MSRPC Socket Activation

SambaXP 2021

Volker Lendecke

Samba Team / SerNet

May 2021
What is MSRPC?

- Microsoft-RPC, an extended version of DCE-RPC
- [https://dcerpc.org](https://dcerpc.org): DCE/RPC is an implementation of the Remote Procedure Call technology developed by the Open Group as part of the Distributed Computing Environment. DCE/RPC is most commonly used to interact with Windows network services.
- A lot of Windows services until today depend on MSRPC:
  - Active Directory multi-master replication
  - Remote workstation management
  - Remote printing
  - Even listing shares
  - ... and a lot more
### Windows share listing

<table>
<thead>
<tr>
<th>Destination</th>
<th>Protocol</th>
<th>Length Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>190</td>
<td>Create Request File: srvsvc</td>
<td></td>
</tr>
<tr>
<td>210</td>
<td>Create Response File: srvsvc</td>
<td></td>
</tr>
<tr>
<td>330</td>
<td>Bind: call_id: 2, Fragment: Single, 3 context items</td>
<td></td>
</tr>
<tr>
<td>138</td>
<td>Write Response</td>
<td></td>
</tr>
<tr>
<td>171</td>
<td>Read Request Len:1024 Off:0 File: srvsvc</td>
<td></td>
</tr>
<tr>
<td>206</td>
<td>Bind_ack: call_id: 2, Fragment: Single, max_xmit: 42</td>
<td></td>
</tr>
<tr>
<td>282</td>
<td>NetShareEnumAll request</td>
<td></td>
</tr>
<tr>
<td>486</td>
<td>NetShareEnumAll response</td>
<td></td>
</tr>
<tr>
<td>148</td>
<td>Close Request File: srvsvc</td>
<td></td>
</tr>
<tr>
<td>182</td>
<td>Close Response</td>
<td></td>
</tr>
</tbody>
</table>

- Frame 48: 282 bytes on wire (2256 bits), 282 bytes captured (2256 bits)
- Internet Protocol Version 4, Src: 172.18.103.80, Dst: 172.18.103.109
- NetBIOS Session Service
- SMB2 (Server Message Block Protocol version 2)
- Distributed Computing Environment / Remote Procedure Call (DCE/RPC) Request, Fragment: Single
- Server Service, NetShareEnumAll

Volker Lendecke
MSRPC (3 / 16)
Windows Interoperability

- SMB is just a tiny part of the full Windows client experience
  - Your artificial tests for open/read/write/close work fine
  - Then you try with Windows, and you can’t see your shares
- File srvsvc in share IPC$??
  - Welcome to the wonderful world of Distributed Computing Environment Remote Procedure Calls
- Listing shares in the early days was easy
  - Anybody looked at [MS-RAP]?
  - OS/2 legacy protocol
  - Not sufficient for flexible RPCs
- Microsoft with Windows NT decided to use DCERPC
RPC protocol flow

- Open “srvsvc”
  - “srvsvc” is the server end of a named pipe, like a TCP socket
  - RPC also works over TCP, but MS-RPC predates the ubiquity of TCP, SMB worked over IPX and NetBEUI
  - RPC can even run directly on UDP, but that is less common

- SMB2 write into the srvsvc handle: RPC bind
  - Specify which daemon service to connect (LSA, SAMR, SRVSVC, etc)
  - Authenticate with the service, typically GSSAPI
  - Negotiate transport crypto (plaintext, sign, sign/seal)

- SMB2 IOCTL
  - RPC requests proper
  - NetShareEnumAll lists shares

- RPC over TCP just transmits the raw packets that are encapsulated in SMB2 read/write/ioctl
Samba RPC

- Started in the mid-1990s
  - Windows NT domain interop
- Started with hand-marshalling packets
  - Still done today in fresh implementations
- In 2000, Tridge started PIDL for Samba
  - DCERPC IDL compiler with both omissions and extensions
  - Outputs readable C code
- Back then, the Windows IDLs were still secret
  - Big part of the EU vs MS case
  - Published as part of the Microsoft Protocols
Samba RPC implementation

- All RPC servers are linked into smbd
- Easy to implement, but not “the right thing”
- In May 2010, Simo Sorce started to split spoolssd
  - SMB is just a transport for RPC traffic
  - Goal: Separate printing into a daemon of its own
  - Infrastructure for other RPC services
  - RPC traffic passed through a unix domain socket
- This talk builds upon Simo’s work
- Spoolssd, lsasd and others fork from smbd
  - Separate processes, but still part of /usr/sbin/smbd
DEMO
New daemons

- samba_dcerpcd
  - "inetd" for Samba RPC daemons
  - Listens on behalf of RPC server implementations
- rpcd_epmapper
  - Implementation of DCERPC endpoint mapper
- rpcd_spoolss
  - Simo’s spoolssd in a separate binary
- rpcd_winreg
  - You guessed it – the remote registry server
- rpcd_classic
  - Implement everything else (netlogon, samr, lsa, etc)
- No RPC server code in smbd anymore, just opening named pipes
At startup, ask every rpcd about the interfaces it implements:

```bash
# ./rpdc_winreg --list--interfaces
338cd001−2244−31f1−aaaa−900038001003/0x00000001 winreg
ncacn_np:[\pipe\winreg]
ncacn_ip_tcp:
ncalrpc:
```

- Listens on all sockets for the rpcd_* implementations
- From that specification, create and listen on sockets
- When a client connects, the corresponding rpcd implementation is forked/exec’ed and the socket is passed on via messaging
- samba_dcerpcd completely hands off handling of the connection
  - No DCERPC server implementation required in samba_dcerpcd
Two modes of operation

- `–list-interfaces` just shows what services are provided
- Without `–list-interfaces`, listen on messages from `samba_dcerpcd` for sockets

RPCD implementations don’t create and listen on sockets

Every process can handle multiple RPC connections

- Based on earlier work in the RPC server space

At client disconnect, report number of connections to `samba_dcerpcd`

`samba_dcerpc` knows how many clients each process serves

- Shutdown `rpcd` processes after a timeout (right now 10 sec)
samba_dcerpcd knows all interfaces and endpoints from –list-interfaces

In current master, every source3 based RPC service registers explicitly using epm_Insert

samba_dcerpcd fills a new tdb with all services:

key(48) = "338cd001–2244–31f1–aaaa–900038001003/0x00000001\00"
data(74) = "winreg ncacn_np:[\pipe\winreg] ncacn_ip_tcp:[49152]
calrpc:[rpcd\winreg]"

rccd_epmapper queries and walks this tdb

epm_Insert/Delete right now not not needed
Association Groups

- Policy Handles ("pointers" to server-side objects) can be shared across multiple RPC connections
- An analogy for Unix people

<table>
<thead>
<tr>
<th>association group</th>
<th>unix process</th>
</tr>
</thead>
<tbody>
<tr>
<td>network connection</td>
<td>unix thread</td>
</tr>
<tr>
<td>policy handle</td>
<td>file descriptor</td>
</tr>
</tbody>
</table>

- Not solved in source3 based servers
- "Solved" in source4 by putting all RPC servers that do policy handles into one process
- Clients can ask for association groups, the server assigns the ID
C706 (dcerpc spec) 12.6.3.6: The client should set the assoc_group_id field either to 0 (zero), to indicate a new association group, or to the known value. When the server receives a value of 0, this indicates that the client has requested a new association group, and it assigns a server unique value to the group. This value is returned in the rpc_bind_ack PDU.
Association groups with samba_dcerpcd

- samba_dcerpcd accepts the socket and reads the bind packet
- How to pick one of the N winreg daemons?
- Metze’s idea: 8 of the 32 assoc id bits are a process index
- The socket sent to rpcd_winreg also carries the bind packet
Thanks for your attention

vl@samba.org / vl@sernet.de

http://www.sambaxp.org/