



# Joining a domain the windows way

Günther Deschner  
gd@samba.org



(Red Hat / Samba Team)

## 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
- 3<sup>rd</sup> 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 Joining methods in Samba 3.2

- Support for 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  
&**

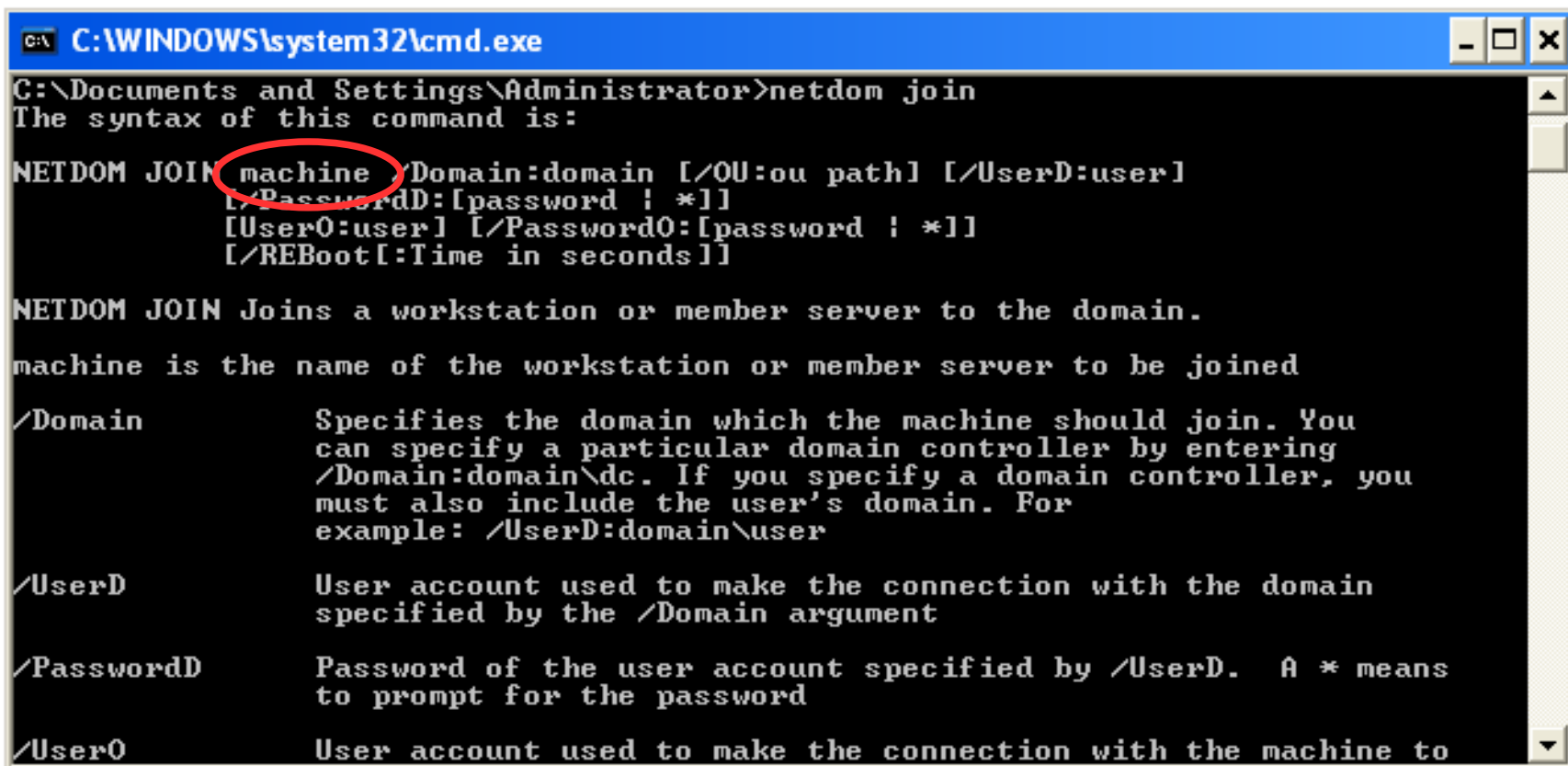
**“config backend = registry”**

# Joining and Unjoining on Windows ?

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

# Windows netdom join tool

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



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrator>netdom join
The syntax of this command is:

NETDOM JOIN machine [/Domain:domain [/OU:ou path] [/UserD:user]
[/PasswordD:[password ! *]]
[User0:user] [/Password0:[password ! *]]
[/REBoot[:Time in seconds]]

NETDOM JOIN Joins a workstation or member server to the domain.
machine is the name of the workstation or member server to be joined

/Domain          Specifies the domain which the machine should join. You
                  can specify a particular domain controller by entering
                  /Domain:domain\dc. If you specify a domain controller, you
                  must also include the user's domain. For
                  example: /UserD:domain\user

/PasswordD       Password of the user account specified by /UserD. A * means
                  to prompt for the password

/Password0       Password of the user account specified by /User0. A * means
                  to prompt for the password

/REBoot          Reboot the machine after joining the domain. The time in
                  seconds to wait before rebooting.

/OU              Organizational Unit (OU) path to join to.

/UserD           User account used to make the connection with the domain
                  specified by the /Domain argument

/User0          User account used to make the connection with the machine to
```

# Windows netdom join tool

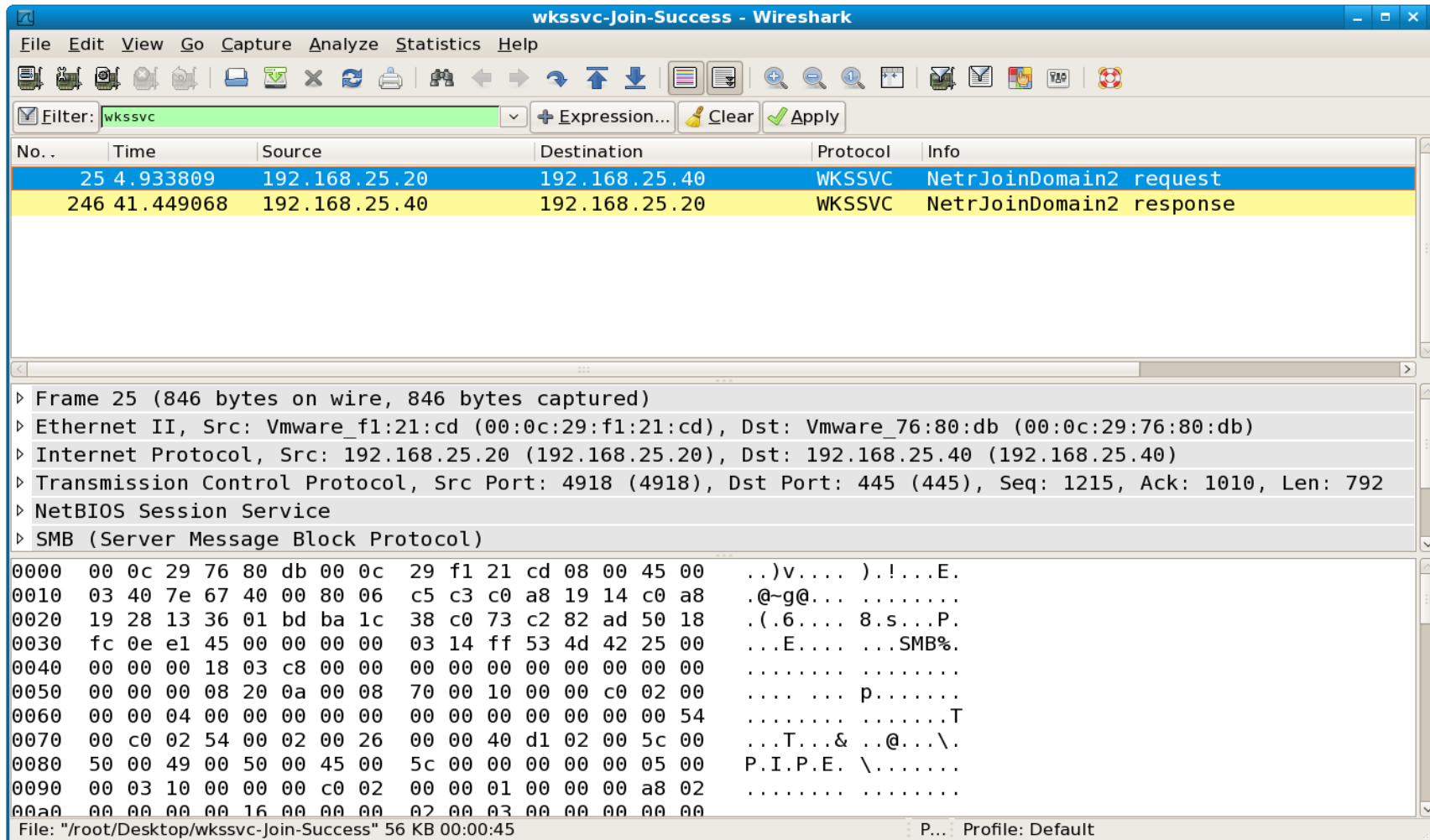
- At that time there was no documentation available
- We wanted to know:
- How does this actually work?
- How are admin credentials transferred to the remote computer?
- What crypto is being used for this?
- What does that command trigger on the remote computer ?
- And once we have all these answers: can we do this as well ?



# 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

# What NetrJoinDomain2 triggers...



The image shows a Wireshark capture window titled "wkssvc-Join-Success - Wireshark". The filter is set to "wkssvc". The packet list shows two packets:

No.	Time	Source	Destination	Protocol	Info
25	4.933809	192.168.25.20	192.168.25.40	WKSSVC	NetrJoinDomain2 request
246	41.449068	192.168.25.40	192.168.25.20	WKSSVC	NetrJoinDomain2 response

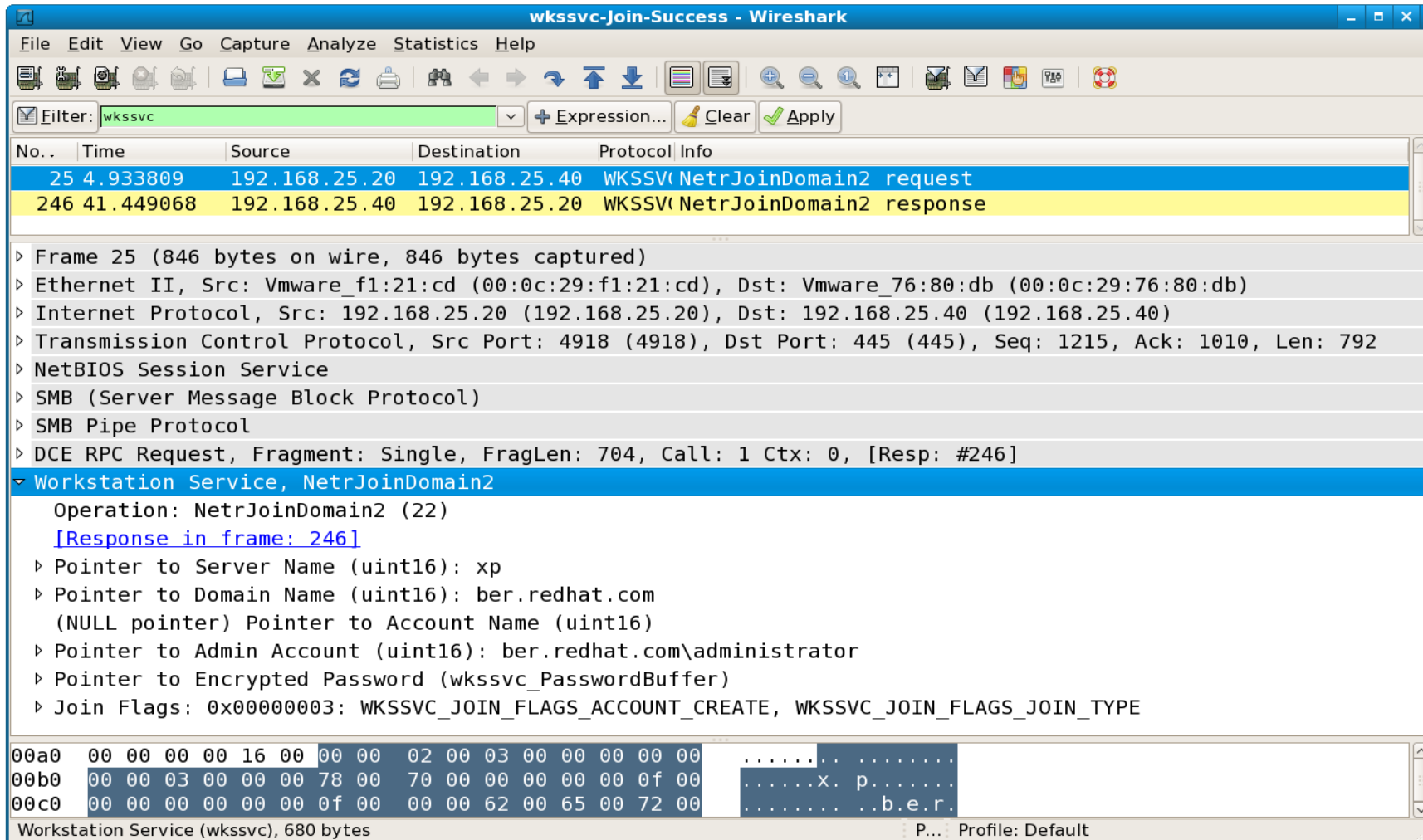
The packet details for the first packet (Frame 25) are:

- Frame 25 (846 bytes on wire, 846 bytes captured)
- Ethernet II, Src: Vmware\_f1:21:cd (00:0c:29:f1:21:cd), Dst: Vmware\_76:80:db (00:0c:29:76:80:db)
- Internet Protocol, Src: 192.168.25.20 (192.168.25.20), Dst: 192.168.25.40 (192.168.25.40)
- Transmission Control Protocol, Src Port: 4918 (4918), Dst Port: 445 (445), Seq: 1215, Ack: 1010, Len: 792
- NetBIOS Session Service
- SMB (Server Message Block Protocol)

The raw data for the SMB protocol shows the following hex and ASCII:

```
0000 00 0c 29 76 80 db 00 0c 29 f1 21 cd 08 00 45 00  ..)v....)!....E.
0010 03 40 7e 67 40 00 80 06 c5 c3 c0 a8 19 14 c0 a8  .@~g@... ..
0020 19 28 13 36 01 bd ba 1c 38 c0 73 c2 82 ad 50 18  .(.6....8.s...P.
0030 fc 0e e1 45 00 00 00 00 03 14 ff 53 4d 42 25 00  ...E....SMB%.
0040 00 00 00 18 03 c8 00 00 00 00 00 00 00 00 00 00  ..
0050 00 00 00 08 20 0a 00 08 70 00 10 00 00 c0 02 00  ....p.....
0060 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 54  ....T
0070 00 c0 02 54 00 02 00 26 00 00 40 d1 02 00 5c 00  ...T...& ..@...\.
0080 50 00 49 00 50 00 45 00 5c 00 00 00 00 00 05 00  P.I.P.E. \.....
0090 00 03 10 00 00 00 c0 02 00 00 01 00 00 00 a8 02  ..
00a0 00 00 00 00 16 00 00 00 02 00 03 00 00 00 00 00
```

# ... the mysterious encrypted password



The image shows a Wireshark capture window titled "wkssvc-Join-Success - Wireshark". The filter is set to "wkssvc". The packet list shows two packets: a request (No. 25) and a response (No. 246). The response packet is expanded to show the "Workstation Service, NetrJoinDomain2" structure. The structure includes fields for "Operation: NetrJoinDomain2 (22)", "Pointer to Server Name (uint16): xp", "Pointer to Domain Name (uint16): ber.redhat.com", "Pointer to Admin Account (uint16): ber.redhat.com\administrator", "Pointer to Encrypted Password (wkssvc\_PasswordBuffer)", and "Join Flags: 0x00000003: WKSSVC\_JOIN\_FLAGS\_ACCOUNT\_CREATE, WKSSVC\_JOIN\_FLAGS\_JOIN\_TYPE". The hex dump at the bottom shows the raw data for the password buffer, which contains the ASCII string "xp.ber".

No.	Time	Source	Destination	Protocol	Info
25	4.933809	192.168.25.20	192.168.25.40	WKSSV	NetrJoinDomain2 request
246	41.449068	192.168.25.40	192.168.25.20	WKSSV	NetrJoinDomain2 response

```
Workstation Service, NetrJoinDomain2
  Operation: NetrJoinDomain2 (22)
  [Response in frame: 246]
  Pointer to Server Name (uint16): xp
  Pointer to Domain Name (uint16): ber.redhat.com
  (NULL pointer) Pointer to Account Name (uint16)
  Pointer to Admin Account (uint16): ber.redhat.com\administrator
  Pointer to Encrypted Password (wkssvc_PasswordBuffer)
  Join Flags: 0x00000003: WKSSVC_JOIN_FLAGS_ACCOUNT_CREATE, WKSSVC_JOIN_FLAGS_JOIN_TYPE
```

00a0 00 00 00 00 16 00 00 00 02 00 03 00 00 00 00 00 .....  
00b0 00 00 03 00 00 00 78 00 70 00 00 00 00 00 0f 00 .....x. p.....  
00c0 00 00 00 00 00 00 0f 00 00 00 62 00 65 00 72 00 .....ber.

# An undocumented 524 byte buffer

- What is the algorithm? Tried all kinds of known and possible combinations using `samba4 smbtorure` tests
- After some weeks, got `WERR_BAD_PASSWORD` instead of `WERR_LOGON_FAILURE` so found the first part of the puzzle
- Final result:
- Cleartext admin password encoded in a 516 byte buffer (like `samr_CryptPassword`)
- 8 byte confounder in front of the 516 byte buffer
- `MD5Update` *first* called with `session_key`, *then* with confounder
- Finally encrypted with RC4 using confounded session key

# Finding a home for remote domain join

- Samba4 `smbtorture` became able to remotely join clients in a domain
- Long term `smbtorture` was surely not the best place for such a feature
- On Windows `NetApi32.lib` provides a `NetJoinDomain` call
- The `NetJoinDomain` call looks very similar to `wkssvc_NetrJoinDomain2` as seen on the wire (except the password buffer)
- Started very basic `libnetapi.so` trying to implement this call

# libnetapi – An API for network management

- Offer similar interface as on windows
- Initially only to support join related calls
- Extended `pidl` version to autogenerate libnetapi based on IDL
- Most calls wrap around rpc call sequences
- Example: NetUserAdd on windows calls:
  - `samr_Connect5,`  
`samr_EnumDomains,`  
`samr_LookupDomain,`  
`samr_OpenDomain,`  
`samr_CreateUser2,`  
`samr_QueryUserInfo,`  
`samr_GetUserPwInfo,`  
`samr_SetUserInfo2`

# libnetapi.so – RPC client library

- Samba 4 IDL based auto-generated rpc client code everywhere
- WKSSVC and WINREG was already done
- migration of LSARPC, SAMR, NETLOGON, DSSETUP and SRVSVC to Samba4 IDL was missing
- Initially only pidl generated *client* was planned but server code needed to be migrated at the same time
- That took some time
- Now fixing or adding new features / infolevels to Samba RPC client or server is much easier

# libnetapi.so - local vs. remote

- In Windows each Net\* call has a `server_name` field as the first argument
- `server_name` is used to distinguish local or remote processing most of the time
- In Samba, local processing could use `LIBSMB_PROG` to talk to the local RPC server.



# Implemented calls in libnetapi

- **NetJoinDomain**
- **NetUnjoinDomain**
- **NetGetJoinInformation**
- **NetGetJoinableOUs**
- **NetServerGetInfo**
- **NetServerSetInfo**
- **NetGetDCName**
- **NetGetAnyDCName**
- **DsGetDcName**
- **NetUserAdd**
- **NetUserDel, NetUserEnum, NetQueryDisplayInformation, etc., etc.**

# NetJoinDomain in netapi.h

- Defined in `netapi.h`
- Very simple call, hides all complexity from the caller
- 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 example binaries

- 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 example binaries

- `net dom join` as a sub command of `net`

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

# NetJoinDomain locally - 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
```

# Samba 3.2 NetrJoinDomain2 server-side

- Samba 3.2 has also support for remote join and unjoin server side
- Admin credentials need to be provided
- Only members of the Domain Admin and Local Administrators group can call this
- Needs to handle more post-processing (local modification, group policy)
- Needs be transaction based



## Live Demo

**Samba joins XP to W2K3**  
**XP joins Samba to W2K3**

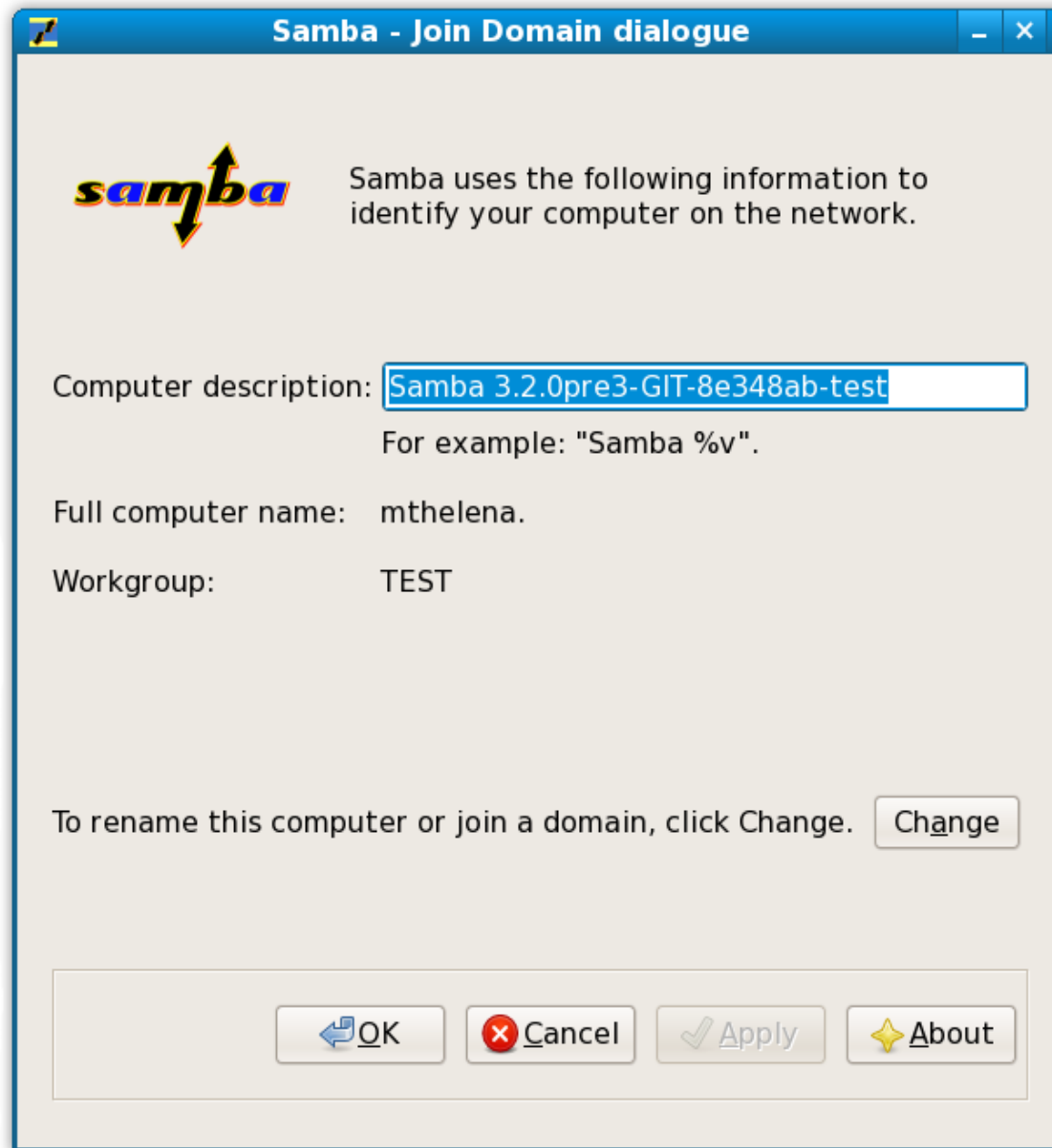
## Grow your own...

- Now that we had the abstraction in libnetjoin, as well as the abstraction in libnetapi, all kinds of new applications can be written
- One experiment was to offer a Windows-like interface for joining a workstation into a domain
- gtk domainjoin gui in lib/netapi/examples



# Live Demo

## netdomjoin-gui



**Computer Name Changes**

You can change the name and membership of this computer. Changes may affect access to network resources.

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 resources.

Computer name:

Full computer name:

mthelena.

Member Of

Domain

Workgroup

Advanced Options


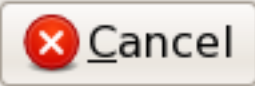
Scan for joinable OUs

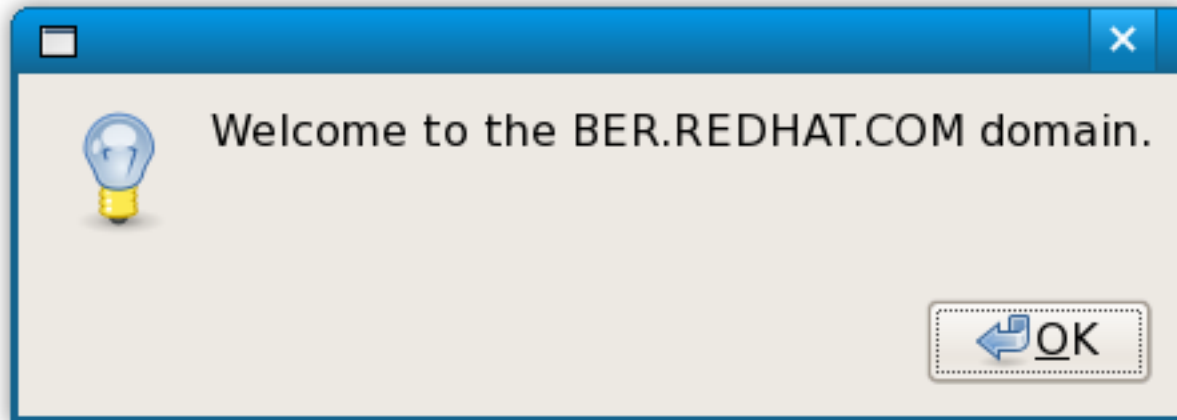
Modify winbind configuration

**Computer Name Changes**

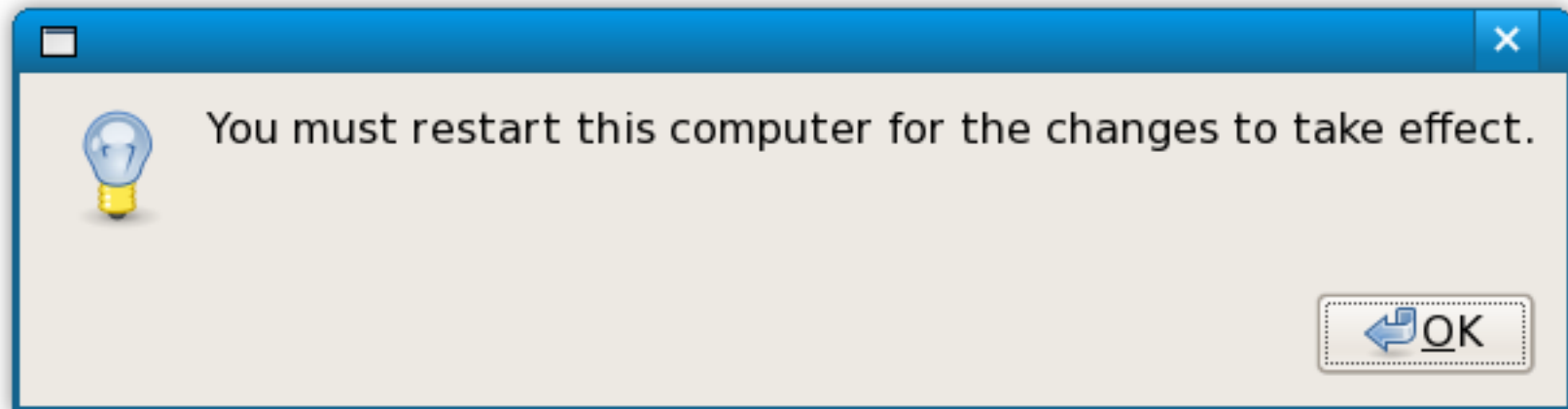
Enter the name and password of an account with permission to leave the domain.  
User name:

Password:







# What's next:

## ■ libnetjoin:

- joining using vendor extensions (script plugin that vendors use to integrate with local system management frameworks)
- Set 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



**Thank you for your attention!**