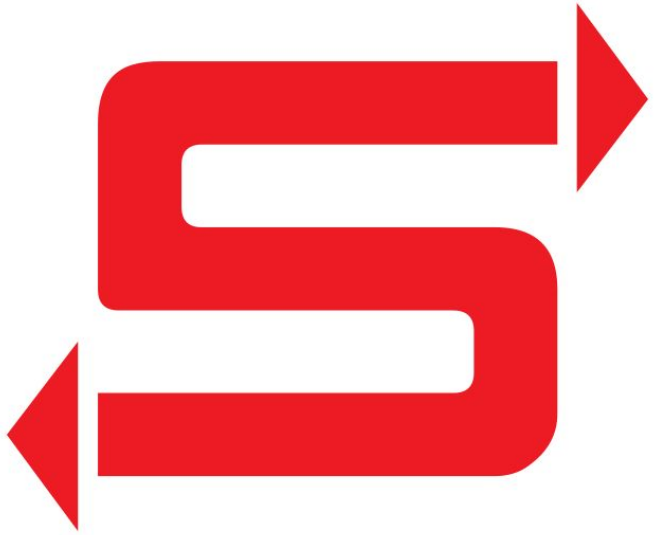
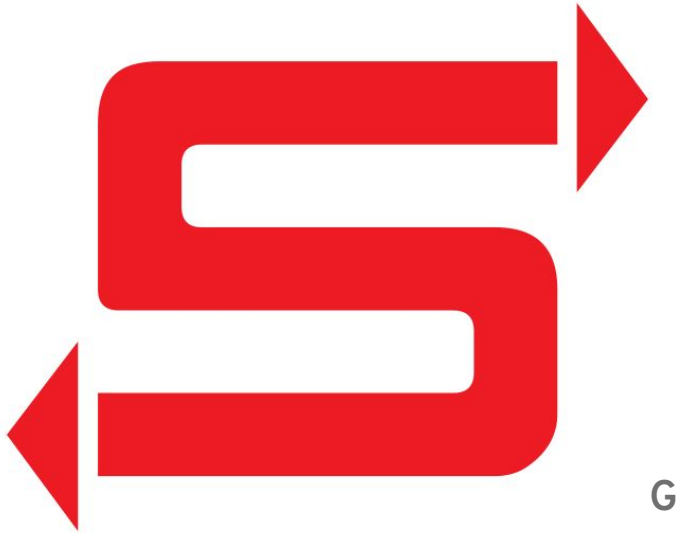


Serve yourself!

Winbind improvements for user token acquisition

Günther Deschner



Günther Deschner

- **Software Engineer Manager at IBM**
- Samba Team member since 2004
- Manages Ceph-SMB team

Where are my AD group memberships?

- Winbind's main mission:
  - Representation of Active Directory account data on Unix systems
- Expectation:
  - Availability of full group membership list
  - Completeness and accurateness of group membership
  - Independent of previous authentication
  - `id W2K25DOM\\gd` should always give correct and accurate output
- User and groups are defined by SIDs, names as well as logon properties (principals, etc.) can change
- Unix needs to map SIDs to Unix IDs (idmapping)

DC Authorization Data

- Domain Controllers attach authorization data to successful authentication (when requested)
- NTLMSSP:
  - DCE/RPC `netr_Validation` returned via NETLOGON
    - `netr_LogonSamLogon()`
    - `netr_LogonSamLogonEx()`
    - `netr_LogonSamLogonWithFlags()`
- Kerberos:
  - PAC_LOGON_INFO returned from AS-REP, TGS-REP calls
- Common subset of returned authorization data:
  - `netr_SamInfo3`

info3

```
netr_SamInfo3: struct netr_SamInfo3
    base: struct netr_SamBaseInfo
        logon_time               : Fri Jan  9 14:50:45 2026 CET
        logoff_time              : Thu Sep 14 04:48:05 30828 CEST
        kickoff_time             : Thu Sep 14 04:48:05 30828 CEST
        last_password_change     : Fri Jun 20 21:50:53 2025 CEST
        allow_password_change    : Sat Jun 21 21:50:53 2025 CEST
        force_password_change    : Thu Sep 14 04:48:05 30828 CEST
        account_name: struct lsa_String
            length                   : 0x0004 (4)
            size                     : 0x0004 (4)
            string                   : *
                string                   : 'gd'
        full_name: struct lsa_String
            length                   : 0x0022 (34)
            size                     : 0x0022 (34)
            string                   : *
                string                   : 'Guenther Deschner'
```

info3

```
logon_script: struct lsa_String
     length                   : 0x0000 (0)
     size                     : 0x0000 (0)
     string                   : *
         string                   : ''
 profile_path: struct lsa_String
     length                   : 0x0000 (0)
     size                     : 0x0000 (0)
     string                   : *
         string                   : ''
 home_directory: struct lsa_String
     length                   : 0x0000 (0)
     size                     : 0x0000 (0)
     string                   : *
         string                   : ''
 home_drive: struct lsa_String
     length                   : 0x0000 (0)
     size                     : 0x0000 (0)
     string                   : *
         string                   : ''
```

info3

```
logon_count              : 0x0029 (41)
bad_password_count       : 0x0000 (0)
rid                      : 0x00000450 (1104)
primary_gid              : 0x00000201 (513)
groups: struct samr_RidWithAttributeArray
    count                    : 0x00000003 (3)
    rids                     : *
        rids: ARRAY(3)
            rids: struct samr_RidWithAttribute
                rid                      : 0x00000201 (513)
                attributes               : 0x00000007 (7)
                       1: SE_GROUP_MANDATORY
                       1: SE_GROUP_ENABLED_BY_DEFAULT
                       1: SE_GROUP_ENABLED
                       0: SE_GROUP_OWNER
                       0: SE_GROUP_USE_FOR_DENY_ONLY
                       0: SE_GROUP_INTEGRITY
                       0: SE_GROUP_INTEGRITY_ENABLED
                       0: SE_GROUP_RESOURCE
                    0x00: SE_GROUP_LOGON_ID         (0)
```

info3

```
rids: struct samr_RidWithAttribute
        rid                     : 0x00000458 (1112)
        attributes              : 0x00000007 (7)
                1: SE_GROUP_MANDATORY
                1: SE_GROUP_ENABLED_BY_DEFAULT
                1: SE_GROUP_ENABLED
                0: SE_GROUP_OWNER
                0: SE_GROUP_USE_FOR_DENY_ONLY
                0: SE_GROUP_INTEGRITY
                0: SE_GROUP_INTEGRITY_ENABLED
                0: SE_GROUP_RESOURCE
            0x00: SE_GROUP_LOGON_ID         (0)
rids: struct samr_RidWithAttribute
        rid                     : 0x0000045a (1114)
        attributes              : 0x00000007 (7)
                1: SE_GROUP_MANDATORY
                1: SE_GROUP_ENABLED_BY_DEFAULT
                1: SE_GROUP_ENABLED
                0: SE_GROUP_OWNER
                0: SE_GROUP_USE_FOR_DENY_ONLY
                0: SE_GROUP_INTEGRITY
```

info3

```
user_flags               : 0x00000220 (544)
       0: NETLOGON_GUEST
       0: NETLOGON_NOENCRYPTION
       0: NETLOGON_CACHED_ACCOUNT
       0: NETLOGON_USED_LM_PASSWORD
       1: NETLOGON_EXTRA_SIDS
       0: NETLOGON_SUBAUTH_SESSION_KEY
       0: NETLOGON_SERVER_TRUST_ACCOUNT
       0: NETLOGON_NTLMV2_ENABLED
       1: NETLOGON_RESOURCE_GROUPS
       0: NETLOGON_PROFILE_PATH_RETURNED
       0: NETLOGON_GRACE_LOGON
key: struct netr_UserSessionKey
    key: ARRAY(16): <REDACTED SECRET VALUES>
logon_server: struct lsa_StringLarge
    length                   : 0x0012 (18)
    size                     : 0x0014 (20)
    string                   : *
        string                   : 'GDW2K25DC'
```

info3

```
logon_domain: struct lsa_StringLarge
    length                   : 0x0010 (16)
    size                     : 0x0012 (18)
    string                   : *
        string                   : 'W2K25DOM'
domain_sid               : *
    domain_sid               : S-1-5-21-3193548755-1705638237-2474812355
LMSessKey: struct netr_LMSessionKey
    key: ARRAY(8): <REDACTED SECRET VALUES>
acct_flags               : 0x00000210 (528)
       0: ACB_DISABLED
       0: ACB_HOMDIRREQ
       0: ACB_PWNOTREQ
       0: ACB_TEMPDUP
       1: ACB_NORMAL
       0: ACB_MNS
       0: ACB_DOMTRUST
       0: ACB_WSTRUST
       0: ACB_SVRTRUST
```

info3

```
            0: ACB_SVRTRUST
            1: ACB_PWNOEXP
            0: ACB_AUTOLOCK
            0: ACB_ENC_TXT_PWD_ALLOWED
            0: ACB_SMARTCARD_REQUIRED
            0: ACB_TRUSTED_FOR_DELEGATION
            0: ACB_NOT_DELEGATED
            0: ACB_USE_DES_KEY_ONLY
            0: ACB_DONT_REQUIRE_PREAUTH
            0: ACB_PW_EXPIRED
            0: ACB_TRUSTED_TO_AUTHENTICATE_FOR_DELEGATION
            0: ACB_NO_AUTH_DATA_REQD
            0: ACB_PARTIAL_SECRETS_ACCOUNT
            0: ACB_USE_AES_KEYS
sub_auth_status           : 0x00000000 (0)
last_successful_logon     : NTTIME(0)
last_failed_logon         : NTTIME(0)
failed_logon_count        : 0x00000000 (0)
reserved                  : 0x00000000 (0)
```

info3

```
sidcount                  : 0x00000002 (2)
sids                      : *
    sids: ARRAY(2)
        sids: struct netr_SidAttr
            sid                       : *
                sid                       : S-1-18-2
            attributes                : 0x00000007 (7)
                   1: SE_GROUP_MANDATORY
                   1: SE_GROUP_ENABLED_BY_DEFAULT
                   1: SE_GROUP_ENABLED
                   0: SE_GROUP_OWNER
                   0: SE_GROUP_USE_FOR_DENY_ONLY
                   0: SE_GROUP_INTEGRITY
                   0: SE_GROUP_INTEGRITY_ENABLED
                   0: SE_GROUP_RESOURCE
               0x00: SE_GROUP_LOGON_ID         (0)
```

info3

```
sids: struct netr_SidAttr
    sid                      : *
        sid                      : S-1-5-21-3193548755-1705638237-2474812355-1115
    attributes               : 0x20000007 (536870919)
           1: SE_GROUP_MANDATORY
           1: SE_GROUP_ENABLED_BY_DEFAULT
           1: SE_GROUP_ENABLED
           0: SE_GROUP_OWNER
           0: SE_GROUP_USE_FOR_DENY_ONLY
           0: SE_GROUP_INTEGRITY
           0: SE_GROUP_INTEGRITY_ENABLED
           1: SE_GROUP_RESOURCE
        0x00: SE_GROUP_LOGON_ID         (0)
```

The netsamlogoncache

- Whenever user authenticates successfully we store info3 authorization data in netsamlogoncache.tdb
- samlogoncache entries are used extensively by winbind for various queries
- Cli frontend:
  - `net cache samlogon`
  - Commands for list, show, ndrdump, delete
- Problems:
  - **Expiry:** Entries in that tdb do not expire, they're valid forever
  - **Consistency:** Database is not clustered, each node creates an independent cache
  - Entries are only overwritten when new data gets available
    - problematic for applications that cannot get samlogoncache entries to be written
  - MR: Allow netsamlogoncache.tdb to expire cache entries

    https://gitlab.com/samba-team/samba/-/merge_requests/4399

Winbind LDAP lookups

- Winbind will always consult netsamlogoncache first
- When no entry is in the cache winbind attempts to lookup group membership itself via LDAP using three different mechanisms:
  - tokenGroups
  - memberOf
  - member

Winbind LDAP lookups - tokenGroups

- Scope: `LDAP_SCOPE_BASE`
- Base DN: `CN=gd,CN=Users,DC=w2k25dom,DC=ber,DC=redhat,DC=com`
- Filter: `(objectclass=*)`
- Attributes: `tokenGroups`
- Example cli:
  - `net ads sid S-1-5-21-3193548755-1705638237-2474812355-1104 -P tokengroups`

Winbind LDAP lookups - memberOf

- Scope: `LDAP_SCOPE_BASE`
- Base DN: `CN=gd,CN=Users,DC=w2k25dom,DC=ber,DC=redhat,DC=com`
- Filter: `(objectclass=*)`
- Attributes: `memberof;Range=0-*`
- Query makes use of ranged attribute ranged LDAP searches
- Example cli:
  - `net ads dn CN=gd,CN=Users,DC=w2k25dom,DC=ber,DC=redhat,DC=com --extended-dn 'memberof;Range=0-*' -P`

Winbind LDAP lookups - member

- Scope: `LDAP_SCOPE_SUBTREE`
- Base DN: `CN=Users,DC=w2k25dom,DC=ber,DC=redhat,DC=com`
- Filter:

  `(&(member=CN=gd,CN=Users,DC=w2k25dom,DC=ber,DC=redhat,DC=com)(objectCategory=group)(groupType:dn:1.2.840.113556.1.4.803:=2147483648))`
- Attributes: `objectSid`
- Use of extended dn LDAP control to get distinguished names including SID and GUID attributes
- Example cli:
  - `net ads search "(&(member=CN=gd,CN=Users,DC=w2k25dom,DC=ber,DC=redhat,DC=com)(objectCategory=group)(groupType:dn:1.2.840.113556.1.4.803:=2147483648))" -P objectSid`

Winbind LDAP lookups - and?

- All these LDAP lookups do not reliably deliver the accurate list of group membership SIDs
  - Permissions
  - Nested memberships
  - Etc.
- Even if they would, we cannot compose a full info3 structure out of these LDAP results
- **=> Much better option is not rely on LDAP but to use Krb5 S4U2SELF**

S4U2SELF Kerberos Extension

- **“The S4U2self extension allows a service to obtain a service ticket to itself on behalf of a user” [MS-SFU]**
- **Service/itself:**
  - Winbind, authenticated using machine account name and machine account password
- **User:**
  - User account to retrieve authorization information for
- Krb5 control flow:
  - AS-REQ -> AS-REP
  - TGS-REQ
    - PA-FOR-USER / PA-S4U-X509-USER
  - TGS-REP
    - PAC-LOGON-INFO

S4U2SELF krb5 API

- Both Kerberos libraries support S4U2SELF APIs
- krb5.h:
  - **Heimdal:**
    - `krb5_get_creds_opt_set_impersonate()`
  - **MIT:**
    - `krb5_get_credentials_for_user()`
- gssapi_ext.h:
  - `gss_acquire_cred_impersonate_name()/gss_add_cred_impersonate_name()`
- MIT asynchronous S4U client (Alexander Bokovoy):
  - https://github.com/krb5/krb5/pull/1500

S4U2SELF krb5 cli

- Samba provides S4U2SELF impersonation cli since many years:
- `net ads kerberos pac dump impersonate=$UPN -P`
  - only displays PAC info3, does not store information
- userPrincipalName always required for s4u2self operations

The user principal name

- Optional `userPrincipalName` LDAP attribute
- When not set:
  - $sAMAccountname@$REALM
  - gd@W2K25DOM.BER.REDHAT.COM
- Role of upnSuffixes
- Any arbitrary suffix might be chosen by AD admins
  - COMPLETLY.RANDOM.EXAMPLE.COM
- gd@COMPLETLY.RANDOM.EXAMPLE.COM

DRSUAPI Cracknames

- DRSUAPI DCE/RPC interface
- Only accessibly via ncacn_ip_tcp: transport and privacy
- DsCrackNames interface allows lookup of all sorts
  - Input:
    - `DRSUAPI_DS_NAME_FORMAT_SID_OR_SID_HISTORY`
  - Output:
    - `DRSUAPI_DS_NAME_FORMAT_USER_PRINCIPAL`
- `Add DRSUAPI client to winbind`
- `wbinfo -sid-to-upn <SID>`

Potentially blocking krb5 calls

- Kerberos libraries behavior w.r.t. DNS lookup and networking cannot be controlled by Samba
- Move potentially blocking krb5 calls to new separate winbind child process (log.winbind-kerberos)
- New winbind call for s4u2self impersionation:

```
NTSTATUS wbint_KerberosImpersonationToken(
        [in,string,charset(UTF8)] char *impersonate_principal,
        [out,ref] wbint_SidArray *sids,
        [out,unique] netr_SamInfo3 **info3
);
```

Demo

Current limitations & next steps

- **Limitations:**
  - Only supports single realm scenarios

    (requester and impersonation account reside in the same domain)
  - UPN suffixes not fully supported

- **Next steps:**
  - Fully support upn suffixes
  - Convince metze!
  - Adopt and test Alexanders new async API
  - Cleanup MR and get review

References

Samba Upstream Contributions:

- Draft: Add Kerberos impersonation support to Winbindd to retrieve user group membership
  https://gitlab.com/samba-team/samba/-/merge_requests/4318

MIT Kerberos Contributions:

- MIT Kerberos Consortium: Projects/Services4User
  https://k5wiki.kerberos.org/wiki/Projects/Services4User
- Asynchronous S4U interfaces PR from Alexander Bokovoy
- https://github.com/krb5/krb5/pull/1500

Documentation:

- [MS-SFU]: Kerberos Protocol Extensions: Service for User and Constrained Delegation Protocol
  https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-sfu/3bff5864-8135-400e-bdd9-33b552051d94
- [MS-PAC]: Privilege Attribute Certificate Data Structure
  https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-pac/166d8064-c863-41e1-9c23-edaaa5f36962
- [MS-KILE]: Kerberos Protocol Extensions
  https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-kile/2a32282e-dd48-4ad9-a542-609804b02cc9
- [MS-DRSR]: Directory Replication Service (DRS) Remote Protocol
  https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-drsr/f977faaa-673e-4f66-b9bf-48c640241d47

Q&A

Questions?