Exploring Ancient Ruins to Find Modern Bugs: Discovering a 0-Day in MS-RPC Service

Ben Barnea & Ophir Harpaz
whoweare

Ben Barnea
Security Researcher
Akamai

@nachoskrnl

Ophir Harpaz
Security Research team lead
Akamai

@OphirHarpaz
Why MS-RPC?
It's everywhere :|
... and between everyone
Yet not much public research

Most information boils down to:

- MSFT documentation
- Several research-oriented blog posts
- Few public vulnerabilities

Why so?
Potential impact:

Lateral Movement & Privilege Escalation
Our agenda for today

- MS-RPC introduction and overview
- MS-RPC (in)security
- A 0-day in a Windows service
MS-RPC Overview
Terminology you’ll soon master

- Interface
- {M}IDL
- Transport
- Endpoint
- Binding
The RPC Client-Server Model
The RPC Client-Server Model

Client

Foo(5, "Hello")

Server
The RPC Client-Server Model

[uuid(12345678-4000-2006-0000-2000000001a)]

interface Test
{
    void Foo([in] int number,
              [in] char *message);
    void Bar([out] int * result);
}
The RPC Client-Server Model

```
[ uuid(12345678-4000-2006-0000-20000000001a) ]

interface Test {
    void Foo([in] int number, [in] char *message);
    void Bar([out] int * result);
}
```

Client

Server

```
Foo(5, "Hello")
```

MIDL.exe

```
Test_s.c
Test.h
Test_c.c
```
The RPC Client-Server Model

```c
[ uuid(12345678-4000-2006-0000-2000000001a) ]

interface Test {
    void Foo([in] int number, [in] char *message);
    void Bar([out] int * result);
}
```

MIDL.exe

```c
Foo(5, "Hello")
```
Endpoints

- The server registers an endpoint using a certain transport

<table>
<thead>
<tr>
<th>Transports</th>
<th>Protocol Sequence</th>
<th>Endpoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>ncacn_ip_tcp</td>
<td>&lt;port number&gt;</td>
</tr>
<tr>
<td>Named pipe</td>
<td>ncacn_np</td>
<td>&lt;pipe name&gt;</td>
</tr>
<tr>
<td>UDP</td>
<td>ncadg_ip_udp</td>
<td>&lt;port number&gt;</td>
</tr>
<tr>
<td>ALPC</td>
<td>ncalrpc</td>
<td>&lt;ALPC port&gt;</td>
</tr>
<tr>
<td>HTTP</td>
<td>ncacn_http</td>
<td>&lt;hostname&gt;</td>
</tr>
<tr>
<td>Hyper-V socket</td>
<td>ncacn_hvsocket</td>
<td>&lt;UUID&gt;</td>
</tr>
</tbody>
</table>
## Endpoint Examples

<table>
<thead>
<tr>
<th>Pid</th>
<th>Protocol</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1260</td>
<td>nacn_hvsocket</td>
<td>DA32E2E81-3B3E-99A1-9000-81B74B9030E</td>
</tr>
<tr>
<td>1260</td>
<td>nacn_ip_tcp</td>
<td>135</td>
</tr>
<tr>
<td>5516</td>
<td>nacn_ip_tcp</td>
<td>4290</td>
</tr>
<tr>
<td>876</td>
<td>nacn_ip_tcp</td>
<td>49666</td>
</tr>
<tr>
<td>2008</td>
<td>nacn_ip_tcp</td>
<td>49667</td>
</tr>
<tr>
<td>5176</td>
<td>nacn_ip_tcp</td>
<td>49668</td>
</tr>
<tr>
<td>2008</td>
<td>nacn_np</td>
<td>\PIPE\atsvc</td>
</tr>
<tr>
<td>1260</td>
<td>nacn_np</td>
<td>\pipe\epmmapper</td>
</tr>
<tr>
<td>876</td>
<td>nacn_np</td>
<td>\pipe\eventlog</td>
</tr>
<tr>
<td>7328</td>
<td>nacn_np</td>
<td>\PIPE\ROUTER</td>
</tr>
<tr>
<td>5176</td>
<td>nacn_np</td>
<td>\pipe\spoolss</td>
</tr>
<tr>
<td>6260</td>
<td>nacn_np</td>
<td>\PIPE\srvsvc</td>
</tr>
<tr>
<td>6472</td>
<td>nacn_np</td>
<td>\pipe\trkvs</td>
</tr>
<tr>
<td>1564</td>
<td>nacn_np</td>
<td>\PIPE\W32TIME_ALT</td>
</tr>
<tr>
<td>5504</td>
<td>nacn_np</td>
<td>\PIPE\wkssv</td>
</tr>
<tr>
<td>25284</td>
<td>ncalrpc</td>
<td>5c2105c3-bbfa-4a23-85b9-da77c736639c</td>
</tr>
<tr>
<td>1148</td>
<td>ncalrpc</td>
<td>actkernel</td>
</tr>
<tr>
<td>6028</td>
<td>ncalrpc</td>
<td>AppV-ISV-APPV-jtv_server</td>
</tr>
<tr>
<td>6028</td>
<td>ncalrpc</td>
<td>AppV-ISV-f432e7e9-769f-460c-a3fe-7de4ed58ed3...</td>
</tr>
<tr>
<td>6028</td>
<td>ncalrpc</td>
<td>AppV-ISV-f432e7e9-769f-460c-a3fe-7de4ed58ed3...</td>
</tr>
<tr>
<td>6028</td>
<td>ncalrpc</td>
<td>AppV-ISV-f432e7e9-769f-460c-a3fe-7de4ed58ed3...</td>
</tr>
<tr>
<td>6028</td>
<td>ncalrpc</td>
<td>AppV-ISV-f432e7e9-769f-460c-a3fe-7de4ed58ed3...</td>
</tr>
</tbody>
</table>
Well-Known Endpoints

Server

Foo(5, “Hello”)
[TCP port 39776]

Client

Dynamic Endpoints
Well-Known Endpoints

- Server
  - Foo(5, “Hello”) [TCP port 3976]
- Client

Dynamic Endpoints

- EP Mapper
  - Hi I need server <UUID> [TCP port 135]
- Server
  - Client
Well-Known Endpoints

Server

Foo(5, “Hello”)
[TCP port 39776]

Client

Dynamic Endpoints

EP Mapper

Hi I need
server <UUID>
[TCP port 135]

Client

Server

Ok talk
to TCP
port
50501

Client
Well-Known Endpoints

Server

```
Foo(5, "Hello")
[TCP port 39776]
```

Client

Dynamic Endpoints

Server

```
Hi I need server <UUID>
[TCP port 135]
```

EP Mapper

```
Ok talk to TCP port 50501
```

Client

```
Foo(5, "Hello")
[TCP port 50501]
```
### Task Scheduler Service Remoting Protocol

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUID_ATSvc</td>
<td>1FF70682-0A51-30E8-076D-740BE8CEE98B</td>
<td>ATSvc UUID version 1.0</td>
</tr>
<tr>
<td>GUID_SASec</td>
<td>378E52B0-C0A9-11CF-822D-00AA0051E40F</td>
<td>SASec UUID version 1.0</td>
</tr>
<tr>
<td>GUID_ITaskSchedulerService</td>
<td>86D35949-83C9-4044-B424-DB3633231FD0C</td>
<td>ITaskSchedulerService UUID version 1.0</td>
</tr>
</tbody>
</table>

### Service control manager remote protocol

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPC interface UUID</td>
<td>{367ABB81-9844-35F1-AD32-98F038001003}</td>
</tr>
<tr>
<td>Named pipe</td>
<td>\PIPE\svcctl</td>
</tr>
</tbody>
</table>

### Encrypting File System Remote (EFSRPC) Protocol

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPC Well-Known Endpoint</td>
<td>\pipe\lsarpc&lt;3&gt;</td>
</tr>
<tr>
<td>RPC Interface UUID</td>
<td>{c681d488-d850-11d0-8c52-00c04fd90f7e}</td>
</tr>
<tr>
<td>RPC Well-Known Endpoint</td>
<td>\pipe\efsrpc</td>
</tr>
<tr>
<td>RPC Interface UUID</td>
<td>{df1941c5-fe89-4e79-bf10-463657ac4f4d}</td>
</tr>
</tbody>
</table>
Task Scheduler Endpoint Resolution

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Destination IP</th>
<th>Protocol</th>
<th>Source Port</th>
<th>Destination Port</th>
<th>Request Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>66</td>
<td>63325</td>
<td>135 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0 MS</td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>TCP</td>
<td>66</td>
<td>135</td>
<td>63325 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192 Len=0</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>54</td>
<td>63325</td>
<td>135 [ACK] Seq=1 Ack=1 Win=2102272 Len=0</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCRPC</td>
<td>214</td>
<td>Bind: call_id: 2</td>
<td>Fragment: Single, 3 context items</td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>DCRPC</td>
<td>162</td>
<td>Bind_ack: call_id: 2</td>
<td>Fragment: Single, max_xmit: 50</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>EPM</td>
<td>222</td>
<td>Map request, TaskSchedulerService, 32bit NDR</td>
<td></td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>EPM</td>
<td>226</td>
<td>Map response, TaskSchedulerService, 32bit NDR</td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>66</td>
<td>63326</td>
<td>49666 [SYN, ECN, CWR] Seq=0 Win=8192 Len=0</td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>TCP</td>
<td>66</td>
<td>49666</td>
<td>63326 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>54</td>
<td>63326</td>
<td>49666 [ACK] Seq=1 Ack=1 Win=2102272 Len=0</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCRPC</td>
<td>262</td>
<td>Bind: call_id: 2</td>
<td>Fragment: Single, 3 context items</td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>DCRPC</td>
<td>388</td>
<td>Bind_ack: call_id: 2</td>
<td>Fragment: Single, max_xmit: 50</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCRPC</td>
<td>594</td>
<td>AUTH3: call_id: 2</td>
<td>Fragment: Single, NTLMSSP_AUTH, U</td>
</tr>
</tbody>
</table>
## Task Scheduler Endpoint Resolution

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Destination IP</th>
<th>Protocol</th>
<th>Source Port</th>
<th>Destination Port</th>
<th>Flags</th>
<th>Sequence</th>
<th>Window</th>
<th>Acknowledgment</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>66</td>
<td>63325</td>
<td>SYN</td>
<td>0</td>
<td>8192</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>TCP</td>
<td>66</td>
<td>135</td>
<td>SYN, ACK</td>
<td>0</td>
<td>8192</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>54</td>
<td>63325</td>
<td>[ACK] Seq=1 Ack=1</td>
<td>2102272</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCERPC</td>
<td>214 Bind: call_id: 2, Fragment: Single, 3 context items</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>DCERPC</td>
<td>162 Bind ack: call_id: 2, Fragment: Single, max xmit: 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>EPM</td>
<td>222 Map request, TaskSchedulerService, 32bit NDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>EPM</td>
<td>226 Map response, TaskSchedulerService, 32bit NDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>66</td>
<td>63326</td>
<td>SYN, ECN</td>
<td>0</td>
<td>8192</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>TCP</td>
<td>66</td>
<td>49666</td>
<td>SYN, ACK, ECN</td>
<td>0</td>
<td>8192</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>54</td>
<td>63326</td>
<td>[ACK] Seq=1 Ack=1</td>
<td>2102272</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCERPC</td>
<td>262 Bind: call_id: 2, Fragment: Single, 3 context items</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>DCERPC</td>
<td>388 Bind ack: call_id: 2, Fragment: Single, max xmit: 50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCERPC</td>
<td>594 AUTH3: call_id: 2, Fragment: Single, NTLMSSP_AUTH, User</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Task Scheduler Endpoint Resolution

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Destination IP</th>
<th>Protocol</th>
<th>Sequence</th>
<th>Acknowledgment</th>
<th>Windows</th>
<th>Length</th>
<th>Additional Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>66 63325 → 135</td>
<td>[SYN, ECN, CWR] Seq=0 Win=8192 Len=0</td>
<td>MSFTP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>TCP</td>
<td>66 135 → 63325</td>
<td>[SYN, ACK, ECN] Seq=0 Ack=1 Win=8192 Len=0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>54 63325 → 135</td>
<td>[ACK] Seq=1 Ack=1 Win=2102272 Len=0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCERPC</td>
<td>214 Bind: call_id: 2, Fragment: Single, 3 context items</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>DCERPC</td>
<td>162 Bind ack: call_id: 2, Fragment: Single, max xmit: 50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>EPM</td>
<td>222 Map request, TaskSchedulerService, 32bit NDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>EPM</td>
<td>226 Map response, TaskSchedulerService, 32bit NDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>66 63326 → 49666</td>
<td>[SYN, ECN, CWR] Seq=0 Win=8192 Len=0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>TCP</td>
<td>66 49666 → 63326</td>
<td>[SYN, ACK, ECN] Seq=0 Ack=1 Win=8192 Len=0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>54 63326 → 49666</td>
<td>[ACK] Seq=1 Ack=1 Win=2102272 Len=0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCERPC</td>
<td>262 Bind: call_id: 2, Fragment: Single, 3 context items</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>DCERPC</td>
<td>388 Bind_ack: call_id: 2, Fragment: Single, max xmit: 50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCERPC</td>
<td>594 AUTH3: call_id: 2, Fragment: Single, NTLMSSP_AUTH, U</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Task Scheduler Endpoint Resolution

<table>
<thead>
<tr>
<th>IP Address</th>
<th>IP Address</th>
<th>Protocol</th>
<th>Source Port</th>
<th>Destination Port</th>
<th>Flags</th>
<th>Sequence</th>
<th>Acknowledgment</th>
<th>Window Size</th>
<th>Length</th>
<th>Method</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>66 63325</td>
<td>135 [SYN, ECN, CWR]</td>
<td>Seq=0</td>
<td>Win=8192</td>
<td>Ack=1</td>
<td>Len=0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>TCP</td>
<td>66 135</td>
<td>63325 [SYN, ACK, ECN]</td>
<td>Seq=0</td>
<td>Ack=1</td>
<td>Win=8192</td>
<td>Len=0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>54 63325</td>
<td>135 [ACK]</td>
<td>Seq=1</td>
<td>Ack=1</td>
<td>Win=2102272</td>
<td>Len=0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCE/RPC</td>
<td>214 Bind: call_id: 2, Fragment: Single, 3 context items</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCE/RPC</td>
<td>162 Bind ack: call id: 2, Fragment: Single, max xmit: 512</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EPM</td>
<td>222</td>
<td></td>
<td>Map request, TaskSchedulerService, 32bit NDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EPM</td>
<td>226</td>
<td></td>
<td>Map response, TaskSchedulerService, 32bit NDR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP</td>
<td>66 63326</td>
<td></td>
<td>135 [SYN, ECN, CWR]</td>
<td>Seq=0</td>
<td>Win=8192</td>
<td>Ack=1</td>
<td>Len=0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP</td>
<td>66 49666</td>
<td></td>
<td>49666 [SYN, ACK, ECN]</td>
<td>Seq=0</td>
<td>Ack=1</td>
<td>Win=8192</td>
<td>Len=0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP</td>
<td>54 63326</td>
<td></td>
<td>135 [ACK]</td>
<td>Seq=1</td>
<td>Ack=1</td>
<td>Win=2102272</td>
<td>Len=0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCE/RPC</td>
<td>262 Bind: call_id: 2, Fragment: Single, 3 context items</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCE/RPC</td>
<td>388 Bind ack: call_id: 2, Fragment: Single, max xmit: 512</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCE/RPC</td>
<td>594 AUTH3: call_id: 2, Fragment: Single, NTLMSSP_AUTH, U</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DCE/RPC Endpoint Mapper, Map
Operation: Map (3)
[Request in frame: 1071]
Handle: 00000000000000000000000000000000
Num Towers: 1

Tower array:
Max Count: 4
Offset: 0
Actual Count: 1

Tower pointer:
Referent ID: 0x0000000000000003
Length: 75
Number of floors: 5
Floor 1 UUID: TaskSchedulerService
Floor 2 UUID: 32bit NDR
Floor 3 RPC operation oriented protocol
Floor 4 TCP Port: 49666
Floor 5 IP: 172.17.0.20

Floor 4 TCP Port: 49666
## Task Scheduler Endpoint Resolution

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Dest IP</th>
<th>Protocol</th>
<th>Source Port</th>
<th>Dest Port</th>
<th>Method</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>66</td>
<td>63325</td>
<td>SYN, ECN, CWR</td>
<td>Seq=0 Win=8192 Len=0 Msqnd=1</td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>TCP</td>
<td>66</td>
<td>135</td>
<td>SYN, ACK, ECN</td>
<td>Seq=0 Ack=1 Win=8192 Len=0</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>54</td>
<td>63325</td>
<td>ACK</td>
<td>Seq=1 Ack=1 Win=2102272 Len=0</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCERPC</td>
<td>214</td>
<td></td>
<td>Bind</td>
<td>call_id: 2, Fragment: Single, 3 context items</td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>DCERPC</td>
<td>162</td>
<td></td>
<td>Bind_ack</td>
<td>call_id: 2, Fragment: Single, max_xmit: 50</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>EPM</td>
<td>222</td>
<td></td>
<td>Map request, TaskSchedulerService, 32bit NDR</td>
<td></td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>EPM</td>
<td>226</td>
<td></td>
<td>Map response, TaskSchedulerService, 32bit NDR</td>
<td></td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>66</td>
<td>63326</td>
<td>SYN, ECN, CWR</td>
<td>Seq=0 Win=8192 Len=0 Msqnd=1</td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>TCP</td>
<td>66</td>
<td>49666</td>
<td>SYN, ACK, ECN</td>
<td>Seq=0 Ack=1 Win=8192 Len=0</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>54</td>
<td>63326</td>
<td>ACK</td>
<td>Seq=1 Ack=1 Win=2102272 Len=0</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCERPC</td>
<td>262</td>
<td></td>
<td>Bind</td>
<td>call_id: 2, Fragment: Single, 3 context items</td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>DCERPC</td>
<td>388</td>
<td></td>
<td>Bind_ack</td>
<td>call_id: 2, Fragment: Single, max_xmit: 50</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCERPC</td>
<td>594</td>
<td></td>
<td>AUTH3</td>
<td>call_id: 2, Fragment: Single, NTLMSPP_AUTH, Signature, 79 bytes</td>
</tr>
</tbody>
</table>
## Task Scheduler Endpoint Resolution

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Destination IP</th>
<th>Protocol</th>
<th>Source Port</th>
<th>Destination Port</th>
<th>Duration</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>66</td>
<td>63325</td>
<td>6661</td>
<td>135</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>66</td>
<td>135</td>
<td>6661</td>
<td>63325</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>54</td>
<td>63325</td>
<td>6661</td>
<td>135</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCERPC</td>
<td>214</td>
<td>Bind: call_id: 2, Fragment: Single, 3 context items</td>
<td>6661</td>
<td>49666</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCERPC</td>
<td>162</td>
<td>Bind_ack: call_id: 2, Fragment: Single, max_xmit: 50</td>
<td>6661</td>
<td>49666</td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>EPM</td>
<td>222</td>
<td>Map request, TaskSchedulerService, 32bit NDR</td>
<td>6661</td>
<td>49666</td>
</tr>
<tr>
<td>172.17.0.20</td>
<td>172.17.0.61</td>
<td>EPM</td>
<td>222</td>
<td>Map response, TaskSchedulerService, 32bit NDR</td>
<td>6661</td>
<td>49666</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>66</td>
<td>63326</td>
<td>6661</td>
<td>49666</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>TCP</td>
<td>54</td>
<td>63326</td>
<td>6661</td>
<td>49666</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCERPC</td>
<td>262</td>
<td>Bind: call_id: 2, Fragment: Single, 3 context items</td>
<td>6661</td>
<td>49666</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCERPC</td>
<td>388</td>
<td>Bind_ack: call_id: 2, Fragment: Single, max_xmit: 50</td>
<td>6661</td>
<td>49666</td>
</tr>
<tr>
<td>172.17.0.61</td>
<td>172.17.0.20</td>
<td>DCERPC</td>
<td>594</td>
<td>AUTH3: call_id: 2, Fragment: Single, NTLMSSP_AUTH</td>
<td>6661</td>
<td>49666</td>
</tr>
</tbody>
</table>
Binding

● The representation of a session between a client and a server
  ○ Practically, a handle
  ○ Client and server can manipulate binding data using designated functions
  ○ Used for authentication (among other things)
An RPC Call’s Flow

Client

Foo(5, “hello”)

Server
An RPC Call’s Flow

Client

- Foo(5, “hello”)
- NdrClientCall3()
An RPC Call’s Flow

Client

- Foo(5, “hello”)
- NdrClientCall3()

Server

- Marshall parameters
- Connect to endpoint
- Bind to server
- Authenticate

RPC Runtime (rpcrt4.dll)
An RPC Call’s Flow

Client

- Foo(5, “hello”)
- NdrClientCall3()

- Marshall parameters
- Connect to endpoint
- Bind to server
- Authenticate

Server

- Listen on endpoint
- Unmarshall parameters
- Perform access checks

RPC Runtime (rpcrt4.dll)
**An RPC Call’s Flow**

**Client**
- Foo(5, “hello”)
- NdrClientCall3()

- Marshall parameters
- Connect to endpoint
- Bind to server
- Authenticate

**Server**
- Foo(5, “hello”)

- Listen on endpoint
- Unmarshall parameters
- Perform access checks

**RPC Runtime (rpcrt4.dll)**

---

© 2022 Akamai | Confidential
Zooming In

IDL:
void Foo([in] int number,
    [in] char* message);
Zooming In

**IDL:**
```plaintext
void Foo([in] int number,
        [in] char* message);
```

**Test c.c:**
```c
void Foo(
    handle_t IDL_handle,
    int number,
    unsigned char *message) {
    NdrClientCall3(
        (PMIDL_STUBLESS_PROXY_INFO
        )&Test_ProxyInfo, 0, 0,
        IDL_handle, number, message);
}
```
Zooming In

IDL:
void Foo([in] int number, [in] char* message);

Test c.c:
void Foo(
    handle_t IDL_handle,
    int number,
    unsigned char *message) {
    NdrClientCall3(
        (PMIDL_STUBLESS_PROXY_INFO )&Test_ProxyInfo, 0, 0, IDL_handle, number, message);
}

Client

Foo(5, “hello”)
Quick Recap

- Interface – describes server functionality
- Transport – the communication medium
- Endpoint – destination to connect to
- Binding – represents a client-server session

[UUID]
[protocol sequence]
[port, pipe name, etc.]
[binding handle]
MS-RPC (In-)Security
Agenda for this part

- MS-RPC built-in security mechanisms
- Security-related problems in MS-RPC
Security Mechanisms

● It’s a complete mess

● We’ll focus on remote communication and cover:
  ○ Authentication
  ○ Security descriptors
  ○ Security callback
Flags (🌈) – specified during interface registration

**RPC_STATUS** `RpcServerRegisterIf3`(
  `RPC_IF_HANDLE IfSpec`,
  `UUID *MgrTypeUuid`,
  `RPC_MGR_EPV *MgrEpv`
  `unsigned int Flags`,
  `unsigned int MaxCalls`,
  `unsigned int MaxRpcSize`,
  `RPC_IF_CALLBACK_FN *IfCallback`,
  `void *SecurityDescriptor`);

Transport Layer Authentication
SMB Authentication

- Named pipes are carried over SMB, requesting IPC$ share
SMB Authentication

- Named pipes are carried over SMB, requesting IPC$ share
- Authentication is on the SMB level
  → requires a valid user
SMB Authentication

- Named pipes are carried over SMB, requesting IPC$ share
- Authentication is on the SMB level
  → requires a valid user
- NULL sessions aren’t supported anymore
  → unless against DC:
    \pipe\netlogon, \pipe\samr, \pipe\lsarpc
Authenticated Binding
Authenticated Binding

- Binding that has authentication info
Authenticated Binding

- Binding that has authentication info
- Both server and client can set auth info using
  \texttt{RpcServerRegisterAuthInfo, RpcBindingSetAuthInfo}
Authenticated Binding

- Binding that has authentication info
- Both server and client can set auth info using `RpcServerRegisterAuthInfo`, `RpcBindingSetAuthInfo`
- Provides identity-based access control and other protections (e.g., Replay prevention, Integrity, Confidentiality) - specified by authentication level
Authenticated Binding

- RPC client and server exchange bind/bind_ack messages with authentication information
Authenticated Binding

- RPC client and server exchange bind/bind_ack messages with authentication information
- End result: a security context - a “security binding”
Authenticated Binding

- The client isn’t forced authenticate, even if the server registered authentication!
Authenticated Binding

- The client isn’t forced to authenticate, even if the server registered authentication!

https://csandker.io/2021/02/21/Offensive-Windows-IPC-2-RPC.html
Authenticated Binding

RPC_IF_ALLOW_SECURE_ONLY
Security Descriptors

```c
RPC_STATUS RpcServerRegisterIf3(
    RPC_IF_HANDLE IfSpec,
    UUID *MgrTypeUuid,
    RPC_MGR_EPV *MgrEpv
    unsigned int Flags,
    unsigned int MaxCalls,
    unsigned int MaxRpcSize,
    RPC_IF_CALLBACK_FN *IfCallback,
    void *SecurityDescriptor
);
```
Security Descriptors

- RPC servers can set security descriptors on both the endpoint and the interface.

```c
RPC_STATUS RpcServerRegisterIf3(
    RPC_IF_HANDLE IfSpec,
    UUID *MgrTypeUuid,
    RPC_MGR_EPV *MgrEpv
    unsigned int Flags,
    unsigned int MaxCalls,
    unsigned int MaxRpcSize,
    RPC_IF_CALLBACK_FN *IfCallback,
    void *SecurityDescriptor
);
```
PS C:\Users\defcon> ConvertFrom-SddlString

Owner : 
Group : 
DiscretionaryAcl : {Everyone: AccessAllowed (GenericRead), NT AUTHORITY\ANONYMOUS LOGON: AccessAllowed (GenericRead), NT AUTHORITY\RESTRICTED: AccessAllowed (GenericRead), NT AUTHORITY\SYSTEM: AccessAllowed (GenericAll)...}
SystemAcl : {} 

appidsvc.dll
RPC_STATUS RpcServerRegisterIf3(
    RPC_IF_HANDLE IfSpec,
    UUID        *MgrTypeUuid,
    RPC_MGR_EPV *MgrEpv
    unsigned int Flags,
    unsigned int MaxCalls,
    unsigned int MaxRpcSize,
    RPC_IF_CALLBACK_FN *IfCallback,
    void          *SecurityDescriptor
);
Security Callback

RPC Client → Security Callback → RPC Server

- RPC Client
- Security Callback
- Interface

RPC Server
Security Callback

RPC_STATUS RpcIfCallbackFn(
    RPC_IF_HANDLE InterfaceUuid,
    void *Context
)
{
    ...
}
Task Scheduler

RPC_STATUS RpcServer::SecurityCallback(RPC_IF_HANDLE InterfaceUuid, void *Context) {
    ...
    Status = RpcServerInqCallAttributesW(Context, &RpcCallAttributes);
    if ( !Status && RpcCallAttributes.AuthenticationLevel >= RPC_C_AUTHN_LEVEL_PKT_PRIVACY ) {
        if ( RpcCallAttributes.ProtocolSequence == RPC_PROTSEQ_LRPC ) {
            return RPC_S_OK;
        }
    } else if ( UuidEqual(&RpcCallAttributes.InterfaceUuid, &GUID_ITaskSchedulerService, &Status) && !Status ) {
        ...
    }
    return RPC_S_ACCESS_DENIED;
}
IAS (Internet Authentication Service)

RPC_STATUS CIasRpcServer::RpcIfSecurityCallback(RPC_IF_HANDLE InterfaceUuid, void *Context) {
    ...  
    if (I_RpcBindingIsClientLocal(0i64, &ClientLocalFlag) && ClientLocalFlag ) {
        if ( RpcBindingInqAuthClientW(Context, 0i64, 0i64, &AuthnLevel, 0i64, 0i64) && AuthnLevel >= RPC_C_AUTHN_LEVEL_PKT_PRIVACY && CIasRpcServer::IsCorrectProtseq(&hBinding) && CIasRpcServer::IsAccessGranted(v3, &hBinding) ) 
            {  
                return RPC_S_OK;
            }
    }
    return RPC_S_ACCESS_DENIED;
}
RPC_STATUS DhcpRpcCallback(RPC_IF_HANDLE InterfaceUuid, void *Context) {
    shouldPass = 0;
    if (!RpcBindingToStringBindingW(Context, &StringBinding)
        && !RpcStringBindingParseW(StringBinding, 0i64, &Protseq, 0i64, 0i64, 0i64)
        && !_wcsicmp(Protseq, L"ncalrpc") ) {
        shouldPass = 1;
    }
    if ( Protseq ) RpcStringFreeW(&Protseq);
    if ( StringBinding ) RpcStringFreeW(&StringBinding);
    if ( shouldPass ) return RPC_S_OK;
    else
        return RPC_S_ACCESS_DENIED;
}
LSASS

RPC_STATUS LsaRpcIfCallbackFn(RPC_IF_HANDLE InterfaceUuid, void *Context) {
    ...
    LastError = RpcServerInqCallAttributesW(a2, &RpcCallAttributes);
    ...
    if (RpcCallAttributes.OpNum >= 0x86u) return RPC_S_PROCNUM_OUT_OF_RANGE;
    ...
    v6 = (*((DWORD *)&LsapRPCFunctionProperties + 2 * RpcCallAttributes.OpNum);
    if (!_bittest(&v6, RpcCallAttributes.ProtocolSequence))
        return RPC_S_PROTSEQ_NOT_SUPPORTED;
    ...
}
Relevant Flags

🚩 RPC_IF_ALLOW_CALLBACKS_WITH_NO_AUTH
What can go wrong?
Be Wary of Other RPC Endpoints Running in the Same Process

“Endpoint Multiplexing”

Server

Interface A (local)  Interface B (remote)

Client
“Endpoint Multiplexing”

Interface A (local) \ ALPC \ Interface B (remote)

Server

Client
“Endpoint Multiplexing”

Server

Interface A (local)

ALPC

Interface B (remote)

TCP

Client
“Endpoint Multiplexing”

Interface A (local)  
ALPC  
TCP  
Interface B (remote)  

Server  

Client
“Endpoint Multiplexing”

- **Server**
  - Interface A (local)
  - ALPC
  - Interface B (remote)
  - TCP

- **Client**
“Endpoint Multiplexing”

Interface A (local)  Interface B (remote)

ALPC  TCP

Server

Client
“Endpoint Multiplexing”
“Endpoint Multiplexing”

- **Interface A (local)**
- **Interface B (remote)**
- **ALPC**
- **TCP**
- **Security Descriptor**

Client
“Endpoint Multiplexing”

Why?

- Interfaces are not bound to endpoints!
“Endpoint Multiplexing”

Why?

● Interfaces are not bound to endpoints!

When?

● Service is hosted with other services in the same svchost process
Relevant Flags

🚩 RPC_IF_ALLOW_LOCALONLY
Security Callback Caching
Security Callback Caching

Server

Interface A

EP

Client

opnum 0  opnum 1  Interface A  opnum 2  opnum 3

opnum 0  opnum 1

opnum 2  opnum 3
Security Callback Caching
Security Callback Caching

Server

Interface A

EP

Client

Security Callback

opnum 0

opnum 1

opnum 2

opnum 3
Security Callback Caching

Server

Interface A

Client

Security Callback

EP

opnum 0

opnum 1

opnum 2

opnum 3

✔
Security Callback Caching

Client

<table>
<thead>
<tr>
<th>Interface A</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>opnum 0</td>
<td></td>
</tr>
<tr>
<td>opnum 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>opnum 2</td>
<td></td>
</tr>
<tr>
<td>opnum 3</td>
<td></td>
</tr>
</tbody>
</table>

Security Callback
Security Callback Caching

Server

- opnum 0
- opnum 1
- opnum 2
- opnum 3

Interface A

EP

Client

Security Callback

✘
Security Callback Caching

Server

Interface A

EP

Client

opnum 0

opnum 1

opnum 2

opnum 3

Security Callback
Security Callback Caching

- Security Callback
- Interface A
- Server
  - opnum 0
  - opnum 1
  - Interface A
  - opnum 2
  - opnum 3
- EP
- Cache
- Client
Security Callback Caching

When?

- Happens by default
- Relies on the context identifier of the security context
  - Binding not authenticated? no caching!
Relevant Flags

- RPC_IF_SEC_NO_CACHE
- RPC_IF_SEC_CACHE_PER_PROC
Quick Recap

- Authentication Bindings
- Security descriptors
- Security callbacks
- Endpoint “multiplexing”
- Security callback response caching
A 0-Day in the Server service

Bug, attack flow & demo
The Server Service (i.e. LanmanServer)

- Accessible through the `\pipe\srvsvc` named pipe
Server’s Security Callback

# Windows 10 19H2
if (((RpcCallAttributes.OpNum - 64) <= 5 && RpcCallAttributes.IsClientLocal != 1))
return ERROR_ACCESS_DENIED;

Accessible remotely

Accessible only locally!

opnum 63  opnum 64  opnum 65  opnum 69
Server’s Security Callback

# Windows 10 19H2
if (((RpcCallAttributes.OpNum - 64) <= 5 && RpcCallAttributes.IsClientLocal != 1))
    return ERROR_ACCESS_DENIED;

Theory - newly added functions can pose a security problem in the future
Server’s Security Callback

# Windows 10 20H2
if ((RpcCallAttributes.OpNum - 64) <= 9 && RpcCallAttributes.IsClientLocal != 1))
return ERROR_ACCESS_DENIED;

💡
Theory - newly added functions can pose a security problem in the future

Accessible remotely

Accessible only locally!

<table>
<thead>
<tr>
<th>opnum 63</th>
<th>opnum 64</th>
<th>opnum 65</th>
<th>opnum 73</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Windows 11

```c
if ((RpcCallAttributes.OpNum - 64) <= 9 && RpcCallAttributes.IsClientLocal != 1))
    return ERROR_ACCESS_DENIED;
```

Theory - newly added functions can pose a security problem in the future
Server’s Security Callback

CVE-2022-30216 - Tampering (CVSS: 8.8)

# Windows 11
if ((RpcCallAttributes.OpNum - 64) <= 9 && RpcCallAttributes.IsClientLocal != 1))
   return ERROR_ACCESS_DENIED;

Accessible remotely

Accessible only locally!

opnum 63  opnum 64  opnum 65

opnum 73  opnum 74

⚠ LocalrServerCertificateMappingModify

Theory - newly added functions can pose a security problem in the future
SMB over QUIC
SMB over QUIC

- Transport layer protocol with low latency, privacy and security
**SMB over QUIC**

- Transport layer protocol with low latency, privacy and security
- Server provides a certificate - prevents server spoofing attacks
SMB over QUIC

- Transport layer protocol with low latency, privacy and security
- Server provides a certificate - prevents server spoofing attacks
- New functions added - manage the “symbolic link” of a QUIC certificate to a certificate in the certificate store
  - LocalrServerCertificateMappingGet
  - LocalrServerCertificateMappingSet
  - LocalrServerCertificateMappingEnum
  - LocalrServerCertificateMappingRemove
  - LocalrServerCertificateMappingModify
CVE-2022-30216

- Tampering - we can change a certificate mapping
- Maybe we can do more?
**CVE-2022-30216**

- Tampering - we can change a certificate mapping
- Maybe we can do more?

```assembly
00000000 certificateStruct struc ; (sizeof=0x58, mappedto_93)
00000000 serverName   dq ?     ; offset
00000008 subject     dq ?     ; offset
00000010 issuer      dq ?     ; offset
00000018 thumbprint  dq ?     ; offset
00000020 friendlyName dq ?     ; offset
00000028 notBefore   dq ?     ; offset
00000030 notAfter    dq ?     ; offset
00000038 storeLocation dq ?    ; offset
00000040 storeName   dq ?     ; offset
00000048 field_48    dq ?     ; offset
00000050 type        dd ?     ; offset
00000054 flags       dd ?     ; offset
00000058 certificateStruct ends
```
CVE-2022-30216

- Tampering - we can change a certificate mapping
- Maybe we can do more?

```c
00000000  certificateStruct    struc ; (sizeof=0x58, mappedto_93)
00000000  serverName           dq ? ; offset
00000008  subject              dq ? ; offset
00000010  issuer                dq ? ; offset
00000018  thumbprint            dq ? ; offset
00000020  friendlyName          dq ? ; offset
00000028  notBefore             dq ? ; offset
00000030  notAfter              dq ? ; offset
00000038  storeLocation         dq ? ; offset
00000040  storeName             dq ? ; offset
00000048  field_48              dq ? ; offset
00000050  type                  dd ?
00000054  flags                 dd ?
00000058  certificateStruct     ends
```
LocalrServerCertificateMappingModify() → RPC Server
LocalrServerCertificateMappingModify()
Here's my authentication info

LocalrServerCertificateMappingModify()
LocalrServerCertificateMappingModify()

Here's my authentication info!

Hi, I'm the RPC server – here's my authentication info!

Attacker Machine

RPC Server

Network Machine
LocalrServerCertificateMappingModify()

Hi, I’m the **Domain Controller** — here’s my authentication info!

© Will Schroeder & Lee Christensen
Hi, I'm the **Domain Controller** – here's my authentication info!

© Will Schroeder & Lee Christensen

```
LocalrServerCertificateMappingModify()
```

Here's the certificate you requested!
## Exploitation Flow

<table>
<thead>
<tr>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth coerce by calling vulnerable RPC function</td>
<td>Victims sends out credentials</td>
</tr>
</tbody>
</table>
## Exploitation Flow

<table>
<thead>
<tr>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth coerce by calling vulnerable RPC function</td>
<td>Victims sends out credentials</td>
</tr>
<tr>
<td>Relay credentials to ADCS</td>
<td>ADCS outputs a certificate</td>
</tr>
</tbody>
</table>
## Exploitation Flow

<table>
<thead>
<tr>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth coerce by calling vulnerable RPC function</td>
<td>Victims sends out credentials</td>
</tr>
<tr>
<td>Relay credentials to ADCS</td>
<td>ADCS outputs a certificate</td>
</tr>
<tr>
<td>Use Rubeus with certificate</td>
<td>Computer Kerberos TGT is granted</td>
</tr>
</tbody>
</table>
## Exploitation Flow

<table>
<thead>
<tr>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth coerce by calling vulnerable RPC function</td>
<td>Victims sends out credentials</td>
</tr>
<tr>
<td>Relay credentials to ADCS</td>
<td>ADCS outputs a certificate</td>
</tr>
<tr>
<td>Use Rubeus with certificate</td>
<td>Computer Kerberos TGT is granted</td>
</tr>
<tr>
<td>Perform DCSync</td>
<td>NTLM hash is obtained</td>
</tr>
</tbody>
</table>
## Exploitation Flow

<table>
<thead>
<tr>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth coerce by calling vulnerable RPC function</td>
<td>Victims sends out credentials</td>
</tr>
<tr>
<td>Relay credentials to ADCS</td>
<td>ADCS outputs a certificate</td>
</tr>
<tr>
<td>Use Rubeus with certificate</td>
<td>Computer Kerberos TGT is granted</td>
</tr>
<tr>
<td>Perform DCSync</td>
<td>NTLM hash is obtained</td>
</tr>
<tr>
<td>Pass the hash</td>
<td>Get shell</td>
</tr>
</tbody>
</table>
Exploit Demo
C:\Users\Administrator>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

Connection-specific DNS Suffix : 
Link-local IPv6 Address : fe80::6085:c673:f1e2:de25%9
IPv4 Address : 10.0.200.132
Subnet Mask : 255.255.0.0
Default Gateway : 10.0.0.138

C:\Users\Administrator>whoami
research\administrator

C:\Users\Administrator>
Summary

● Security callbacks are an interesting attack surface
  ○ Specifically dealing with opnums
  ○ Specifically due to caching

● Future research directions
  ○ More services, SMB over QUIC, RPC runtime, tooling

● Blog post & PoC available at https://akamai.com/blog/security/
References

- Offensive Windows IPC Internals 2: RPC (0xcsandker)
- How to secure a Windows RPC Server, and how not to (@tiraniddo)
- ADCS + PetitPotam NTLM Relay: Obtaining krbtgt Hash with Domain Controller Machine Certificate
Thank you

Questions?

@nachoskrnl  @OphirHarpaz