DCERPC and Endpoint Mapper

Andreas Schneider <asn@samba.org>

Red Hat

May 11th, 2011
DCERPC and Endpoint Mapper

1. DCERPC
   - How does RPC work?

2. Endpoint Mapper
   - Concept
   - Functions and Details

3. Samba3 RPC Server
   - Overview
   - Robustness
   - Scalability

4. Why?
   - Franky
   - FreelPA
DCERPC and Endpoint Mapper

1. DCERPC
   - How does RPC work?

2. Endpoint Mapper
   - Concept
   - Functions and Details

3. Samba3 RPC Server
   - Overview
   - Robustness
   - Scalability

4. Why?
   - Franky
   - FreelPA

Andreas Schneider <asn@samba.org>
Red Hat
Abbreviations

- DCE: Distributed (Disturbed) Computing Environment
- RPC: Remote Procedure Call
- NDR: Network Data Representation
- IDL: Interface description language
The RPC process

- Client
  - Application
  - Client Stub
  - RPC Library
  - Transport

- Server
  - Application
  - Server Stub
  - RPC Library
  - Transport

Andreas Schneider <asn@samba.org>
Red Hat
Application

- spoolss: Printing application displaying a list of printers
- regedit: Display all values of a key
Client Stubs

- spoolss: Your application calling dcerpc_spoolss_ENUMPrinters
- regedit: Your application calling dcerpc_winreg_ENUMValues
Run-time Library

RPC client implementation creating a RPC bind
- Establishes the connection
- Authenticates the user
Transports

- `ncacn_np`: SMB Named Pipes transport
- `ncacn_ip_tcp`: DCE/RPC over TCP/IP
- `ncalrpc`: Local interprocess communication
- `ncacn_http`: DCE/RPC over HTTP
- `ncadg_ip_udp`, `ncacn_at_dsp`, `ncacn_nb_ipx`, `ncacn_dnet_nsp`, ...
Run-time Library

- The RPC Server accepting a connection over a transport and creating the RPC bind
- After successful authentication it calls the Server Stub
Server Stubs

This unmarshals the packet and calls the application implementation

- spoolss: _spoolss_ENUMPrinters
- regedit: _winreg_ENUMValues
DCERPC and Endpoint Mapper

1. DCERPC
   - How does RPC work?

2. Endpoint Mapper
   - Concept
   - Functions and Details

3. Samba3 RPC Server
   - Overview
   - Robustness
   - Scalability

4. Why?
   - Franky
   - FreeIPA
Abbreviations

- EPM: Endpoint Mapper
- UUID: Universally Unique Identifier (man uuidgen)
- NDR: Network Data Representation
Terminology

- **Endpoint**: An endpoint could be a port or a pipe and provide several interfaces.
- **Interface**: An interface is a RPC service provided by an endpoint.

The named pipe `\\PIPE\netlogon` can be used for netlogon and `lsarpc` connections.
Remember: The RPC process
Endpoint operations

- Each RPC service allocates one or more endpoints dynamically on server startup
- Endpoint mapper maintains information about those endpoints
- The Endpoint Mapper listens on port 135 TCP/IP or on `\PIPE\epmapper`
DCERPC and Endpoint Mapper

1. DCERPC
   - How does RPC work?

2. Endpoint Mapper
   - Concept
   - Functions and Details

3. Samba3 RPC Server
   - Overview
   - Robustness
   - Scalability

4. Why?
   - Franky
   - FreeIPA
Function overview

The most important function of the endpoint mapper.

- **epm_Insert** Add specified entries to an endpoint map.
- **epm_Delete** Delete specified entries from an endpoint map.
- **epm_Lookup** Lookup entries in an endpoint map.
- **epm_Map** Apply some algorithm to an endpoint map to produce a list of protocol towers. (Provide an uuid and get an endpoint)
- **epm_LookupHandleFree** Free an epm_Lookup or epm_Map entry_handle.
Example

Wireshark trace ...
An endpoint tower

A tower has up to 6 floors, 4 at least

1. **Floor1**: Provides the RPC interface identifier (netlogon uuid).
2. **Floor2**: Transfer syntax (NDR endcoded)
3. **Floor3**: RPC protocol identifier (ncacn_tcp_ip, ncacn_np, ...)
4. **Floor4**: Port address (e.g. TCP Port: 49156, PIPE)
5. **Floor5**: Transport (e.g. IP:192.168.51.10, NB:krikkit)
6. **Floor6**: Routing
DCERPC and Endpoint Mapper

1. DCERPC
   - How does RPC work?

2. Endpoint Mapper
   - Concept
   - Functions and Details

3. Samba3 RPC Server
   - Overview
   - Robustness
   - Scalability

4. Why?
   - Franky
   - FreelPA
RPC Endpoints

- Added support for TCP/IP and NCALRPC
- Other processes can register at EPM (OpenChange) over NCALRPC
DCERPC and Endpoint Mapper

1. DCERPC
   - How does RPC work?

2. Endpoint Mapper
   - Concept
   - Functions and Details

3. Samba3 RPC Server
   - Overview
   - Robustness
   - Scalability

4. Why?
   - Franky
   - FreeIPA
Robustness

Client
- RPC service tries to register several times
- After successful registration we do connection monitoring

Server
- We monitor the client connection
- If it goes away, delete the endpoints
DCERPC and Endpoint Mapper

1. DCERPC
   - How does RPC work?

2. Endpoint Mapper
   - Concept
   - Functions and Details

3. Samba3 RPC Server
   - Overview
   - Robustness
   - Scalability

4. Why?
   - Franky
   - FreeIPA
Pre-fork

We started to implement a mutex locking based pre-fork model.

- Parent binds all sockets and then forks a number of children
- Childs have a lock around the accept(3) call
- Prototype working for our spoolss daemon
DCERPC and Endpoint Mapper

1. DCERPC
   - How does RPC work?

2. Endpoint Mapper
   - Concept
   - Functions and Details

3. Samba3 RPC Server
   - Overview
   - Robustness
   - Scalability

4. Why?
   - Franky
   - FreelPA

Andreas Schneider <asn@samba.org>
Red Hat
Franky

- A lot of infrastructure has been created for Franky
- EPM allows us to have multiple daemons
DCERPC and Endpoint Mapper

1. DCERPC
   - How does RPC work?

2. Endpoint Mapper
   - Concept
   - Functions and Details

3. Samba3 RPC Server
   - Overview
   - Robustness
   - Scalability

4. Why?
   - Franky
   - FreeIPA
FreeIPA is something like Active Directory but for Linux only.

- We want to be able to do forest trusts with Active Directory
- For this we need LSA and Netlogon (SAMR)
- pdb_ipa and 'net rpc trust'
Questions & Answers

- Slides http://www.samba.org/~asn/