PowerPool malware exploits ALPC LPE zero-day vulnerability

By Matthieu Faou

Archived: 2026-04-05 20:46:22 UTC

Update (11 September 2018): [Microsoft has provided a patch](#) for this in today's Windows Update.

On August 27, 2018, a so-called *zero-day* vulnerability affecting Microsoft Windows was published on GitHub and publicized via a rather acerbic tweet.

Source: Twitter

It seems obvious that this was not part of a coordinated vulnerability disclosure and there was no patch at the time this tweet (since deleted) was published to fix the vulnerability.

It affects Microsoft Windows OSes from Windows 7 to Windows 10, and in particular the Advanced Local Procedure Call (ALPC) function, and allows a Local Privilege Escalation (LPE). LPE allows an executable or

process to escalate privileges. In that specific case, it allows an executable launched by a restricted user to gain administrative rights.

The tweet linked to a GitHub repository that contains Proof-of-Concept code for the exploit. Not only was a compiled version released – the source code was also. Consequently, anyone can modify and recompile the exploit, in order to "improve it", evade detection, or even incorporate it into their code.

As one could have predicted, it took only two days before we first identified the use of this exploit in a malicious campaign from a group we have dubbed PowerPool. This group has a small number of victims and according to both our telemetry and uploads to VirusTotal (we only considered manual uploads from the web interface), the targeted countries include Chile, Germany, India, the Philippines, Poland, Russia, the United Kingdom, the United States and Ukraine.

PowerPool arsenal

This newly isolated group already has quite a range of tools at its disposal. We will provide brief analyses of some of them here.

ALPC Local Privilege Escalation exploit

The PowerPool developers did not reuse the binary that was provided by the vulnerability's discloser. They modified the source code slightly and recompiled it.

The exploit has been documented by its original author and has been covered by security researchers and CERTs.

```
//************************************************************//
// Windows LPE - Non-admin/Guest to system - by SandboxEscaper   //
//************************************************************//

/* _SchRpcSetSecurity which is part of the task scheduler ALPC endpoint allows us to set an arbitrary DACL.
It will Set the security of a file in c:\windows\tasks without impersonating, a non-admin (works from Guest too) user can write here.
Before the task scheduler writes the DACL we can create a hard link to any file we have read access over.
This will result in an arbitrary DACL write.
This PoC will overwrite a printer related dll and use it as a hijacking vector. This is ofcourse one of many options to abuse this.*/
```

Figure 1 - Description of the exploit by its author

The flaw is in the SchRpcSetSecurity API function, which does not check the user's permissions correctly. Thus, a user can have write permissions on any file in C:\Windows\Task regardless of its actual permissions. That allows a user with only read permissions to replace the content of a write-protected file.

As any user is able to write in C:\Windows\Task, it is possible to create a file in this folder that is a hard link to any *target* file. Then, by calling the broken function SchRpcSetSecurity, it is possible to gain write access to that *target* file. To create a Local Privilege Escalation, the attacker needs to choose the *target* file that will be overwritten. This needs to be done carefully: it needs to be a file that is executed automatically with administrative rights. For example, it can be a system file, or the updater of previously installed software that is regularly executed by a task. The final step is to replace the content of this protected *target* file with malicious code. Thus, at the next automatic execution, the malware will have administrative rights regardless of its original rights.

PowerPool's developers chose to change the content of the file C:\Program Files (x86)\Google\Update\GoogleUpdate.exe. This is the legitimate updater for Google applications and is regularly run under administrative privileges by a Microsoft Windows task.

```
qmemcpy(&google_update_path, L"C:\\Program Files (x86)\\Google\\Update\\GoogleUpdate.exe", 0x6Cui64);
memset(&v5, 0, 0x19Cui64);
Hardlink::_CreateNativeHardlink((__int64)L"c:\\windows\\tasks\\UpdateTask.job", (__int64)&google_update_path);
F_to_RunExploit();
```

Figure 2 - Creation of a hardlink to the Google Updater

```
v4 = CreateBindingHandle((__int64)&v3);
SchRpcCreateFolder(
  v3,
  (__int64)L"UpdateTask",
  (__int64)L"D:(A;;FA;;;BA)(A;OICIIO;GA;;;BA)(A;;FA;;;SY)(A;OICIIO;GA;;;SY)(A;;0x1301bf;;;AU)(A;OICIIO;SDGXGWGR;;;AU)(A;"
           ";0x1200a9;;;BU)(A;OICIIO;GXGR;;;BU)",
  0i64);
SchRpcSetSecurity(
  v3,
  (__int64)L"UpdateTask",
  (__int64)L"D:(A;;FA;;;BA)(A;OICIIO;GA;;;BA)(A;;FA;;;SY)(A;OICIIO;GA;;;SY)(A;;0x1301bf;;;AU)(A;OICIIO;SDGXGWGR;;;AU)(A;"
           ";0x1200a9;;;BU)(A;OICIIO;GXGR;;;BU)",
  0i64);
```

Figure 3 - Abuse of SchRpcCreateFolder to change the permissions of the Google Updater executable

The sequence of operations shown in the figure above allows the PowerPool operators to gain write access to the executable GoogleUpdate.exe. Then, they overwrite it with a copy of their second-stage malware, described below, in order to gain SYSTEM privileges the next time the updater is called.

Initial compromise

The PowerPool group uses different approaches to initially compromise a victim. One is to send emails with their first-stage malware as an attachment. It may be too early to say, but to date we've seen very few occurrences in our telemetry, so we assume that the recipients are carefully chosen rather than PowerPool running a massive spam campaign.

On the other hand, we know that their spams have been spotted in the past. According to a SANS blogpost released in May 2018, they used a trick with Symbolic Link (.slk) files to distribute their malware. Microsoft Excel can load these files that update a cell and force Excel to execute PowerShell code. These .slk files seem to be distributed in spam messages too. Pivoting from the first file mentioned in the SANS blogpost (SHA-1: b2dc703d3af1d015f4d53b6dbbeb624f5ade5553), on VirusTotal, it is possible to find the related spam sample (SHA-1: e0882e234cba94b5cf3df2c05949e2e228bedd2b):

```
Received: from 71.177.222.4 by s214a.ik2.com (IK2 SMTP Server); Mon, 21 May 2018 01:26:45 +0000
Received: from TCXSERVE ([127.0.0.1]) by TCXSERVE with Microsoft SMTPSVC(7.0.6002.18264);
         Sun, 20 May 2018 18:25:32 -0700
From: "Gabriel" <b38094e38@def86.ca>
Subject: Invoice 287718 unpaid
To: "domains" <e4e46de@220e63.com>
Content-Type: multipart/mixed; boundary="smOkAqnCHdKkRalmHnuTyThAW2L1av=_1K"
MIME-Version: 1.0
Date: Sun, 20 May 2018 18:25:32 -0700
Message-ID: <TCXSERVEFN03a5sqMaS000016dd@TCXSERVE>
X-OriginalArrivalTime: 21 May 2018 01:25:32.0222 (UTC) FILETIME=[9C576DE0:01D3F0A2]
X-SF-RX-Return-Path: <b38094e38@def86.ca>
X-SF-HELO-Domain: TCXSERVE
X-SF-Originating-IP: 71.177.222.4
X-Rejection-Reason: 12 - 521 The IP 71.177.222.4 is Blacklisted by zen.ik2. ttps://www.spamhaus.org/sbl/query/SBLCSS ---  ---

This is a multi-part message in MIME format

--smOkAqnCHdKkRalmHnuTyThAW2L1av=_1K
Content-Type: multipart/alternative;
        boundary="4RsdsodQXqZzpNJLJaAF5KMTXqCy=_ZH1Z"

--4RsdsodQXqZzpNJLJaAF5KMTXqCy=_ZH1Z
Content-Type: text/plain; charset="utf-8"
Content-Transfer-Encoding: quoted-printable
Content-Disposition: inline

=EF=BB=BF=20
You have not settled this Invoice
=20
Regards.


--4RsdsodQXqZzpNJLJaAF5KMTXqCy=_ZH1Z
Content-Type: text/html; charset="utf-8"
Content-Transfer-Encoding: quoted-printable
Content-Disposition: inline

=EF=BB=BF<HTML><HEAD></HEAD>
<BODY>
<P> </P>
<P>You have not settled this Invoice</P>
<P> </P>
<P>Regards.</P></BODY></HTML>


--4RsdsodQXqZzpNJLJaAF5KMTXqCy=_ZH1Z--

--smOkAqnCHdKkRalmHnuTyThAW2L1av=_1K
Content-Type: application/octet-stream;
        name="Payment_Invoice#287718.slk"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
        filename="Payment_Invoice#287718.slk"
```

Figure 4 - PowerPool spam message

Windows backdoors

The PowerPool group mainly uses two different backdoors: a first-stage backdoor used just after the first compromise and then a second-stage backdoor, probably on the most interesting machines.

First-stage backdoor

This is basic malware used for reconnaissance on the machine. It comprises two Windows executables.

The first of these is the main backdoor. It establishes persistence through a service. It also creates a mutex named MyDemonMutex%d where %d ranges from 0 to 10. It is able to collect proxy information and the address of the C&C server is hardcoded into this binary. It can execute commands and performs some basic reconnaissance of the machine, which is then exfiltrated to the C&C server.

```
v11 = WinHttpGetIEProxyConfigForCurrentUser(&pProxyConfig);
if ( v11 )
{
  if ( pProxyConfig.lpszProxy )
  {
    v3.lpszProxy = pProxyConfig.lpszProxy;
    v3.dwAccessType = 3;
    v3.lpszProxyBypass = 0;
  }
  else if ( pProxyConfig.lpszAutoConfigUrl )
  {
    pAutoProxyOptions.dwFlags = 2;
    pAutoProxyOptions.lpszAutoConfigUrl = pProxyConfig.lpszAutoConfigUrl;
    pAutoProxyOptions.dwAutoDetectFlags = 0;
    pAutoProxyOptions.fAutoLogonIfChallenged = 1;
    pAutoProxyOptions.lpvReserved = 0;
    pAutoProxyOptions.dwReserved = 0;
    if ( WinHttpGetProxyForUrl(hSession, bing_dot_com, &pAutoProxyOptions, &pProxyInfo) )
      v3 = pProxyInfo;
    else
      (v6)(-1);
  }
```

Figure 5 - Gathering of proxy information

The second of these executables has a single purpose. It takes a screenshot of the victim's display and writes it in MyScreen.jpg. This file can then be exfiltrated by the main backdoor.

Second-stage backdoor

This malware is downloaded via the first stage, presumably when the operators believe the machine is interesting enough for them to stay on it for a longer time. However, it is clearly not a state-of-the-art APT backdoor.

Again, the C&C server address is hardcoded in the binary, and has no mechanism to update this crucial configuration item. This backdoor seeks commands from http://[C&C domain]/cmdpool and downloads additional files from http://[C&C domain]/upload. These additional files are mainly the lateral-movement tools mentioned below.

The supported commands are:

- Execute a command
- Kill a process
- Upload a file
- Download a file
- List a folder

They are sent in JSON format. The examples below are instructions to execute a command and to list a folder:

```
{"dos":{"cmd":"arp -a"}}
{"folder":{"path":"C:\\Users\\[redacted]\\AppData\\Local\\*.*"}}
```

Figure 6 - Examples of backdoor commands

Lateral movement tools

Once the PowerPool operators have persistent access to a machine with the second-stage backdoor, they use several open-source tools, mostly written in PowerShell, to move laterally on the network.

- **PowerDump**: This is a Metasploit module that can retrieve usernames and hashes from the Security Account Manager (SAM).
- **PowerSploit**: This is a post-exploitation framework in PowerShell, à la Metasploit.
- **SMBExec**: A PowerShell tool to perform pass-the-hash SMB connections.
- **Quarks PwDump**: A Windows executable that can retrieve Windows credentials.
- **FireMaster**: A Windows executable that can retrieve stored passwords from Outlook, web browsers, etc.

Conclusion

The disclosure of vulnerabilities outside of a coordinated disclosure process generally puts many users at risk. In this case, even the most up-to-date version of Windows could be compromised as no patch was released when the vulnerability and exploit were published. The CERT-CC provides some mitigations but Microsoft has not officially approved them.

This specific campaign targets a limited number of users, but don't be fooled by that: it shows that cybercriminals also follow the news and work on employing exploits as soon as they are publicly available.

ESET Researchers will continue tracking any malicious usage of this new vulnerability. Indicators of Compromise can also be found on GitHub. For any inquiries, or to make sample submissions related to the subject, contact us at threatintel@eset.com.

Update (11 September 2018): Microsoft has provided a patch for this in today's Windows Update.

Indicators of compromise

Hashes

SHA-1	Type	Detection name
038f75dcf1e5277565c68d57fa1f4f7b3005f3f3	First stage backdoor	Win32/Agent.SZS
247b542af23ad9c63697428c7b77348681aadc9a	First stage backdoor	Win32/Agent.TCH
0423672fe9201c325e33f296595fb70dcd81bcd9	Second stage backdoor	Win32/Agent.TIA
b4ec4837d07ff64e34947296e73732171d1c1586	Second stage backdoor	Win32/Agent.TIA
9dc173d4d4f74765b5fc1e1c9a2d188d5387beea	ALPC LPE exploit	Win64/Exploit.Agent.H

Detection names
Win32/Agent.SZS
Win32/Agent.TCH
Win32/Agent.TEL
Win32/Agent.THT
Win32/Agent.TDK
Win32/Agent.TIA
Win32/Agent.TID
C&C servers
newsrental[.]net
rosbusiness[.]eu
afishaonline[.]eu
sports-collectors[.]com
27.102.106[.]149

Source: https://www.welivesecurity.com/2018/09/05/powerpool-malware-exploits-zero-day-vulnerability/