforderlichen Informationen ins neue AD übernommen.

Letzte Handgriffe

Benutzerpasswörter

Zur Nacharbeit zählt natürlich das Setzen neuer Passwörter für die Benutzer. Es empfiehlt sich (sollten Sie als Admin nicht *zufällig* im Besitz aller Passwörter sein 😉) ein Standard-Passwort zu setzen:

```
invis:~ # sudo samba-tool user setpassword TestUser2 --newpassword=passw0rd
--must-change-at-next-login
```

Daraus lässt sich recht schnell ein Script stricken, welches dieses Komanndo für alle vorhandenen Benutzer ausführt. Das überlasse ich Ihrer Fantasie.

rIDNextRID

Der zweite **sehr wichtige** Punkt ist, dass Sie Ihrem AD den letzten vergebene RID (relative Identifier) mitteilen. Der RID ist Bestandteil des SID (Security Identifier). Der SID eines Benutzers, einer Gruppe oder eines Maschinenkontos identifiziert dieses Objekt innerhalb der Domäne eindeutig. Er besteht aus einem statischen (dem Domain SID) und einem dynamischen Teil (RID). Dabei ist der RID einfach ein Wert bei für jedes neu angelegte Objekt jeweils um den Wert **1** hoch gezählt wird.

Suchen Sie in der alten **`sam.ldb`** einfach nach dem dort vergebenen Wert:

```
invis:~/arbeitsverzeichnis # ldbsearch -H sam.ldb -b "CN=RID
Set,CN=IhrDC,OU=Domain Controllers,DC=invis-NET,DC=loc" rIDNextRID
# record 1
dn: CN=RID Set,CN=MARS,OU=Domain Controllers,DC=invis-net,DC=loc
rIDNextRID: 1141

# returned 1 records
# 1 entries
# 0 referrals
```

Hinweis: *Ersetzen Sie „IhrDC“ durch den Hostnamen Ihres AD Domain Controllers.*

Im hier gezeigten Beispiel ergab die Suche den Wert **1141**. Diesen müssen Sie an gleicher Stelle in Ihrem neuen AD einsetzen. Das können Sie entweder über **phpldapadmin** oder mittels ***ldbedit*** erledigen.

Dabei ist letzteres ein wirklich praktisches Tool. Es öffnet eine LDB-Datei an einer Stelle Ihrer Wahl im Editor Ihrer Wahl. Sie können damit LDB-Dateien bearbeiten, als wären es einfache ASCII-Dateien. Es versteht sich von selbst, dass Samba dabei besser nicht aktiv ist und Sie Ihr AD zuvor mit ***adbackup*** sichern sollten.

Geben Sie beim Kommandoaufruf keinen Editor an, wird (was wohl auch sonst) der ***vi*** verwendet.

Achtung: *Sie bearbeiten jetzt nicht die gesicherte **`sam.ldb`** des alten ADs, sondern die des neuen aktiven AD!*

```
invis:~ # ldbedit -H /var/lib/samba/private/sam.ldb -b "CN=RID
Set,CN=IhrDC,OU=Domain Controllers,DC=doc-NET,DC=loc"
```

Sollten Sie ein Problem mit ***vi*** haben, können Sie mit folgender Zeile Ihren Lieblingseditor bestimmen:

```
invis:~ # ldbedit -e lieblingseditor -H /var/lib/samba/private/sam.ldb -b
"CN=RID Set,CN=IhrDC,OU=Domain Controllers,DC=doc-NET,DC=loc"
```

Damit haben Sie Ihr AD gerettet. Viel Spaß bei den anstehenden Domänenbeitritten.... 😉

From:
https://wiki.invis-server.org/ - **invis-server.org**

Permanent link:
https://wiki.invis-server.org/doku.php?id=invis_server_wiki:upgrade:rescuead&rev=1522959726

Last update: **2018/04/05 20:22**