

# New printing protocols in Samba

**Günther Deschner**  
<gd@samba.org>

# Agenda

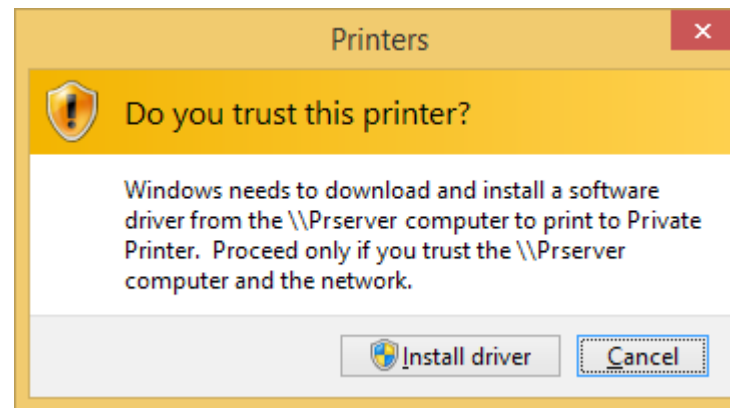
- **MS16-087**
- **RPRN and PAR**
- **PAR support detection**
- **Print Driver Packages**
- **Driver Signing**
- **Core Printer Drivers**
- **Current state of PAR in Samba**
- **Next steps**
- **Further reading & Q/A**

# Samba at RedHat

- Part of Red Hat Gluster Storage Team
- Close relationship with RHEL / Identity Team
- Often collaborate with Andreas Schneider <[asn@samba.org](mailto:asn@samba.org)> on Samba feature or bugfix development such as printing related matters

# MS016-087 (CVE-2016-3238)

- **July 2016: Microsoft released security update to address a critical vulnerability in the Windows spooler components**
- **This update addressed issue by:**
  - Correcting Windows spooler access to filesystem
  - Issue a warning when untrusted printer drivers are attempted to be used
- **V3 non-package aware printer drivers will get security prompt:**



# MS016-087 (CVE-2016-3238)

- **For non-interactive scenarios, the installation of untrusted printer drivers fails completely (!)**
- **September 2016: Discussed issue at Samba / Interoperability Lab in Redmond**
- **October 2016: Microsoft issued follow-up update to mitigate the Point and Print restrictions via white-listing of print servers via Group Policy**
- **Detailed instructions for this mechanism are both described on**
  - [support.microsoft.com](http://support.microsoft.com)
  - [wiki.samba.org](http://wiki.samba.org)
- **What is the real resolution?**

## **MS16-087 resolution:**

**“Update the affected printer driver. Package-aware V3 printer drivers were introduced in Windows Vista. Installing a package-aware printer driver will resolve the issue.”**

**Samba needs to  
support package-  
aware printer  
drivers!**

# What is a package-aware printer driver?

- **A package-aware driver typically comes as a driver package**
- **Microsoft Cabinet Files (.cab)**
  - Printer Driver Inf File (.inf)
  - Driver Catalog File (.cat)
  - “Amd64”, “x86” directories
- **Advantages of Point and Print with driver packages:**
  - All runnable components are part of driver package
  - Driver signing and integrity can be checked on the client during installation
  - Easier to manage (less likely to have overlapping driver files)



# What is a package-aware printer driver?

- **PackageAware keyword in driver.inf:**

- .....  
;These sections are to identify the Vista drivers as "Package Aware" to allow them to  
;take advantage of features such as "Package Point-and-Print" in Vista and above

```
.....  
[PrinterPackageInstallation.amd64]  
PackageAware=TRUE  
CoreDriverDependencies={D20EA372-DD35-4950-9ED8-  
A6335AFE79F0}
```

# What is a package-aware printer driver?

- **Package awareness flag in PrinterDriverAttributes of PRINTER\_INFO\_2**
  - `PRINTER_DRIVER_PACKAGE_AWARE = 0x00000001`
- **Accessible in the driver configuration backend, the Windows registry:**
  - `HKLM\System\CurrentControlSet\Control\Print\Enviroments\Windows x64\Drivers\Version-3\DRIVERNAME\PrinterDriverAttributes`
- **People start manipulating this attribute in the registry to pretend these drivers were properly packaged and securely verified**

# Can we support package-aware drivers?

- **Can we install v3 print driver packages for Point and Print?**
- **Remember: as Samba does not run off Microsoft OS (usually), we need to let a Windows client prepare everything for Point and Print**
- **But: spoolss protocol does not provide means to manage package-aware drivers**
- **=> In order to provide package-aware drivers for Point and Print we need to use a different DCE/RPC protocol**

**For Samba to fully  
support package-  
aware printer  
drivers we need  
MS-PAR!**

# RPRN and PAR

- **PAR is very similar to RPRN**
- **PAR inherits the entire messages and marshalling aspects of RPRN**
- **PAR overcomes fundamental limitations of RPRN**
  - Synchronous delivery of printer change notifications
  - Client print server requirement
  - No driver package management capabilities
- **66 PAR calls out of 74 have a 1:1 match to RPRN calls**
  - 4 new calls for driver management
  - 4 new calls for change notifications

# RPRN and PAR

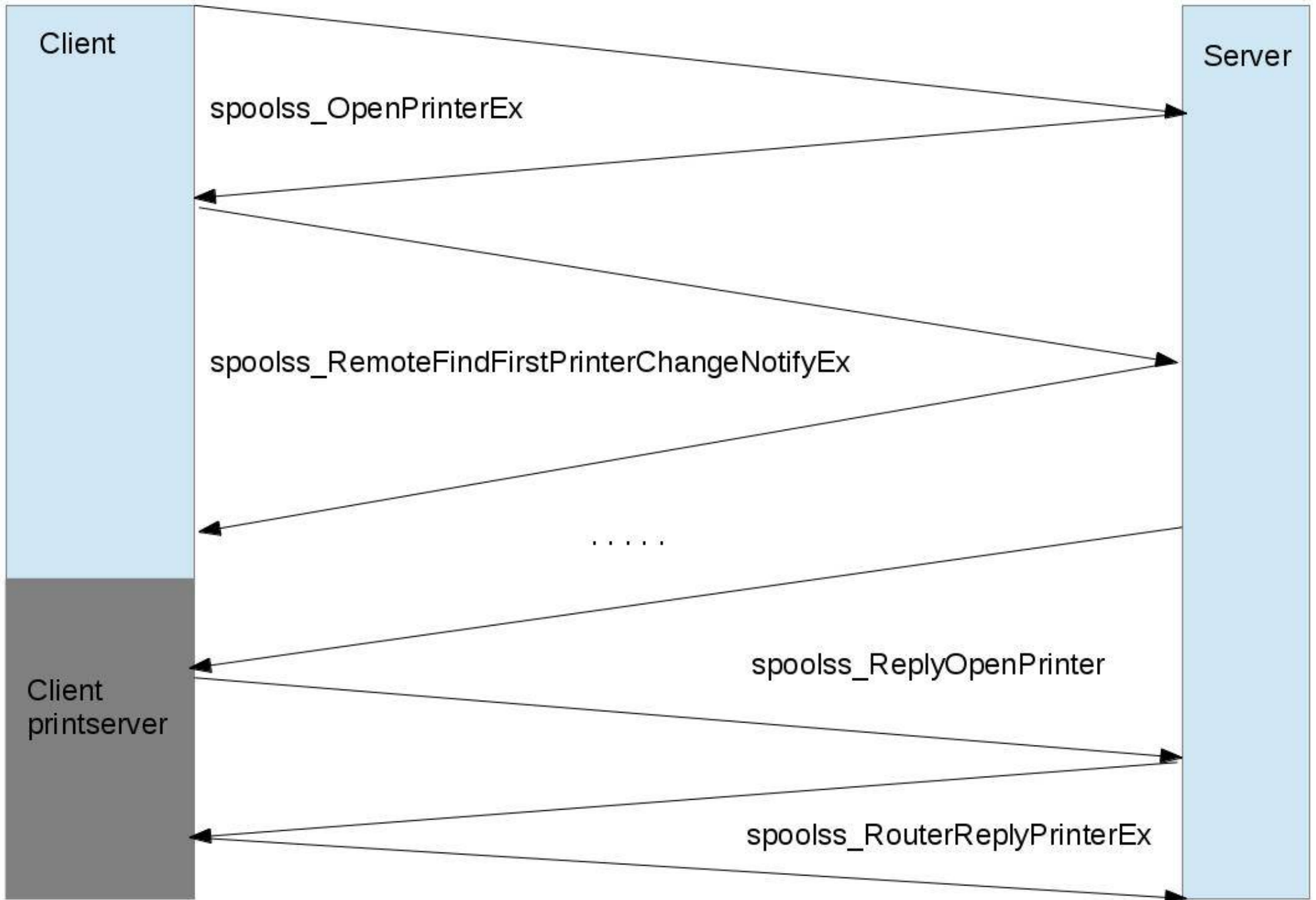
- **RPRN - “Print System Remote Protocol”**
  - ncacn\_np
  - “spoolss”
  - available since Windows NT
  
- **PAR - “Print System Asynchronous Remote Protocol”**
  - ncacn\_ip\_tcp
  - auth\_level >= DCERPC\_AUTH\_LEVEL\_PACKET
  - use of DCE/RPC header object\_uuid  
DCERPC\_PFC\_FLAG\_OBJECT\_UUID
  - “IRemoteWinspool” or “winspool”
  - available since Windows Vista

# PAR support detection

- RPRN named pipe is used for PAR detection
- Client calls “OpenPrinterEx” for a print server handle
- Client calls “GetPrinterData” for “OsVersion”
- Client calls “ClosePrinter” for the print server handle
- Client inspects “OsVersion” binary blob
  - MajorVersion
  - MinorVersion
  - BuildNumber
- If “BuildNumber”  $\geq 3791$  then PAR will be tried

# Printer Change Notifications with RPRN

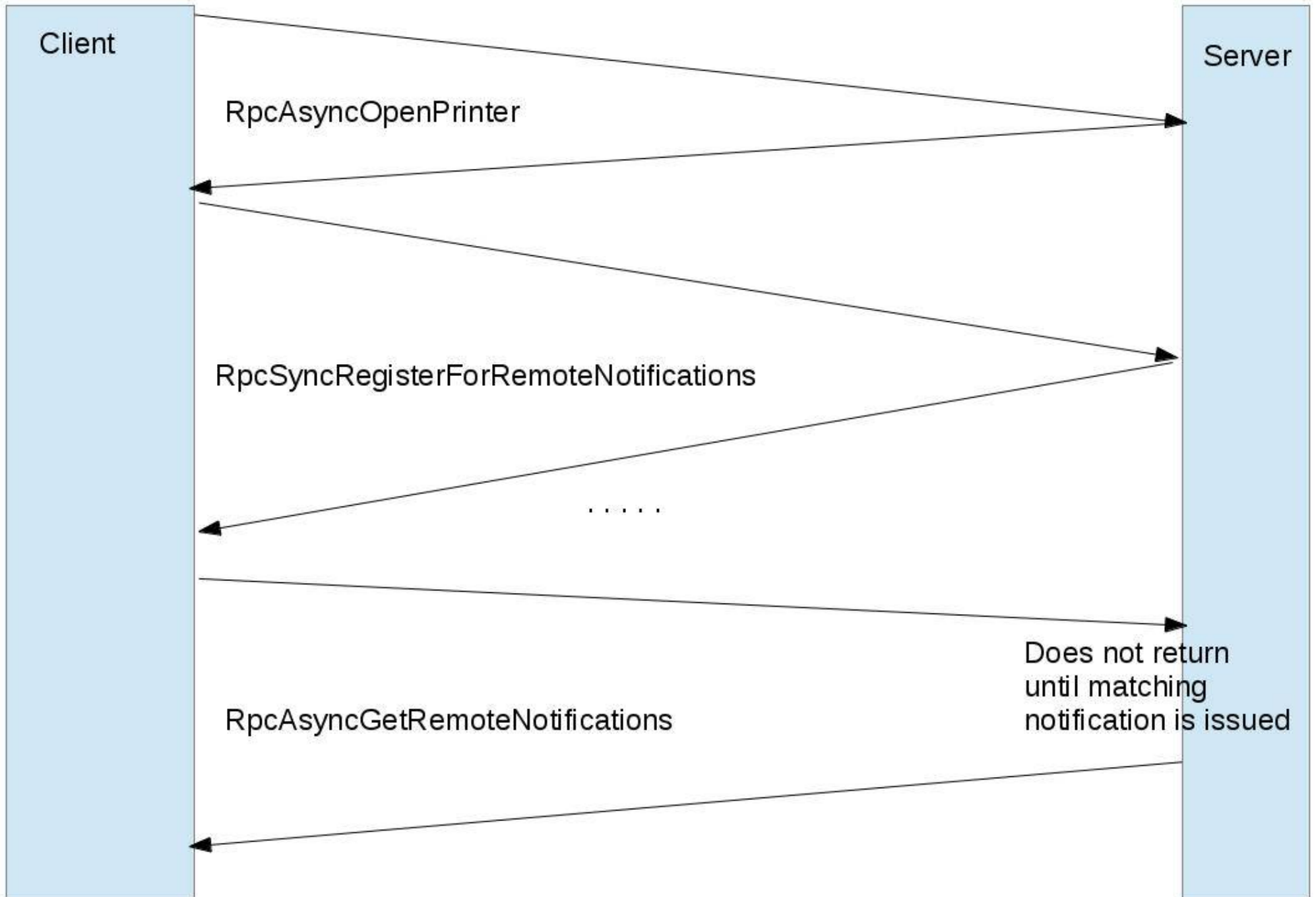




# Printer Change Notifications with PAR

# PAR printer change notify

- **MS-PAR, 3.1.4.9. Printing Related Notification Methods**
  - SyncRegisterForRemoteNotifications
  - SyncUnRegisterForRemoteNotifications
  - SyncRefreshRemoteNotifications
  - AsyncGetRemoteNotifications



# Driver upload with “Print Management” and MS-RPRN

# Driver upload with MS-RPRN

- Client uploads individual driver files via SMB to print\$
- Client calls AddPrinterDriver() DCE/RPC with a fully filled out driver definition
- Server moves files to download area in print\$
- Server registers driver definition in backend (registry)
- Server associates driver with printer (if requested)

# Driver upload with “Print Management” and MS-PAR

# PAR driver management

- **MS-PAR, 3.1.4.2. Printer Driver Management Methods**
  - AsyncInstallPrinterDriverFromPackage
  - AsyncUploadPrinterDriverPackage
  - AsyncCorePrinterDriverInstalled
  - AsyncDeletePrinterDriverPackage



# Driver upload with MS-PAR

- Client uploads driver package components to print\$ via SMB
- Client calls `AsyncUploadPrinterDriverPackage` with SMB path to `driver.inf`
- Server replies returns `driver.inf` path of local, private driver repository
- Client calls `AsyncInstallPrinterDriverFromPackage` with local, private path to `driver.inf` and driver name
- Server parses `driver.inf`, creates driver definition, creates driver package cabinet

# Driver upload with MS-PAR

- **Server has much more work to do with PAR than in RPRN**
  - Printer.inf parsing
  - Cabinet creation

# Prerequisites for implementing MS-PAR in Samba

# DCE/RPC requirements

- Support for `ncacn_ip_tcp` and `endpointmapper`
- Support for `object_uuid` in DCE/RPC header
- Support for `DCERPC_AUTH_LEVEL_PACKET`
- Thanks to Stefan Metzmacher <[metze@samba.org](mailto:metze@samba.org)>

# Print Driver Package components

- **Microsoft Cabinet Files (.cab)**
  - Well documented format, similar to .tar
- **Printer Driver Inf File (.inf)**
  - Well documented format
  - Driver installation instructions
  - Consumed by Windows Setup API
- **Driver Catalog File (.cat)**
  - Mostly undocumented format
  - Cryptographic signatures of Driver Files
- **“Amd64”, “x86” directories**
  - DLLs, XML files

# Cabinet Files – FOSS implementations

- **libmspack** - <https://www.cabextract.org.uk/libmspack/>

- compression support (MSZIP)
- C library API
- Client tool (cabextract)
- NO cabinet creation
- cabinet extraction

- **lcab**

- NO compression
- NO C library API
- Client tool (lcab)
- cabinet creation
- NO cabinet extraction

# Cabinet Files – Samba implementation

- **cab.idl**

- Samba style standard interface definition
- autogenerated marshalling code

- **MSZIP compression builtin in libndr**

- Used for AD replication via DRSUAPI

- **Aurélien Aptel <aaptel@suse.com>**

- Resolved various issues with MSZIP use in libndr
- Created new client tool code

- **libcab.so**

- print server needs to be able to create .cab files on the fly

# Driver signing

- **Andreas Schneider wrote a .cat file parser “parsemscat”**
  - Based on gnutls and libtasn1
  - <https://git.samba.org/?p=asn/samba.git;a=shortlog;h=refs/heads/master-mscat>
  - Needs “Microsoft Root Authority” certificate
- **parsemscat allows to fully verify the integrity of files in a printer driver**
- **PKCS#7 Certificate (Signature) with an embedded data part**
- **The embedded data is an ASN.1 structure call Certificate Trust List**
- **It holds checksums (SHA1, SHA256) for files in the driver package**



# Printer Driver Inf Files

- **.inf files can get very complex**
- **1 Printer Driver Inf File easily describes 100 printers**
- **Created a parser**
  - Based on libgpo and Samba internal ini parsing routines
  - Creates DRIVER\_INFO\_8 structure for in-registry store of Drivers
- **References to Core Printer Drivers**
  - [PrinterPackageInstallation.amd64]  
PackageAware=TRUE  
CoreDriverDependencies={D20EA372-DD35-4950-9ED8-A6335AFE79F0}

# Core Printer Drivers

- **Core Printer Drivers provide basic printing components for UNIDRV, PSCRIPT or XPS based printer drivers**
- **Most Driver Packages refer to Core Printer Drivers**
- **Core Printer Drivers come with the Windows OS (client and server) and are installed via “Windows Update”**
- **Core Printer Drivers are not available for public download**
- **Created “net rpc printer migrate coredrivers” utility to extract Core Printer Drivers directly from a Windows print server**
- **Defined in every Windows OS in ntprint.inf**

# Current State of PAR in Samba

- **Core Iremotewinspool server included in Samba master**
- **90% of Iremotewinspool server calls are detoured to spoolss server automatically (appear as spoolss server calls in the logs)**
  - `winspool_AsyncOpenPrinter (0x00) => spoolss_OpenPrinterEx (0x45)`
  - `winspool_AsyncClosePrinter (0x14) => spoolss_ClosePrinter (0x1d)`
- **Explicit configuration needed for activation:**
  - **OsVersion:**
    - `spoolss:os_major = 6`
    - `spoolss:os_minor = 1`
    - `spoolss:os_build = 9600`
  - **Architecture:**
    - `spoolss:architecture = Windows x64`

# Current State of PAR in Samba

- **spoolssd setup required**

- - rpc\_server:tcpip = yes
  - rpc\_server:epmapper = external
  - rpc\_server:spoolss = external
  - rpc\_server:iremotewinspool = external
  - rpc\_server:register\_embedded\_np = yes

rpc\_daemon:epmd = fork  
rpc\_daemon:spoolssd = fork

spoolssd:prefork\_max\_children = 1  
spoolssd:prefork\_min\_children = 1  
spoolssd:prefork\_spawn\_rate = 1

# Current State of PAR in Samba

## ■ New local DriverStore location

- Currently \$LIBDIR/DriverStore/FileRepository
- Contains extracted driver packages (including Core Drivers)

## ■ Additional remote Driver Packages store

- `\\SERVER\print$\x64\PCC`
- `\\SERVER\print$\W32X86\PCC`
- Contains Driver Packages Cabinet Files

# Current State of PAR in Samba

- **Generously ignoring the new printer change notification calls**
  - AsyncGetRemoteNotifications always returns HRES\_ERROR\_NOT\_SUPPORTED
- **For running “Print Management” under Windows this seems sufficient for the moment**
- **We first need to have a fully async DCE/RPC server in Samba in order to support them**

# Next steps

- **Print Driver Packages**
  - Finish and review Aurelien's MSZIP work for .cab compression
  - Implement Core Printer Drivers
  - Finish .inf parser
- **Support for v4 Printer Driver model ?**
- **DCE/RPC server**
  - Address scalability of ncacn\_ip\_tcp:
  - Properly implement association groups
  - Work on asynchronous DCE/RPC  
(long needed for other protocols like witness as well!)
- **Testing**

# PAR testing client code in Samba

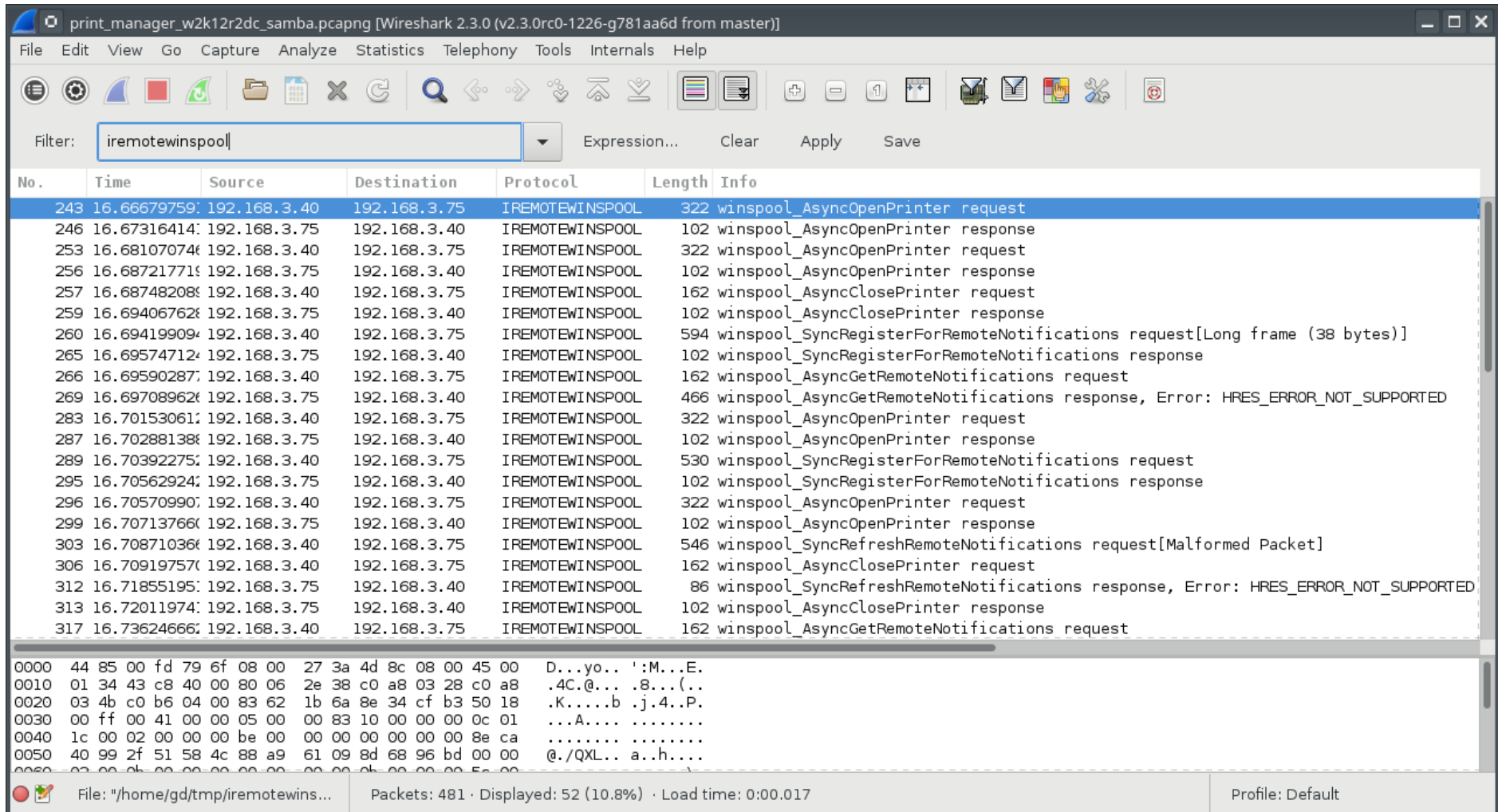
- **rpcclient iremotewinspool command set**
- **smbtorture**
  - **rpc.iremotewinspool**
  - **local.ndr.iremotewinspool**
  - **local.ndr.spoolss (verifies iremotewinspool RPC packets can be understood as spoolss packets)**



# PAR Wireshark dissector

- Fully autogenerated dissector based on Samba winspool.idl
- Has no dependency on older spoolss dissector
  - => does not display all structures yet

# PAR Wireshark dissector



The screenshot shows the Wireshark interface with the filter 'iremotewinspool' applied. The packet list pane displays the following data:

No.	Time	Source	Destination	Protocol	Length	Info
243	16.66679759	192.168.3.40	192.168.3.75	IREMOTEWINSPOOL	322	winspool_AsyncOpenPrinter request
246	16.67316414	192.168.3.75	192.168.3.40	IREMOTEWINSPOOL	102	winspool_AsyncOpenPrinter response
253	16.68107074	192.168.3.40	192.168.3.75	IREMOTEWINSPOOL	322	winspool_AsyncOpenPrinter request
256	16.68721771	192.168.3.75	192.168.3.40	IREMOTEWINSPOOL	102	winspool_AsyncOpenPrinter response
257	16.68748208	192.168.3.40	192.168.3.75	IREMOTEWINSPOOL	162	winspool_AsyncClosePrinter request
259	16.69406762	192.168.3.75	192.168.3.40	IREMOTEWINSPOOL	102	winspool_AsyncClosePrinter response
260	16.69419909	192.168.3.40	192.168.3.75	IREMOTEWINSPOOL	594	winspool_SyncRegisterForRemoteNotifications request[Long frame (38 bytes)]
265	16.69574712	192.168.3.75	192.168.3.40	IREMOTEWINSPOOL	102	winspool_SyncRegisterForRemoteNotifications response
266	16.69590287	192.168.3.40	192.168.3.75	IREMOTEWINSPOOL	162	winspool_AsyncGetRemoteNotifications request
269	16.69708962	192.168.3.75	192.168.3.40	IREMOTEWINSPOOL	466	winspool_AsyncGetRemoteNotifications response, Error: HRES_ERROR_NOT_SUPPORTED
283	16.70153061	192.168.3.40	192.168.3.75	IREMOTEWINSPOOL	322	winspool_AsyncOpenPrinter request
287	16.70288138	192.168.3.75	192.168.3.40	IREMOTEWINSPOOL	102	winspool_AsyncOpenPrinter response
289	16.70392275	192.168.3.40	192.168.3.75	IREMOTEWINSPOOL	530	winspool_SyncRegisterForRemoteNotifications request
295	16.70562924	192.168.3.75	192.168.3.40	IREMOTEWINSPOOL	102	winspool_SyncRegisterForRemoteNotifications response
296	16.70570990	192.168.3.40	192.168.3.75	IREMOTEWINSPOOL	322	winspool_AsyncOpenPrinter request
299	16.70713766	192.168.3.75	192.168.3.40	IREMOTEWINSPOOL	102	winspool_AsyncOpenPrinter response
303	16.70871036	192.168.3.40	192.168.3.75	IREMOTEWINSPOOL	546	winspool_SyncRefreshRemoteNotifications request[Malformed Packet]
306	16.70919757	192.168.3.40	192.168.3.75	IREMOTEWINSPOOL	162	winspool_AsyncClosePrinter request
312	16.71855195	192.168.3.75	192.168.3.40	IREMOTEWINSPOOL	86	winspool_SyncRefreshRemoteNotifications response, Error: HRES_ERROR_NOT_SUPPORTED
313	16.72011974	192.168.3.75	192.168.3.40	IREMOTEWINSPOOL	102	winspool_AsyncClosePrinter response
317	16.73624666	192.168.3.40	192.168.3.75	IREMOTEWINSPOOL	162	winspool_AsyncGetRemoteNotifications request

The bottom pane shows the raw packet data in hexadecimal and ASCII format:

```
0000 44 85 00 fd 79 6f 08 00 27 3a 4d 8c 08 00 45 00  D...yo.. ':M...E.  
0010 01 34 43 c8 40 00 80 06 2e 38 c0 a8 03 28 c0 a8  .4C.@... .8...(.  
0020 03 4b c0 b6 04 00 83 62 1b 6a 8e 34 cf b3 50 18  .K.....b .j.4..P.  
0030 00 ff 00 41 00 00 05 00 00 83 10 00 00 00 0c 01  ...A.... .....  
0040 1c 00 02 00 00 00 be 00 00 00 00 00 00 00 8e ca  ..... .....  
0050 40 99 2f 51 58 4c 88 a9 61 09 8d 68 96 bd 00 00  @./QL.. a..h....  
0060 02 00 0b 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
```

## Further reading

- **Microsoft Protocol Documentation:**
  - MS-RPRN, MS-PAR, MS-PRSOD
- **Printer Driver INF Files:**
  - <https://docs.microsoft.com/en-us/windows-hardware/drivers/print/printer-inf-files>
- **Microsoft Cabinet File Format:**
  - [https://msdn.microsoft.com/library/bb417343.aspx#cabinet\\_format](https://msdn.microsoft.com/library/bb417343.aspx#cabinet_format)
- **MS16-087: Security update for Windows print spooler components: July 12, 2016**
  - <https://support.microsoft.com/en-us/help/3170005/ms16-087-security-update-for-windows-print-spooler-components-july-12,-2016>

# Questions and answers

- Mail [gd@samba.org](mailto:gd@samba.org), [asn@samba.org](mailto:asn@samba.org)
- #samba-technical on [irc.freenode.net](https://www.freenode.net/)
- WIP branches:
  - <https://git.samba.org/?p=gd/samba/.git;a=shortlog;h=refs/heads/master-par-ok>
  - <https://git.samba.org/?p=gd/wireshark/.git;a=shortlog;h=refs/heads/master-iremotewi>  
nspool

# Thank you for your attention!

**[www.redhat.com](http://www.redhat.com)  
[www.samba.org](http://www.samba.org)**

**<[gd@samba.org](mailto:gd@samba.org)>**