Samba Tutorial

Günther Deschner
gd@samba.org

(Red Hat / Samba Team)

Agenda Day 1 – new ways of joining

- Joining from scratch
  - libsmbconf
  - libnetjoin
- Joining remotely
  - Windows NetJoinDomain call
  - `libnetapi.so`
  - Samba3 server support
- Joining using GUI
  - Domainjoin gtk GUI

Agenda Day 2 – netapi and Group Policy

- Programmatic access to the account database
  - User account management functions
  - Plan: User manager GUI
- Group Policy client
  - Basic Intro
  - Samba – a group policy engine
  - Currently supported options

New ways of joining

Before Samba 3.2

- `net rpc join` and `net ads join` implement all join code in the “net” binary monolithically
- Configuration must always have been set appropriately before the join
- 3rd party applications could just do upcalls to the “net” binary, and in case of error, parse stdout return
- We wanted to change this and provide something better

New in Samba 3.2

- Bugfixes (joining Windows 2008)
- Internal library libnetjoin
- Shared library offering a “NetJoinDomain” call
- Ability to join with an empty configuration file
- Ability to join remotely
- Ability to be joined remotely
- Example gtk GUI for joining

Joining with an (almost) empty smb.conf

- Samba 3.2 has a new `libsmbconf` internal interface
- Provides read/write access for storing Samba configuration in the local samba registry
- Frontend Samba: `net conf`
- Frontend Windows: `regedit.exe`
- Based on this `libsmbconf`, libnetjoin can join a client with a minimal smb.conf file:

```
[global]
        config backend = registry
```

Live Demo

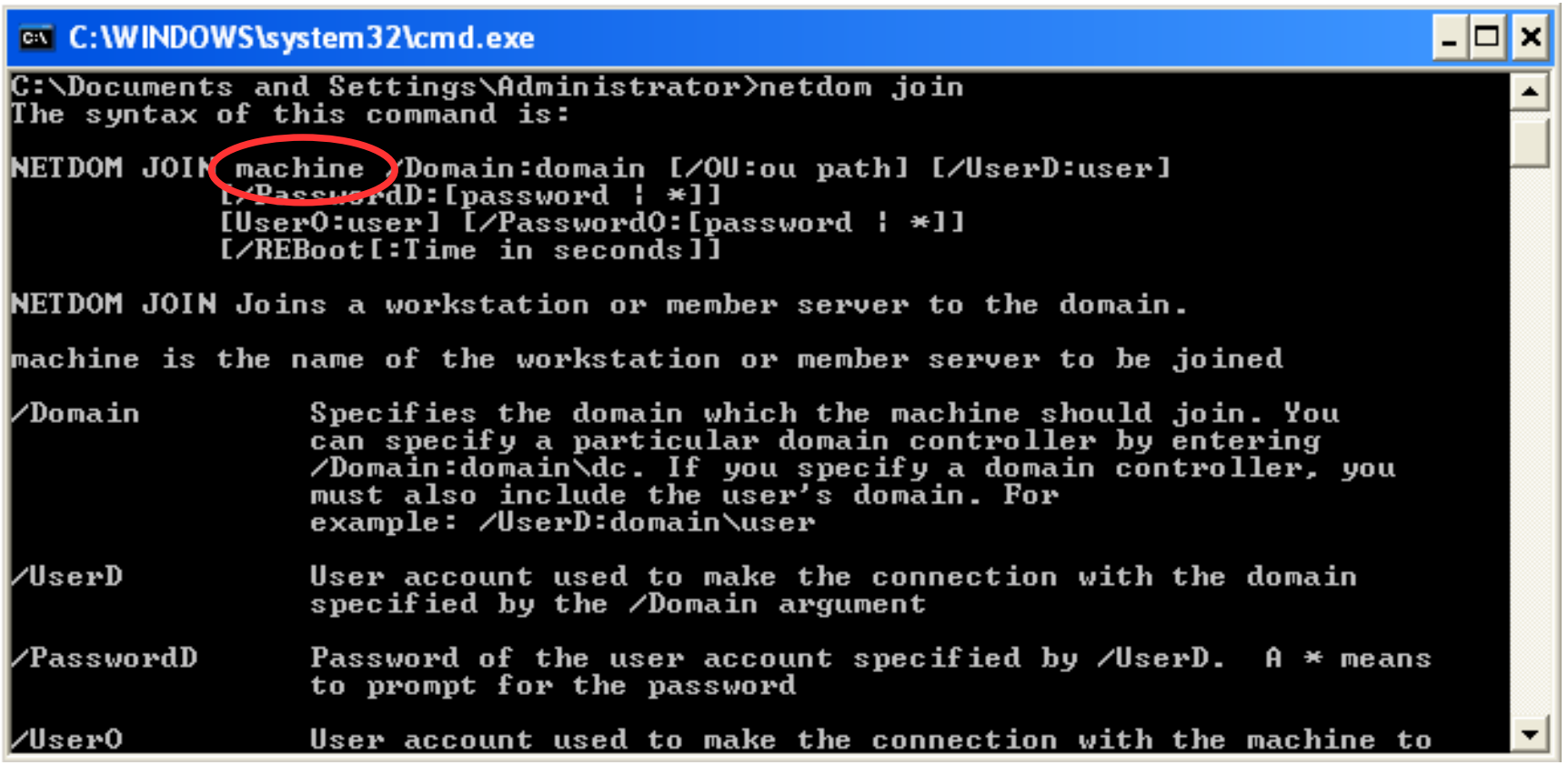
net ads join
&
“config backend = registry”

Joining and Unjoining on Windows ?

- Joining using the gui
- Joining using the command line “netdom join”

The Windows netdom join tool

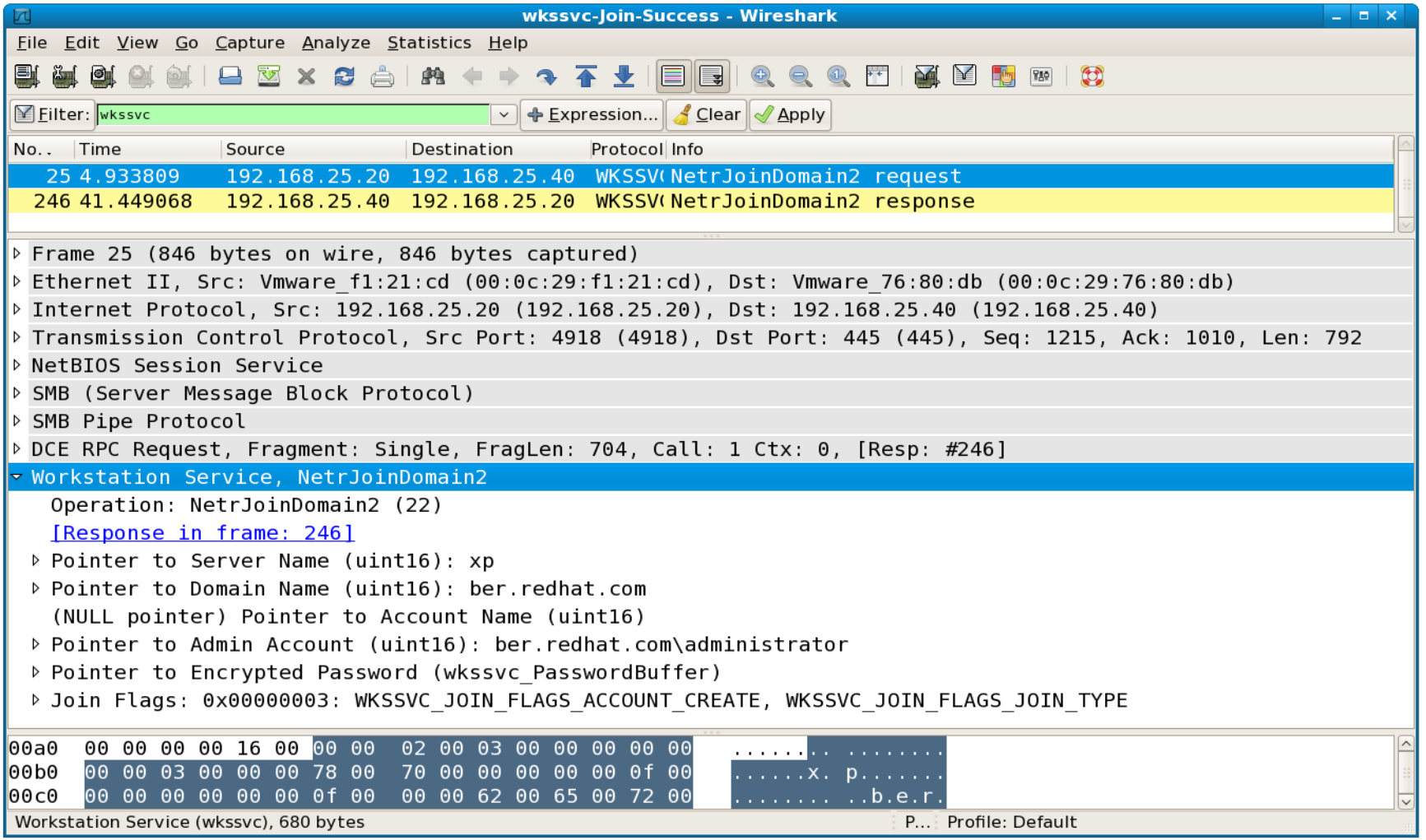
- Windows can join the local or a ***remote*** host to a domain



The Windows netdom join tool

- Connects to remote computer using current or given credentials
- Opens the remote Workstation Service Named Pipe (WKSSVC)
- Calls NetrJoinDomain2 which transports another set of domain credentials which are used by the remote computer to join the domain
- Also allows to specify Account OU to be joined to
- Once the remote computer receives this request, it gets very active

“netdom join” calls NetrJoinDomain2



A Cryptographical Challenge

- Domain account credentials were encrypted in an unknown, undocumented format
- Analyzed buffer using Samba4 smbtorture
- After some weeks of experiments, smbtorture was able to join XP into Windows 2003 remotely
- Needed to find a good home for this feature long-term
- On Windows, NetApi32.lib provides a NetJoinDomain call
- Plan: Start a `libnetapi.so` library to support this call on Linux
- All crypto is done in the library, caller has a simple API

NetJoinDomain

- Defined in netapi.h
- Does anything required to join the local or a remote computer into a domain
- Admin credentials are given in clear, NetJoinDomain takes care of encryption details
- Header:

```
NET_API_STATUS NetJoinDomain(const char * server /* [in] */,
                             const char * domain /* [in] [ref] */,
                             const char * account_ou /* [in] */,
                             const char * account /* [in] */,
                             const char * password /* [in] */,
                             uint32_t join_flags /* [in] */);
```

NetJoinDomain remote, local: netdomjoin

- **`netdomjoin` in lib/netapi/examples directory**

```
Usage: netdomjoin hostname
     --ou=ACCOUNT_OU                 Account ou
     --domain=DOMAIN                 Domain name (required)
     --userd=USERNAME                Domain admin account
     --passwordd=PASSWORD            Domain admin password

Help options:
     -?, --help                      Show this help message
     --usage                         Display brief usage message

Common samba netapi example options:
     -U, --user=USERNAME             Username used for connection
     -p, --password=PASSWORD         Password used for connection
     -d, --debuglevel=DEBUGLEVEL     Debuglevel
     -k, --kerberos                  Use Kerberos
```

NetJoinDomain remote, local: net

- **Sub command of `net`: `net dom join`**

```
usage: net dom join <domain=DOMAIN> <ou=OU> <account=ACCOUNT>
<password=PASSWORD> <reboot>
```

NetJoinDomain local uses libnetjoin

- Needed to abstract all join calls of “net ads join” to make them available outside of the net binary
- Started internal library libnetjoin
- NetJoinDomain calls libnetjoin when server_name is NULL.
- libnetjoin uses IDL for the join and unjoin context

Libnetjoin - input

```
libnet_JoinCtx: struct libnet_JoinCtx
        in: struct libnet_JoinCtx
             dc_name                  : 'w2k3dc-rhber'
             machine_name             : 'MTHELENA'
             domain_name              : *
                  domain_name             : 'BER.REDHAT.COM'
             account_ou               : NULL
             admin_account            : 'administrator'
             admin_password           : 'password'
             machine_password         : NULL
             join_flags               : 0x00000023 (35)
                  0: WKSSVC_JOIN_FLAGS_JOIN_WITH_NEW_NAME
                  0: WKSSVC_JOIN_FLAGS_JOIN_DC_ACCOUNT
                  0: WKSSVC_JOIN_FLAGS_DEFER_SPN
                  0: WKSSVC_JOIN_FLAGS_MACHINE_PWD_PASSED
                  0: WKSSVC_JOIN_FLAGS_JOIN_UNSECURE
                  1: WKSSVC_JOIN_FLAGS_DOMAIN_JOIN_IF_JOINED
                  0: WKSSVC_JOIN_FLAGS_WIN9X_UPGRADE
                  0: WKSSVC_JOIN_FLAGS_ACCOUNT_DELETE
                  1: WKSSVC_JOIN_FLAGS_ACCOUNT_CREATE
                  1: WKSSVC_JOIN_FLAGS_JOIN_TYPE
             os_version               : NULL
             os_name                  : NULL
             create_upn               : 0x00 (0)
             upn                      : NULL
             modify_config            : 0x00 (0)
             ads                      : NULL
             debug                    : 0x01 (1)
             secure_channel_type      : SEC_CHAN_WKSTA (2)
```

Libnetjoin - output

```
libnet_JoinCtx: struct libnet_JoinCtx
        out: struct libnet_JoinCtx
             account_name             : NULL
             netbios_domain_name      : 'BER'
             dns_domain_name          : 'ber.redhat.com'
             dn                       :
                  'CN=mthelena,CN=Computers,DC=ber,DC=redhat,DC=com'
             domain_sid               : *
                  domain_sid              : S-1-5-21-1800104011-1129049609-1243822444
             modified_config          : 0x00 (0)
             error_string             : NULL
             domain_is_ad             : 0x01 (1)
             result                   : WERR_OK
```

NetrJoinDomain2 server support

- Samba 3.2 has also initial support for remote join and unjoin server side
- NetrJoinDomain2 server uses libnetjoin
- Admin credentials are needed
- Only members of the Domain Admin and Local Administrators group can call this
- Probably needs more post-processing
- Needs be transaction based

Live Demo

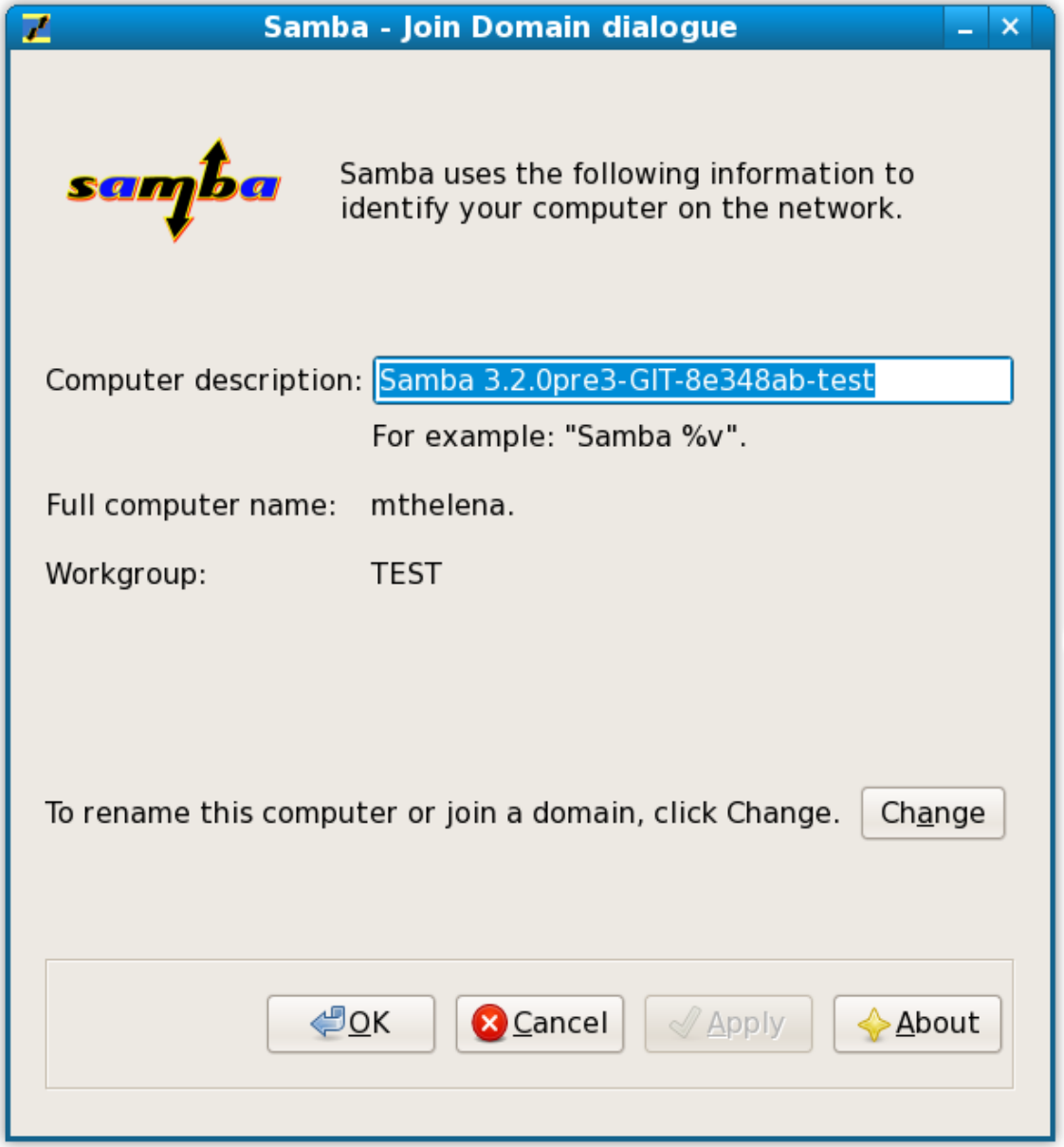
Samba joins XP to W2K3
XP joins Samba to W2K3

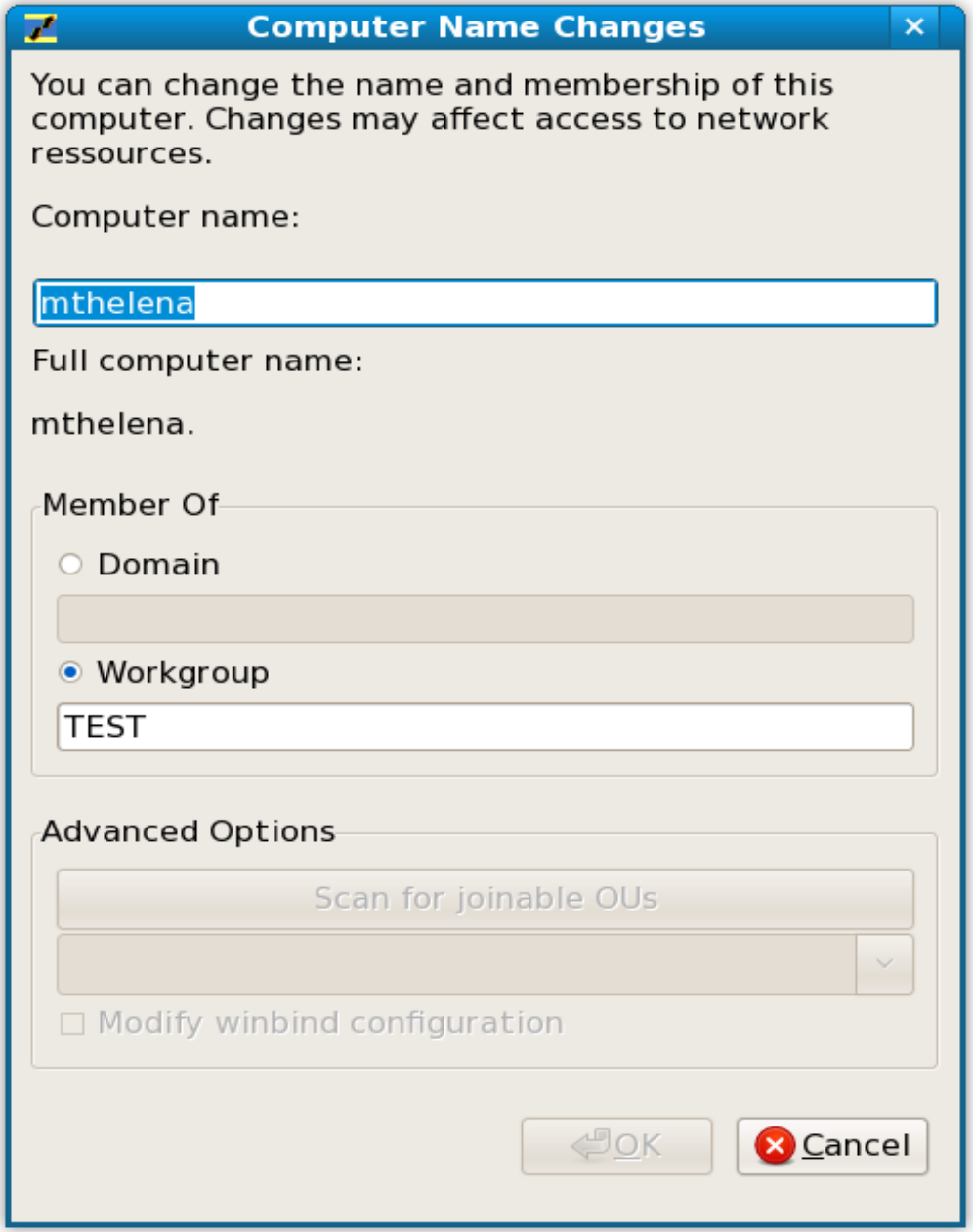
Grow your own...

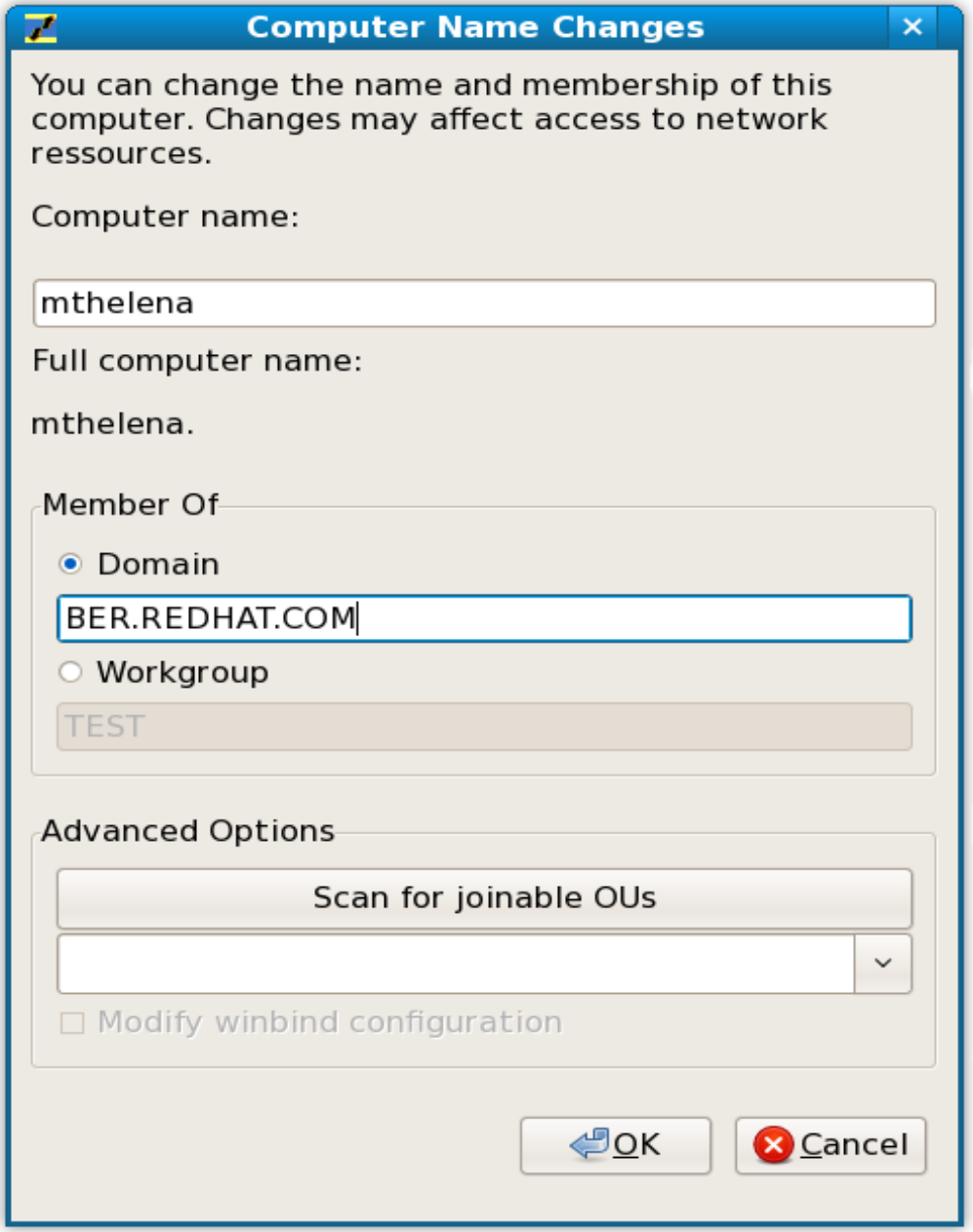
- We have abstracted the local join in libnetjoin
- We have abstracted the join in libnetapi
- Now new applications can be written by anyone to join computers into domains
- One proof of concept experiment was to offer a Windows-like interface for joining a workstation into a domain
- gtk domainjoin gui in `lib/netapi/examples`:
  - `netdomjoin-gui`

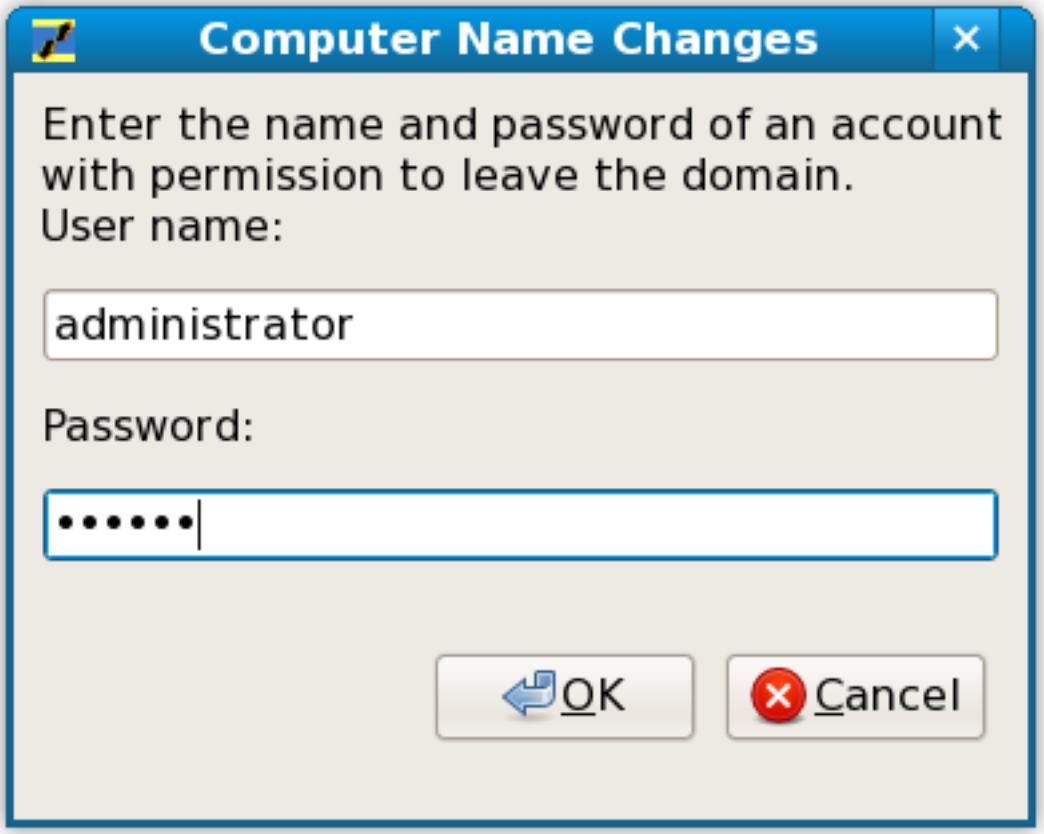
Live Demo

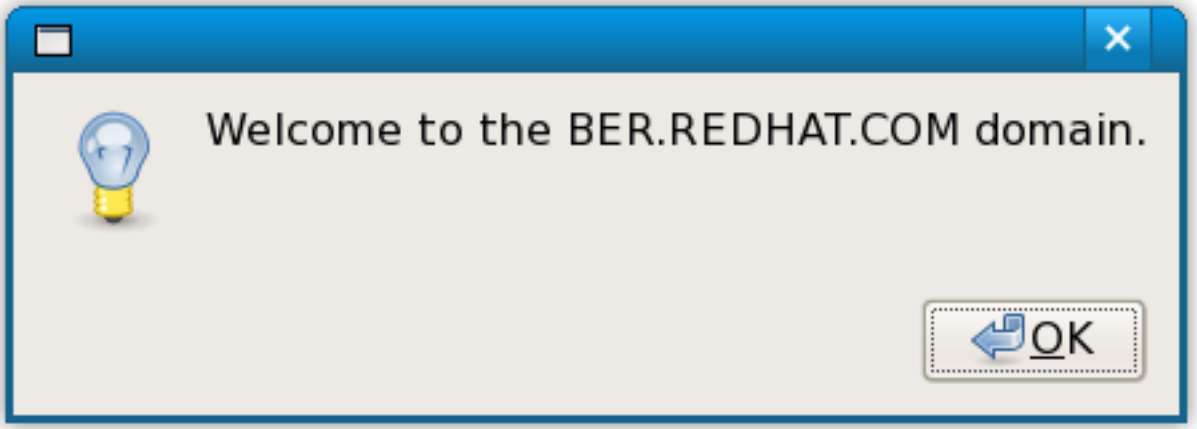
netdomjoin-gui

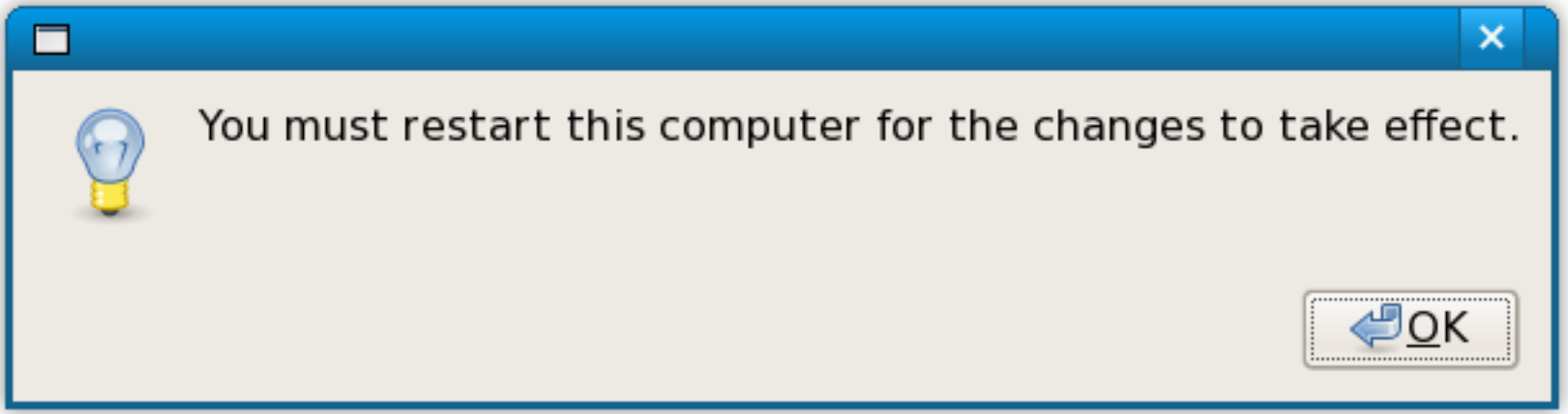

Samba - Join Domain dialogue
Samba uses the following information to identify your computer on the network.
Computer description:
Samba 3.2.0pre3-GIT-8e348ab-test
For example: "Samba %v".
Full computer name:
mthelena.
Workgroup:
TEST
To rename this computer or join a domain, click Change.
Change
OK
Cancel
Apply
About


Computer Name Changes
You can change the name and membership of this computer. Changes may affect access to network ressources.
Computer name:
mthelena
Full computer name:
mthelena.
Member Of
Domain
Workgroup
TEST
Advanced Options
Scan for joinable OUs
Modify winbind configuration
OK
Cancel


Computer Name Changes
You can change the name and membership of this computer. Changes may affect access to network ressources.
Computer name:
mthelena
Full computer name:
mthelena.
Member Of
Domain
BER.REDHAT.COM
Workgroup
TEST
Advanced Options
Scan for joinable OUs
Modify winbind configuration
OK
Cancel


Computer Name Changes
Enter the name and password of an account with permission to leave the domain.
User name:
administrator
Password:
OK
Cancel


Welcome to the BER.REDHAT.COM domain.
OK


You must restart this computer for the changes to take effect.
OK

What's next:

- libnetjoin:
  - joining using vendor extensions (plugin/script that vendors use to integrate with local system management frameworks)
  - hostname, krb5, ntp, winbind/idmap, nsswitch, pam configuration
  - remote “unsecure” join
- libsmbconf:
  - write support for smb.conf backend in libsmbconf
- libnetapi:
  - make the library less heavyweight
  - python bindings
- start an internal libnet

Day 2

libnetapi

libnetapi

- Modeled after the windows equivalent NetApi32.lib
- Initially created only to support NetJoinDomain
- Wrapped around new Samba4 IDL based rpc client calls
- NetApi calls distinguish local or remote execution based on the first argument (usually `server_name`)
- Various examples available under `lib/netapi/examples`

Implemented calls in libnetapi

- NetJoinDomain
- NetUnjoinDomain
- NetGetJoinInformation
- NetGetJoinableOUs
- NetServerGetInfo
- NetServerSetInfo
- NetGetDCName
- NetGetAnyDCName
- DsGetDcName
- NetUserAdd
- NetUserDel
- NetUserEnum
- NetQueryDisplayInformation

Example: DsGetDCName

- Returns information about a DC for a specific domain
- Example in lib/netapi/examples/dsgetdc/dsgetdc.c

```
Usage: dsgetdc hostname domain
  Help options:
      -?, --help                            Show this help message
      --usage                               Display brief usage message

  Common samba netapi example options:
      -U, --user=USERNAME                   Username used for
  connection
      -p, --password=PASSWORD               Password used for
  connection
      -d, --debuglevel=DEBUGLEVEL           Debuglevel
      -k, --kerberos                        Use Kerberos
```

DsGetDCName examples

- **lib/netapi/examples/bin/dsgetdc w2k3dc-rhber nt4dom -U administrator**
  Password:
  domain NT4DOM has name: \\NT4-PDC
- **lib/netapi/examples/bin/dsgetdc w2k3dc-rhber ber.redhat.com -U administrator**
  Password:
  domain ber.redhat.com has name: \\w2k3dc-rhber.ber.redhat.com

Example: NetUserAdd

- Creates user accounts on the remote server, following the exactly same pattern as Windows would do
- Example in lib/netapi/examples/user/user_add.c

```
Usage: user_add hostname username password
  Help options:
      -?, --help                            Show this help message
      --usage                               Display brief usage message

  Common samba netapi example options:
      -U, --user=USERNAME                   Username used for
  connection
      -p, --password=PASSWORD               Password used for
  connection
      -d, --debuglevel=DEBUGLEVEL           Debuglevel
      -k, --kerberos                        Use Kerberos
```

Example: NetUserDel

- Deletes user accounts from remote servers, following the exactly same pattern as Windows would do
- Example in lib/netapi/examples/user/user_del.c

```
Usage: user_del hostname username
  Help options:
      -?, --help                            Show this help message
      --usage                               Display brief usage message

  Common samba netapi example options:
      -U, --user=USERNAME                   Username used for
  connection
      -p, --password=PASSWORD               Password used for
  connection
      -d, --debuglevel=DEBUGLEVEL           Debuglevel
      -k, --kerberos                        Use Kerberos
```

Future of account management

- Once we have added the missing calls, someone can start on a User Manager gui
- We still need to solve how to execute the account management calls locally
- One idea is to start a local RPC server on the fly (when running as root)

Group Policy

Why bother with Group Policy ?

- Primary motivation: `winbind` enables AD integration but has no support for Group Policy
- There are many Group Policy engines for Unix around – just none of them is open source
- Samba / `winbind` has all technology on board to do that itself
- Plan: provide a group policy framework based on Samba with an open architecture
- Main obstacle: there is no – established - centralized configuration framework on Unix
- But Samba has a registry

Why bother with Group Policy ?

- Triggered by winbindd offline feature
- `cached logon count`
- That integer value describes how many logons are cached locally.
- On Windows “0” disables the feature of caching logons locally.
- On Windows it is stored in the registry at:
  `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\CachedLogonsCount`
- In Samba this value is hard-coded in winbind
- On Windows, Group Policy modifies this registry key

Learning Group Policy

- We wanted to understand how this works
- Analyzed network traffic between XP workstation and Windows 2003 Domain Controller:
  - Kerberos calls for authentication
  - LDAP calls for receiving the list of Group Policies
  - SMB calls for downloading policy files
  - => Nothing Samba couldn't do as well
- A Group Policy Object consists of:
  - Group Policy Container (gpc)
  - Group Policy Template (gpt)

Group Policy Container (LDAP)

```
CN={00F0991D-
DCBF-4A7A-9723-3F7D3977FDB8},CN=Policies,CN=System,DC=ber,DC=redhat,DC=com
-----------------------------------------------------------------------------
cn: {00F0991D-DCBF-4A7A-9723-3F7D3977FDB8}
displayName: gd new privs
distinguishedName: CN={00F0991D-
DCBF-4A7A-9723-3F7D3977FDB8},CN=Policies,CN=System,DC=ber,DC=redhat,DC=com
flags: 0 (0x00000000)
     - GPFLAGS_ALL_DISABLED (0x00000003)
     X GPFLAGS_ALL_ENABLED (0x00000000)
     - GPFLAGS_MACHINE_SETTINGS_DISABLED (0x00000002)
     - GPFLAGS_USER_SETTINGS_DISABLED (0x00000001)
gPCFileSysPath: \\ber.redhat.com\SysVol\ber.redhat.com\Policies\{00F0991D-
DCBF-4A7A-9723-3F7D3977FDB8}
gPCFunctionalityVersion: 2
gPCMachineExtensionNames: [{827D319E-6EAC-11D2-4EA-00C04F79F83A}{803E14A0-B4FB-11D0-
A0D0-00A0C90F574B}]
name: {00F0991D-DCBF-4A7A-9723-3F7D3977FDB8}
objectCategory: CN=Group-Policy-
Container,CN=Schema,CN=Configuration,DC=ber,DC=redhat,DC=com
objectClass: container
objectClass: groupPolicyContainer
objectClass: top
objectGUID: a3d552f1-491f-4219-8aa9-85c71a9505d5
showInAdvancedViewOnly: TRUE
versionNumber: 11
whenChanged: Mon Apr 14 00:45:21 2008 (20080413224521.0Z)
whenCreated: Sun Apr 13 14:34:38 2008 (20080413123438.0Z)
```

Group Policy Link on Domain (LDAP)

- **DC=ber,DC=redhat,DC=com**

```
----------------------------------------------------------------------------
creationTime: Thu Jun  7 17:15:31 2007 (128257029314062500)
dc: ber
distinguishedName: DC=ber,DC=redhat,DC=com
gPLink: [LDAP://cn={00F0991D-
DCBF-4A7A-9723-3F7D3977FDB8},cn=policies,cn=system,DC=ber,DC=redhat,DC=com;0]
[LDAP://cn={537D6625-EE33-4252-BB40-
E8AF8C48152F},cn=policies,cn=system,DC=ber,DC=redhat,DC=com;0]
[LDAP://CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=ber,DC=re
dhat,DC=com;0][LDAP://cn={E5C3D7F3-9392-4443-
AC45-4EBDA1D8219B},cn=policies,cn=system,DC=ber,DC=redhat,DC=com;0]
```

Group Policy Template (CIFS)

- **Replicated SYSVOL share on the Windows DC Fileserver**
- `ber.redhat.com/Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}`
- ```
  ./USER
  ./USER/Microsoft
  ./USER/Microsoft/RemoteInstall
  ./USER/Microsoft/RemoteInstall/oscfilter.ini
  ./GPT.INI
  ./Adm
  ./Adm/wuau.adm
  ./Adm/inetres.adm
  ./Adm/system.adm
  ./Adm/wmplayer.adm
  ./Adm/conf.adm
  ./Adm/admfiles.ini
  ./MACHINE
  ./MACHINE/Scripts
  ./MACHINE/Scripts/Startup
  ./MACHINE/Scripts/Shutdown
  ./MACHINE/Microsoft
  ./MACHINE/Microsoft/Windows NT
  ./MACHINE/Microsoft/Windows NT/SecEdit
  ./MACHINE/Microsoft/Windows NT/SecEdit/GptTmpl.inf
  ./MACHINE/Registry.pol
  ```
  ```

Windows Group Policy Client

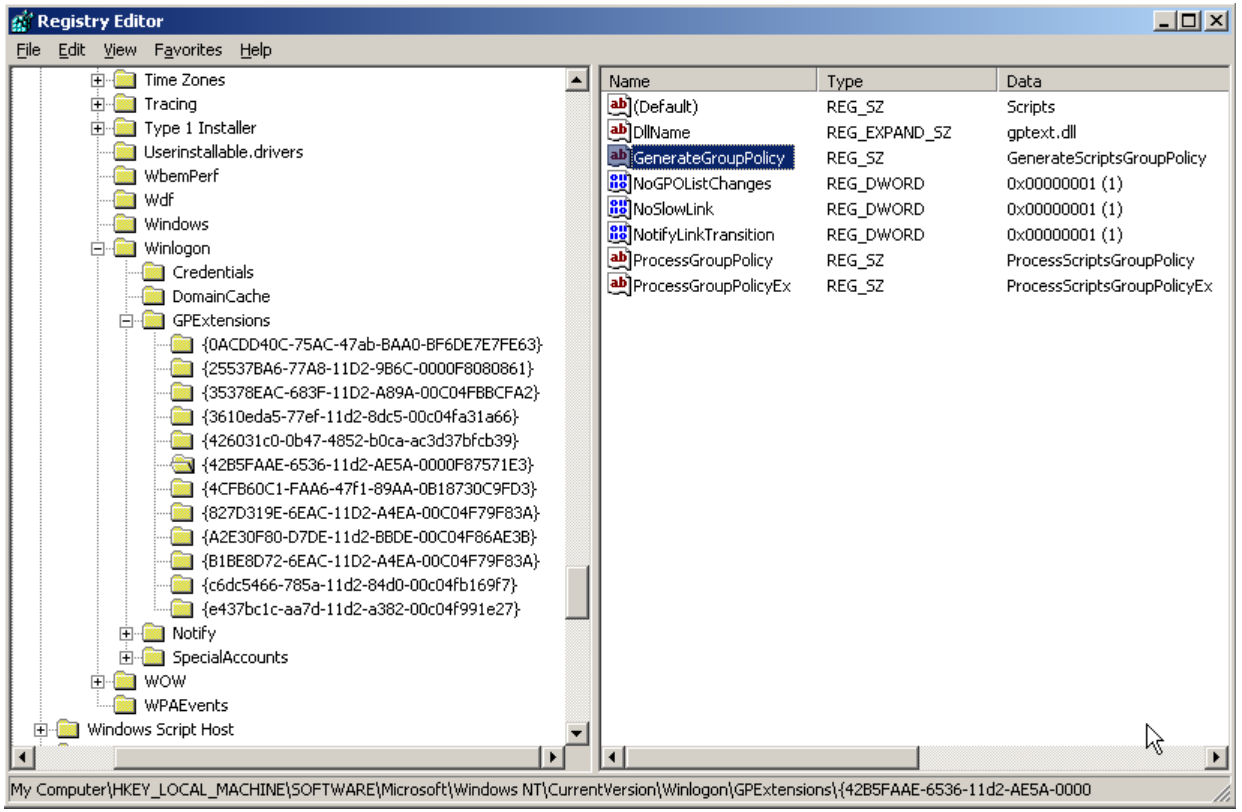
- Winlogon process does call the Group Policy client
- Group Policy client authenticates to LDAP, retrieves list of Group Policy objects
- Client loads Client Side Extensions to process the Group Policy Object list

Windows Group Policy CSEs

- Group Policy assigns specific sets of policy to Client Side Extensions (CSEs)
- There are CSEs for:
  - Registry settings
  - Disc Quota
  - Folder Redirection
  - Internet Explorer Branding
  - Scripts
  - Security
  - Wireless Group Policy
  - and many, many more.

Windows Group Policy CSEs

- Client side extensions on a Windows 2003 Domain controller



How to implement GPOs in Samba?

- We can identify policy and download it
- How can we process it?
- On Windows almost anything is stored in the registry
- Monitored Windows XP registry changes after manipulating Group Policies on the Domain Controller
- Analyzed processing information which is stored in the registry
- Started libgpo internal library to identify list of Group Policies for a workstation/user

Samba Group Policy components

- `libgpo/`
  - `ads_get_gpo_list()` => **looks into LDAP, stores registry**
  - `check_refresh_gpo_list()` => **connects sysvol, gets policy**
  - `gpo_process_gpo_list()` => **calls extensions**
- `net ads gpo` **commands**
  - `net ads gpo refresh`

    Does retrieve the list of GPOs in LDAP, and downloads policy from SYSVOL
  - `net ads gpo apply`

    Does call the client side extensions and processes the Group Policy list

Samba Group Policy components

- `winbind` async group policy child (with `smbcontrol` signaling, `winbind` queries)
  - `winbind` is processing Group Policy for the machine account
  - `winbind` is also processing user-based Group Policy when using `pam_winbind`
- Client side extensions modular API: `gpext`

Samba Client side extensions: gpext

- Stored below `/usr/lib{64}/samba/gpext`
- Will automatically register by GUID in local Samba registry below:

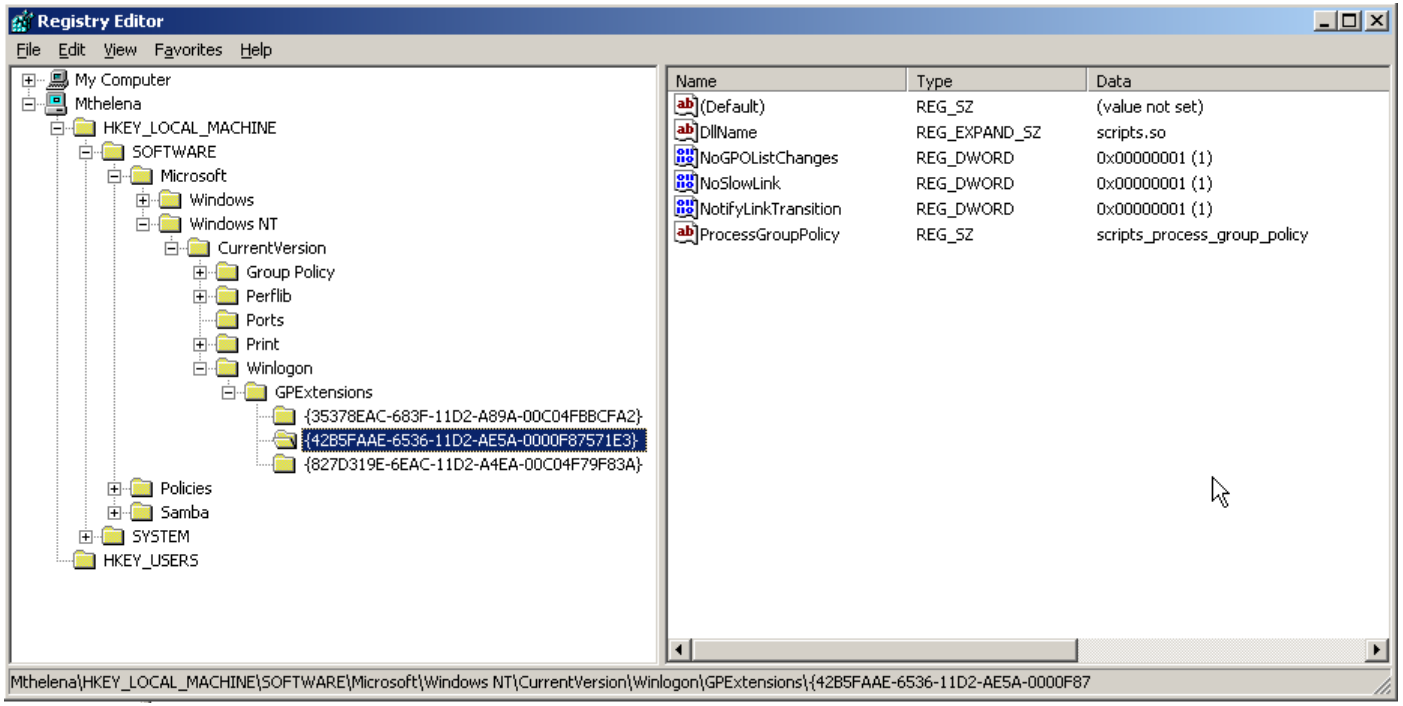
  `HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\GPExtensions`

- CSE module configuration stored in registry itself:
  - DllName, ProcessGroupPolicy, NoMachinePolicy, NoUserPolicy, NoSlowLink, NoBackgroundPolicy, NoGPOListChanges, PerUserLocalSettings, RequiresSuccessfulRegistry, EnableAsynchronousProcessing, ExtensionDebugLevel

Client side extensions: Existing modules

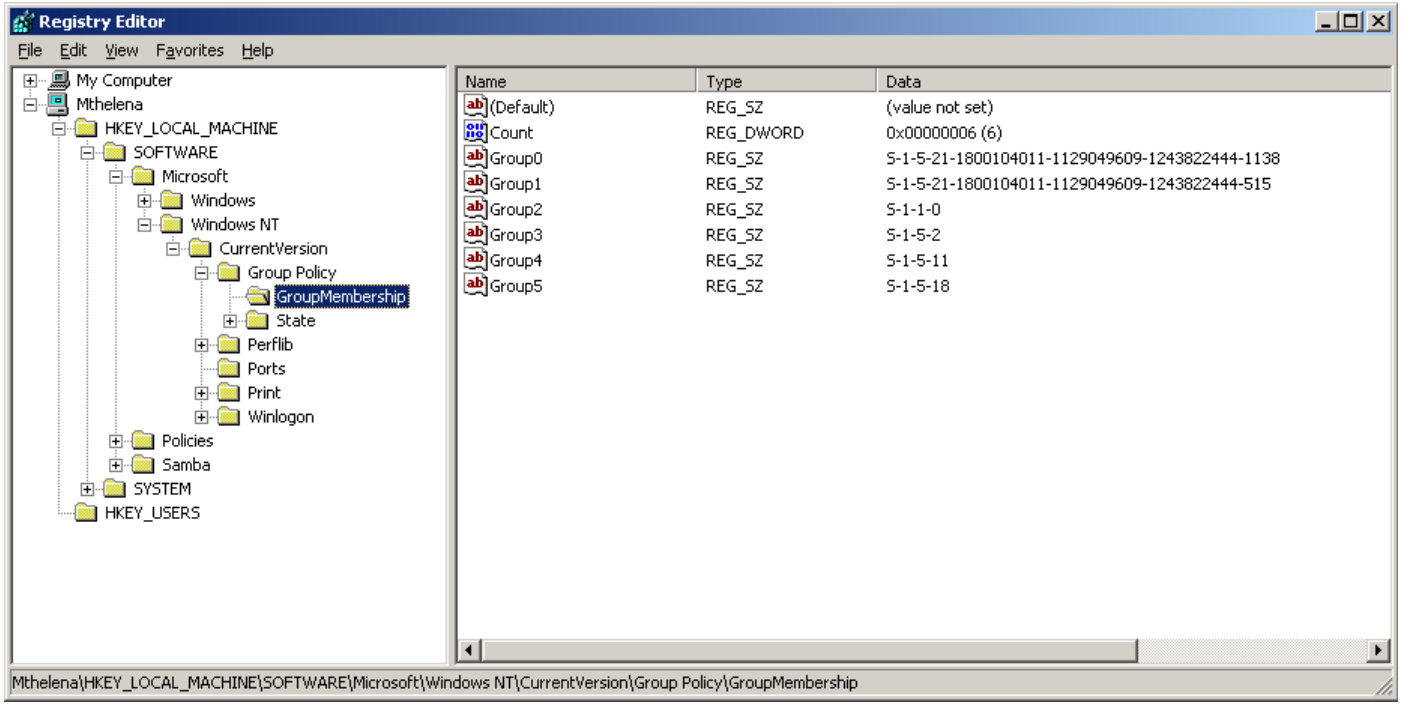
- `registry.so {35378EAC-683F-11D2-A89A-00C04FBBCFA2}`
  - parses PReg files (`Registry.pol`)
  - applies them to local Samba registry
- `security.so {827D319E-6EAC-11D2-A4EA-00C04F79F83A}`
  - parses ini files (`GptTmpl.inf`)
  - applies them to the local Samba registry (under `HKLM\SOFTWARE\Samba\Group Policy`)
- `scripts.so {42B5FAAE-6536-11D2-AE5A-0000F87571E3}`
  - parses ini files (`scripts.ini`)
  - applies the script info to the local Samba registry

Samba registry: GPExtensions



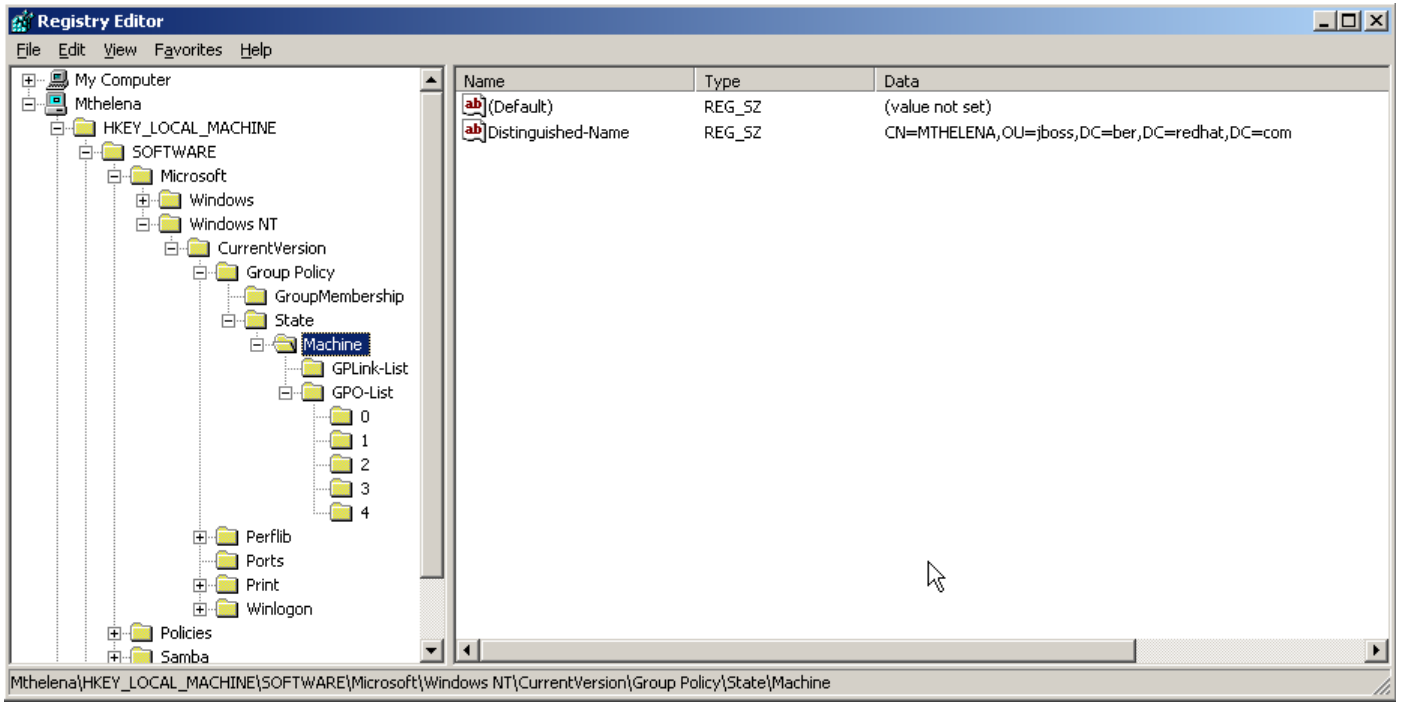
- **The three implemented Samba Group Policy Extensions viewed via regedit**

Samba registry: Group Policy State



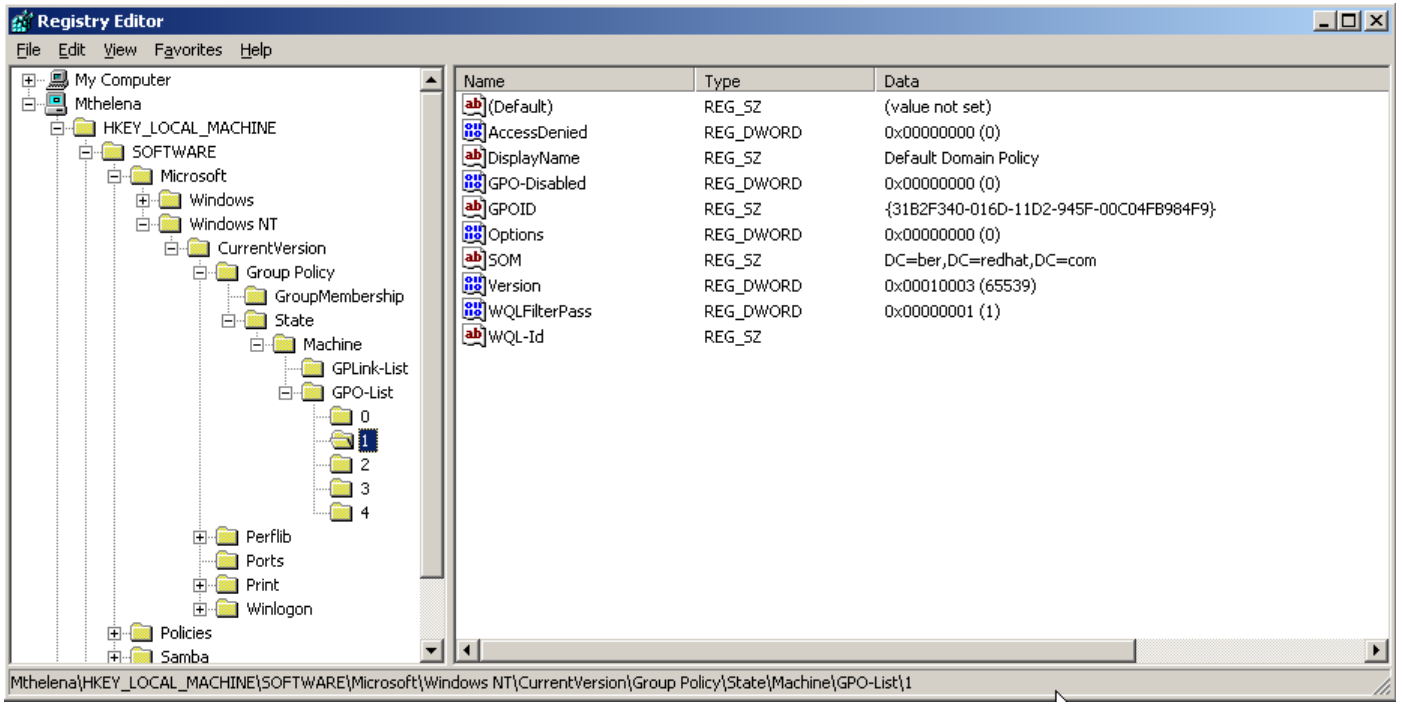
- **Group Policy stores current GroupMembership in order to identify Policy changes based on membership**

Samba registry: Group Policy State



- **Samba also stores the Distinguished Name in the State registry object**

Samba registry: Group Policy State



- **Finally, the current list of Group Policies is stored in an ordered list in the registry**

Samba Group Policy support

- Certainly not all of Group Policy can be applied and enforced
- Samba will support everything it needs and can support
- Security Settings
  - Security Options
  - Rights
- And possibly:
  - Logon/Logoff Scripts
  - Auditing Settings

Group Policy examples:

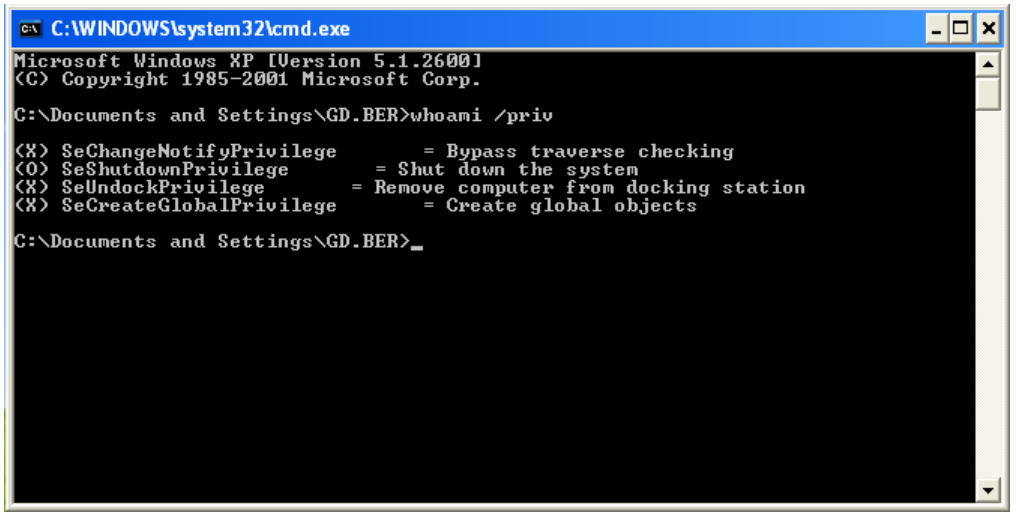
- Deploy granting of user rights to one user
- Setting SMB server signing policy to one machine
- Using Administrative Templates for Unix Settings

Distributing Privileges with Group Policy

- User privileges always have a local scope
- In Samba they are stored in account_pol.tdb
- In Active Directory, user privileges are deployed using Group Policy
- Example: Grant `SeTakeOwnershipPrivilege` to BER\gd

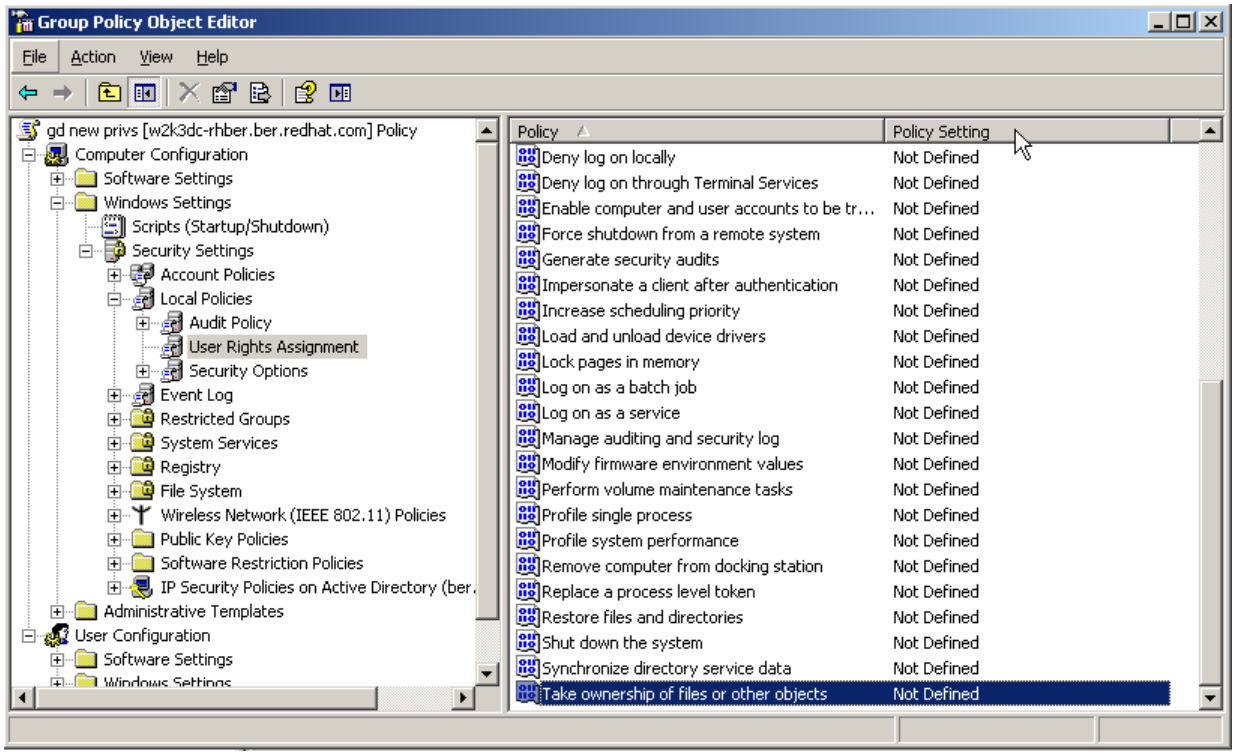
Distributing Privileges with Group Policy

- Logging into Windows XP as BER\gd
- `whoami /priv` does show a total of 4 Privileges
- Take Ownership Privilege is not included



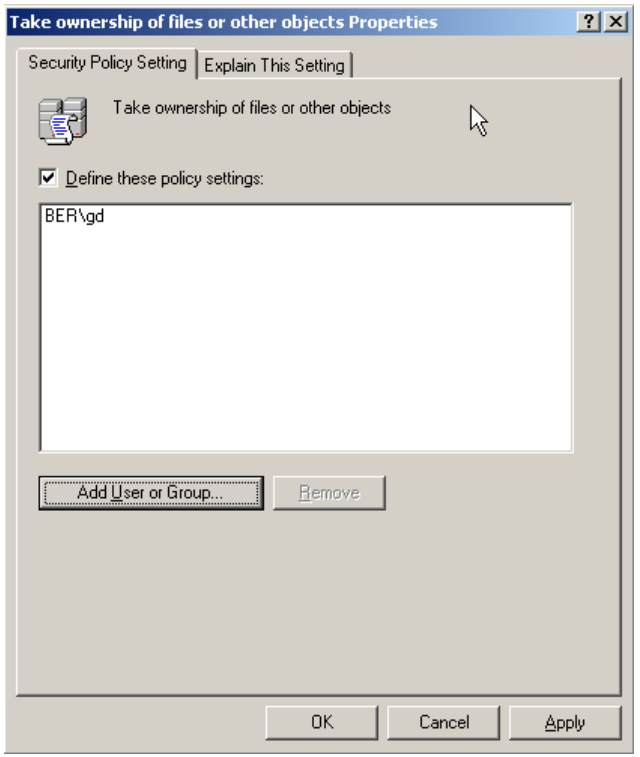
Distributing Privileges with Group Policy

- Open Group Policy Editor / Group Policy Management Console



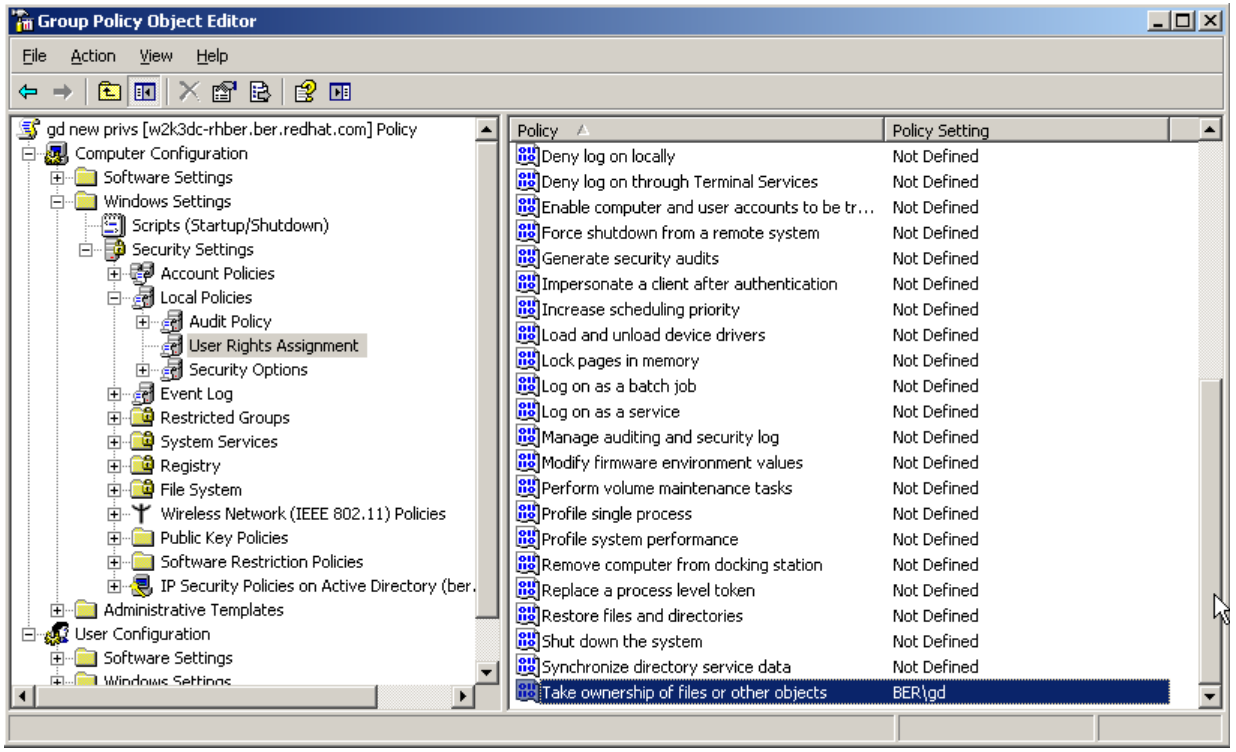
Distributing Privileges with Group Policy

- **Assign “Take Ownership of files” to one user**



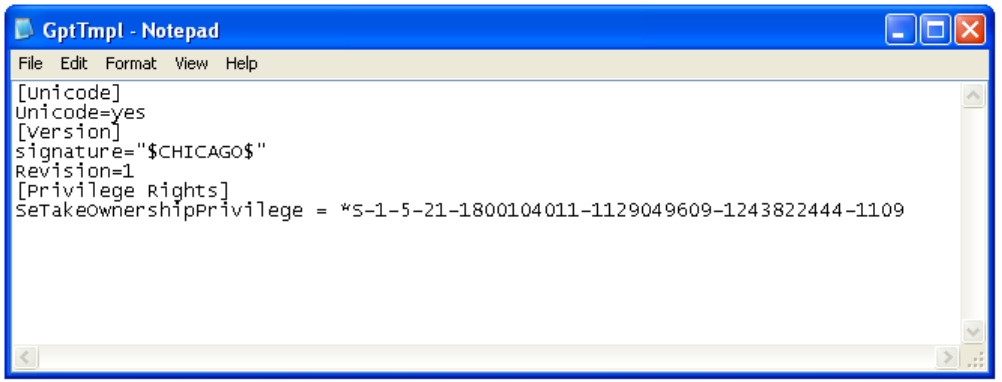
Distributing Privileges with Group Policy

- **Refreshed view:**



Distributing Privileges with Group Policy

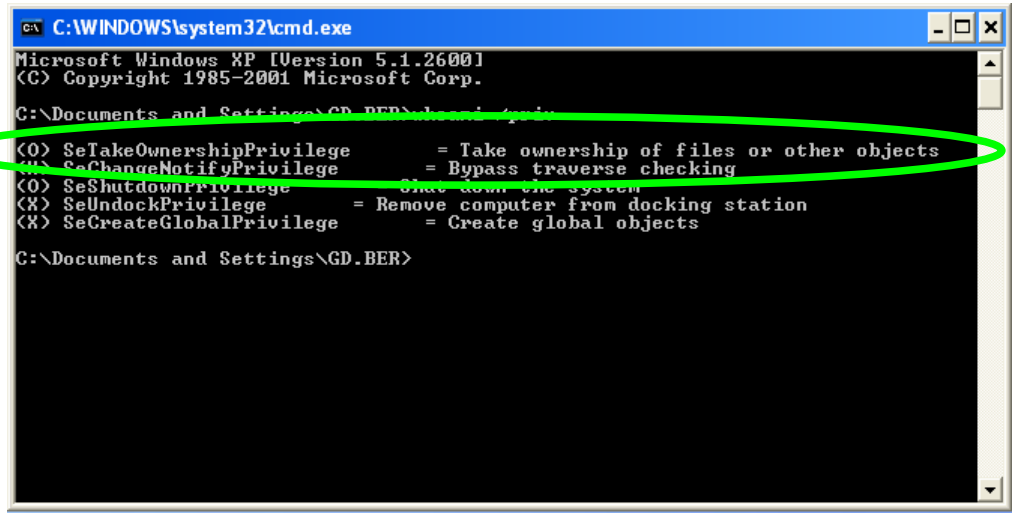
- File generated on the Domain Controller's SYSVOL share:
- `\\mydc\sysvol\policies\ber.redhat.com\{00F0991D-DCBF-4A7A-9723-3F7D3977FDB8}\Machine\Microsoft\Windows NT\SecEdit\GptTmpl.inf`



- The SID is the SID of BER\gd

Distributing Privileges with Group Policy

- Re-login into Windows XP as BER\gd
- `whoami /priv` now shows 5 User Rights including the newly applied User Right
- the "(O)" indicates it was retrieved via Group Policy



Distributing Privileges with Group Policy

- When BER\gd connects remotely to the Samba server, smbd inspects the internal privilege store to see if there are any privileges to grant for that user
- In a next step, smbd inspects the local registry for any extra privileges to apply for that user
- From the Samba smbd log, when user BER\gd connects:

```
[2008/04/13 15:08:38, 10] registry/reg_backend_db.c:regdb_fetch_values(850)
     regdb_fetch_values: Looking for value of key
   [HKLM\SOFTWARE\Samba\Group Policy\State\Machine\Privilege Rights]

[2008/04/13 15:08:38, 10] lib/privileges_gp.c:gp_get_privileges(136)
     gp_get_privileges: no sids for priv: SeTakeOwnershipPrivilege
   Privilege set:
       SE_PRIV  0x0 0x0 0x0 0x0
```

Distributing Privileges with Group Policy

- Samba needs to get aware of the new policy:

```
net ads gpo refresh -P mthelena$

   machine: 'mthelena$' has dn:
   'CN=MTHELENA,OU=jboss,DC=ber,DC=redhat,DC=com'

   * fetching token finished
   * fetching GPO List finished
   * refreshing Group Policy Data finished
   * storing GPO list to registry finished
   * re-reading GPO list from registry finished
```

- Identifies Group Policy Objects for workstation “mthelena$”, downloads policy to $LOCKDIR/gpo_cache
- Stores state information in the registry

Distributing Privileges with Group Policy

- **Finally, Samba needs to apply the new policy:**

```
net ads gpo apply -P mthelena$ -v

machine: 'mthelena$' has dn:
   'CN=MTHELENA,OU=jboss,DC=ber,DC=redhat,DC=com'

[2008/04/13 15:16:46,  0] libgpo/gpext/gpext.c:debug_gpext_header(653)
   security_process_group_policy
       gpo:           {00F0991D-DCBF-4A7A-9723-3F7D3977FDB8}
                      (gd new privs)
       cse extension: 827D319E-6EAC-11D2-A4EA-00C04F79F83A (Security)
       gplink:        DC=ber,DC=redhat,DC=com
       snapin:        803E14A0-B4FB-11D0-A0D0-00A0C90F574B
                      (Security Settings)
       flags:         0x00000041
                      GPO_INFO_FLAG_VERBOSE GPO_INFO_FLAG_MACHINE

[2008/04/13 15:16:46,  0] libgpo/gpo_reg.c:dump_reg_val(792)
   dump_reg_val: STORE 'SeTakeOwnershipPrivilege' '(null)' REG_SZ:
   *S-1-5-21-1800104011-1129049609-1243822444-1109 (length: 47)
```

Distributing Privileges with Group Policy

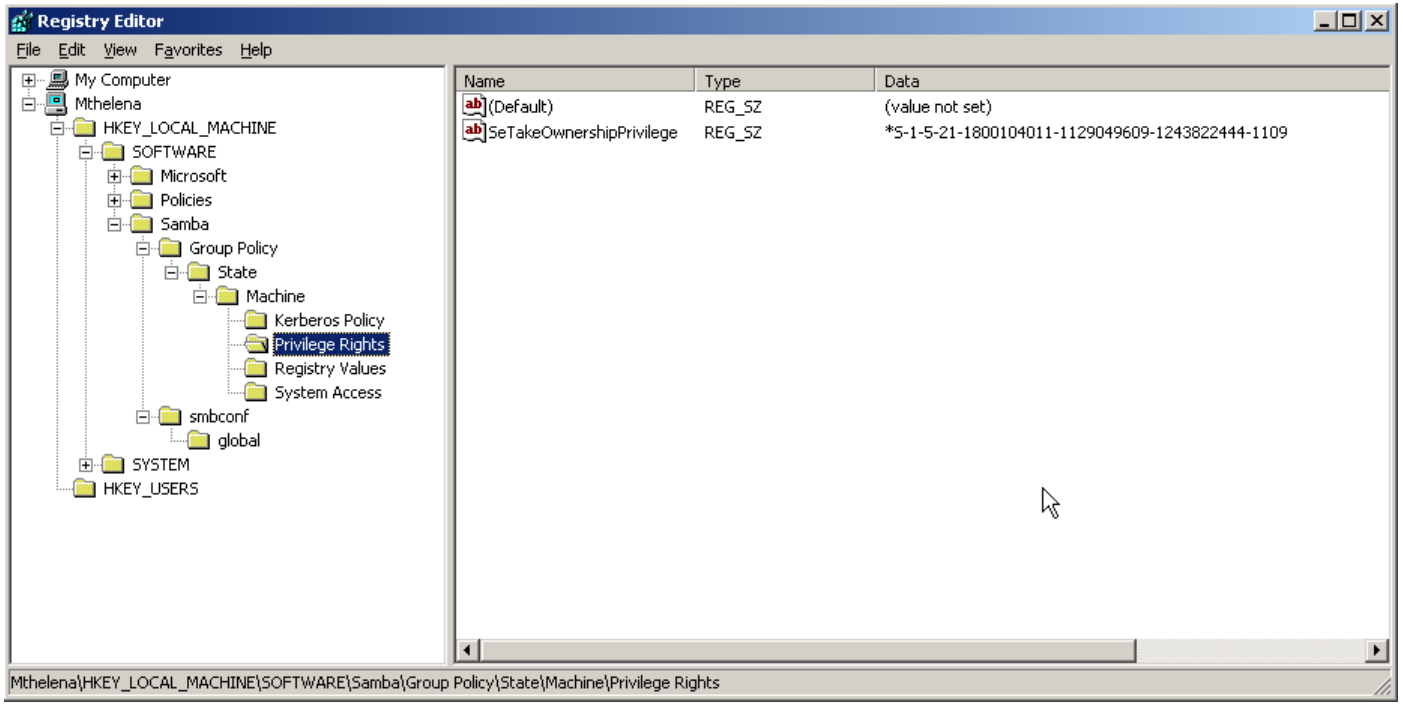
- BER\gd connects again to Samba server
- smbd reinspects the Samba registry and looks for new privileges
- smbd will grant the Ownership privilege to BER\gd:

```
[2008/04/13 15:27:43,  1] lib/privileges_gp.c:gp_get_privileges(149)

  gp_get_privileges: granting priv 'SeTakeOwnershipPrivilege' to sid
   S-1-5-21-1800104011-1129049609-1243822444-1109

  Privilege set:
   SE_PRIV  0x800 0x0 0x0 0x0
```

Distributing Privileges with Group Policy

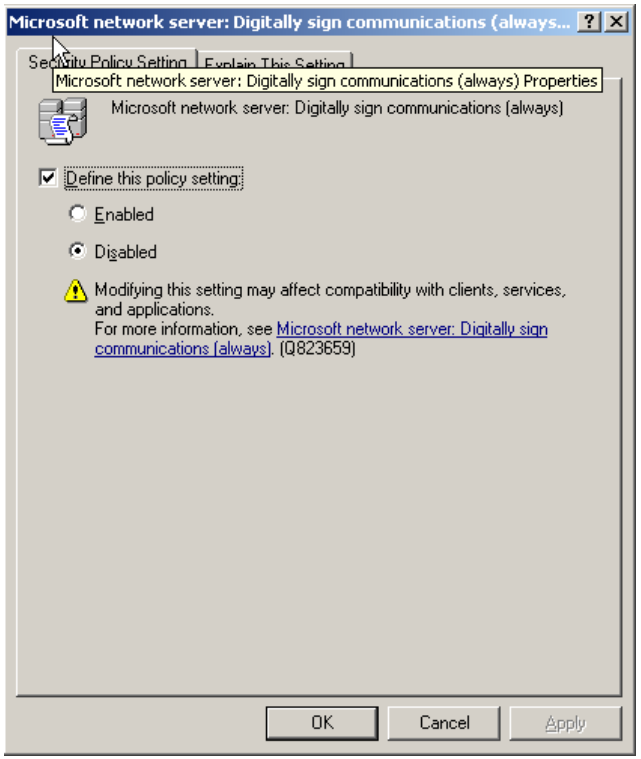


- View on Samba's GPO Privilege Rights Location

SMB Server signing via Group Policy

- SMB server signing enforced in a Windows domain
- Samba is joined to such a domain
- After join no-one in the domain can access Samba's Fileserver as long as an admin manually sets `server signing = yes` in smb.conf
- On Windows, the mandatory server signing is distributed via Group Policy:
- Computer Configuration/Windows Settings/Security Settings/Local Policies/Security Options/Microsoft Network Server: Digitally sign communications (always)
- Samba should transparently follow this policy using its own Policy engine

SMB Server signing via Group Policy



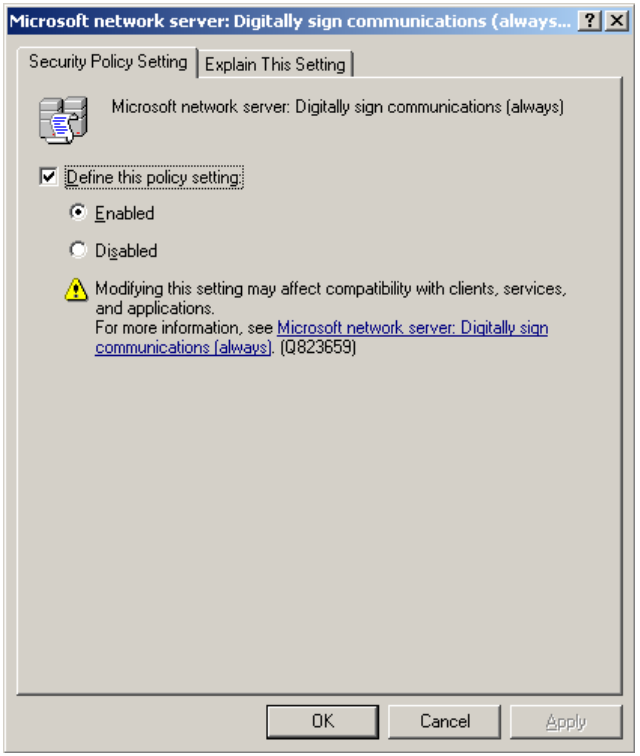
- **SMB signing disabled**

SMB Server signing via Group Policy

- **Initial Samba configuration:**
- **net conf list**

```
[global]
        workgroup = BER
        security = ads
        realm = ber.redhat.com
        server signing = no
```

SMB Server signing via Group Policy



- **SMB signing enabled**

SMB Server signing via Group Policy

- Samba refreshes and applies Policy:
- `net ads gpo refresh mthelena$ -P`
- `net ads gpo apply mthelena$ -P`
- `net conf list`

```
[global]
        workgroup = BER
        security = ads
        realm = ber.redhat.com
        server signing = yes
```

Group Policy and libsmbconf

- Currently only works when using registry configuration backend
- For most of Security Options, the Group Policy engine should feed Samba's internal configuration
- Today libsmbconf can only write into the registry

Thank you for your attention!