Openchange.org

samba eXperience

Openchange NextStep

Julien Kerihuel <j.kerihuel@openchange.org>
Pauline Khun <p.khun@openchange.org>

OpenChange overview and current server implementation

Openchange.org

1 Introduction

- What is OpenChange?
  - Open Source implementation of Microsoft Exchange protocols under unix/linux
  - Exchange server replacement under unix/linux
  - Runs on top of Samba4 libraries and as a samba4 module

- Compatibility
  - Libmapi is compatible with Exchange up to 2003
  - Openchange server compatible with Outlook 2000/2003

- License
  - OpenChange source code is under GPL license

- Origin
  - Founded in 2003 in EPITECH – final study project

2

What’s new since SambaXP 2006

- NSPI Protocol
  - Used by Outlook to resolve names from the Exchange Message Store Address Book Provider
  - Used by the mail account wizard when setting up an Exchange Server

- What’s new?
  - Parts of the IDL improved
  - Functions used by Outlook when resolving mail recipients introduced:
    - NspiResolveNames
    - NspiResolveNamesW (UNICODE)
  - EMSABP provider is now working properly

3

EMSABP provider

- What is it?
  - Server side implementation which provides name resolution against Samba4 Active Directory using the NSPI protocol
- How does it work?
  - Extend Samba4 Active Directory with Exchange classes and attributes during openchange setup with:
    - ./setup/oc_provision
  - Extend Samba4 user Active Directory entry with Exchange attributes using:
    - ./setup/oc_newuser
  - At this point the user can be resolved with EMSABP provider
- EMSABP Demonstration

Openchange.org

libmapi
design and implementation

Libmapi underlying components

- Relies on a limited set of Samba4 libraries
  - dcerpc
  - ndr
  - talloc
  - ldb
  - samba-config
- More recently (SambaXP dev sessions)
  - tdr
  - libmagic
- IDL compiled with PIDL

5

MAPI traffic and EcDoRpc

- 99% of MAPI traffic is processed through:
  - EcDoRpc call (0x2)
    - Data obfuscated with XOR 0xA5
  - EcDoRpc_2k3 call (0xb) on Exchange 2003
    - Data obfuscated with XOR 0xA5
    - Data compressed using the XPRESS compression algorithm
    - Additional blob of data

- EcDoRpc packets are divided into 2 parts:
  - Serialized MAPI operations
  - List of MAPI handles used by MAPI operations

6 MAPI calls and serialization

- MAPI calls request:
  - opnum on 1 byte
  - mapi flags on 1 byte
  - handle array index on 1 byte
  - MAPI call data
- MAPI calls response:
  - opnum on 1 byte
  - Handle idx on 1 byte
  - MAPI return value on 4 bytes
  - MAPI call data
- A single EcDoRpc packet generally contains many MAPI calls.

MAPI object handles

- Stored in an array of integers
- Handles are used by MAPI operations to refer to an existing object
- Require Exchange to assign a new handle (0xffffffff value set)

8 Mapitrace analysing tool

- PERL tool designed to analyze MAPI communication and trace calls hierarchy
- Wrapper over ndrdump output
- Provides graph using GraphViz
- Demonstration

Libmapi opacity

- Tends to hide MAPI internal mechanisms from developers
  - Memory allocation and context
  - MAPI objects management

- Focused on application development

MAPI Profiles

- Easy way to use libmapi:
  - Accessed through IProfAdmin *interface*
  - Stored in a LDB database
- Store information users would normally have to supply by hand:
  - Credentials: username, password, domain, workstation
  - Codepage, language and method information
  - Information fetched from the exchange server:
    - Exchange mailbox path
    - Display name
    - Email address
    - Etc.
- Mapiprofile stand-alone binary provides easy access to profile administration

openchangeclient

- MAPI command line messaging tool
- Features:
  - Fetch mails
  - Store attachments on the local filesystem
  - Send mails with to, cc, bcc recipients
  - Send mails with UTF8 only content
  - Send mails with inline HTML content
  - Send mails with HTML body retrieved from a file
  - Send multiple attachments
- Demonstration

exchange2mbox

- Convert Exchange mailbox to mbox file
- Maintain synchronization between mbox and Exchange mailbox through a message ID index database (tdb).
- Support MIME Types for:
  - Content
  - Attachments
- Demonstration

Current libmapi limitations

- We have arbitrary set a stream buffer size to 0x4000 bytes for each WriteStream call
- We do not handle serialization but generate one EcDoRpc packet for each MAPI call
- Mails can’t be sent to SMTP recipients (outside Exchange organization)
- Possible memory consumption leaks
- Bugs due to the lack of testing environments and mystic configurations

Related Tools

Gnome Evolution plugin

- Preview release is available for download
- It should still be considered as unstable code
- Provide a set of features almost similar to openchangeclient:
  - Fetch mails
  - Send mails
  - Delete mails
  - Attachment support
- Demonstration (Please note segmentation fault may occur ;p)

Conclusion

Conclusion

- This is still alpha code but this is a necessary step in openchange development

- OpenChange libmapi roadmap:
  - 0.2 MAILOOK: Mail support (available)
  - 0.4 Calendar support
  - 0.6 Contacts support
  - 0.8 Public folder support
  - 0.9 Tasks support
  - 1.0 Final release candidate

- August 2007: libmapi 1.0 deadline

- Moving from alpha to stable release will depends on how many beta testers we have and how effective is the bug discovery.

-

Conclusion

- Openchange Message Store R&D process will start after SambaXP
- Objective: provide a stable server side for December 2007.

Questions?