blackarrowsec / redteam-research Public

Notifications Fork 186 Star 1.1k

Code Issues 1 Pull requests Actions Projects Security Insights

Files

26e6fc0

Go to file

- CVE-2018-1685
- CVE-2020-1472
- EDR_AV Bypass
- Hanshell
- LPE via StorSvc
  - RpcClient
  - SprintCSP
  - FactoryResetUICC.png
  - PoC.gif
  - README.md
- README.md
- logo.png

redteam-research / LPE via StorSvc /

Kudaes Update README.md 26e6fc0 · last year History

Name	Last commit message	Last commit date
..		
RpcClient	Added support for windows server 2022	last year
SprintCSP	LPE via StorSvc	last year
FactoryResetUICC.png	LPE via StorSvc	last year
PoC.gif	LPE via StorSvc	last year
README.md	Update README.md	last year

README.md

LPE via StorSvc

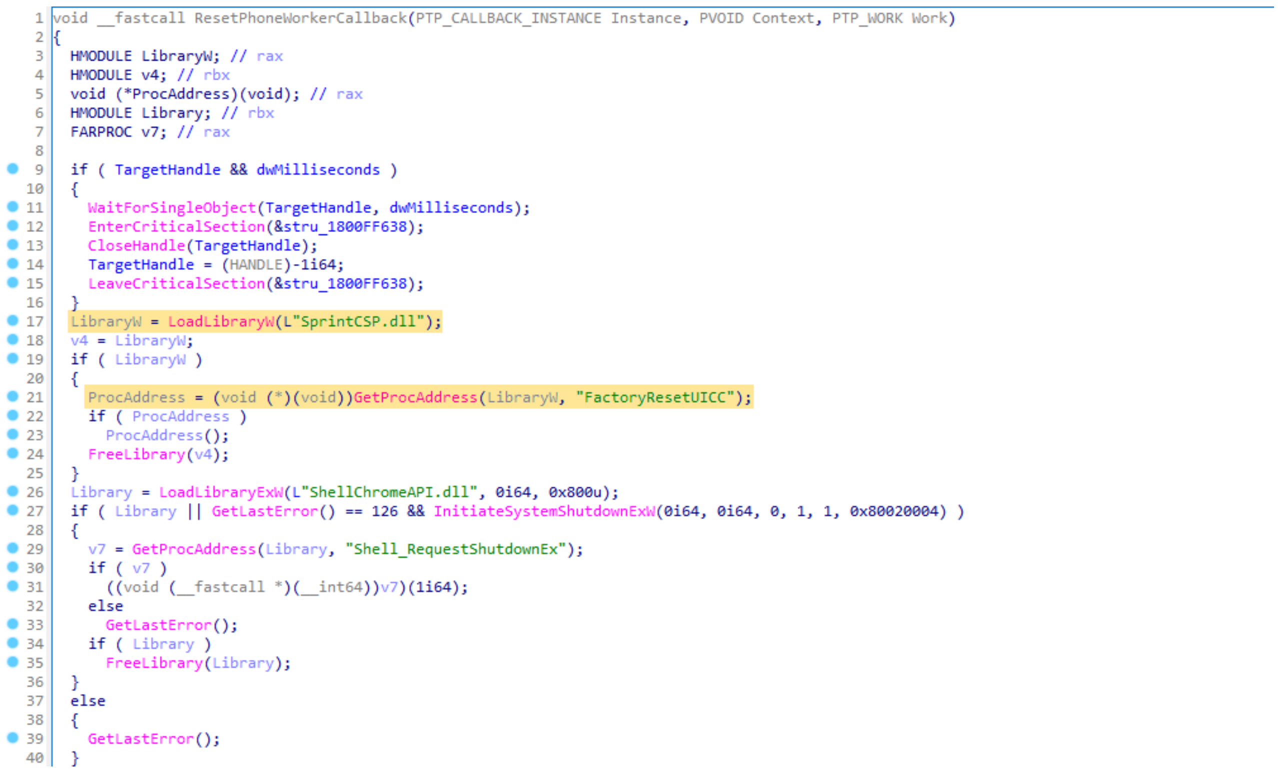
Windows Local Privilege Escalation via StorSvc service (writable SYSTEM path DLL Hijacking)

Summary

StorSvc is a service which runs as `NT AUTHORITY\SYSTEM` and tries to load the missing **SprintCSP.dll** DLL when triggering the `SvcRebootToFlashingMode` RPC method locally.

Description

The `StorSvc.dll!SvcRebootToFlashingMode` RPC method, calls `StorSvc.dll!InitResetPhone` which also calls `StorSvc.dll!ResetPhoneWorkerCallback`, that tries to load **SprintCSP.dll** as shown in the image below:



```
void __fastcall ResetPhoneWorkerCallback(PTP_CALLBACK_INSTANCE Instance, PVOID Context, PTP_WORK Work)
{
  HMODULE LibraryW; // rax
  HMODULE v4; // rbx
  void (*ProcAddress)(void); // rax
  HMODULE Library; // rbx
  FARPROC v7; // rax

  if ( TargetHandle && dwMilliseconds )
  {
    WaitForSingleObject(TargetHandle, dwMilliseconds);
    EnterCriticalSection(&stru_1800FF638);
    CloseHandle(TargetHandle);
    TargetHandle = (HANDLE)-1i64;
    LeaveCriticalSection(&stru_1800FF638);
  }
  LibraryW = LoadLibraryW(L"SprintCSP.dll");
  v4 = LibraryW;
  if ( LibraryW )
  {
    ProcAddress = (void (*)(void))GetProcAddress(LibraryW, "FactoryResetUICC");
    if ( ProcAddress )
      ProcAddress();
    FreeLibrary(v4);
  }
  Library = LoadLibraryExW(L"ShellChromeAPI.dll", 0i64, 0x800u);
  if ( Library || GetLastError() == 126 && InitiateSystemShutdownExW(0i64, 0i64, 0, 1, 1, 0x80020004) )
  {
    v7 = GetProcAddress(Library, "Shell_RequestShutdownEx");
    if ( v7 )
      ((void (__fastcall *)(__int64))v7)(1i64);
    else
      GetLastError();
    if ( Library )
      FreeLibrary(Library);
  }
  else
  {
    GetLastError();
  }
```

As this DLL is missing, it is loaded following the **DLL Search Order** flow and we can take advantage of this behaviour by placing a malicious DLL in a writable folder contained in the SYSTEM `%PATH%`. Then, the malicious DLL should be executed with **SYSTEM privileges**.

It is worth noting that the service is launched as `NT AUTHORITY\SYSTEM` in the service group `LocalSystemNetworkRestricted` which has the following privileges:

```
Privilege Name                Description
============================= =========================================
SeTcbPrivilege                Act as part of the operating system
SeLoadDriverPrivilege         Load and unload device drivers
SeBackupPrivilege             Back up files and directories
SeRestorePrivilege            Restore files and directories
SeSystemEnvironmentPrivilege Modify firmware environment values
SeChangeNotifyPrivilege       Bypass traverse checking
SeManageVolumePrivilege       Perform volume maintenance tasks
```

The command line that corresponds to this service is `C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted -p -s StorSvc`.

Proof of Concept

In this repo we provide 2 different source codes:

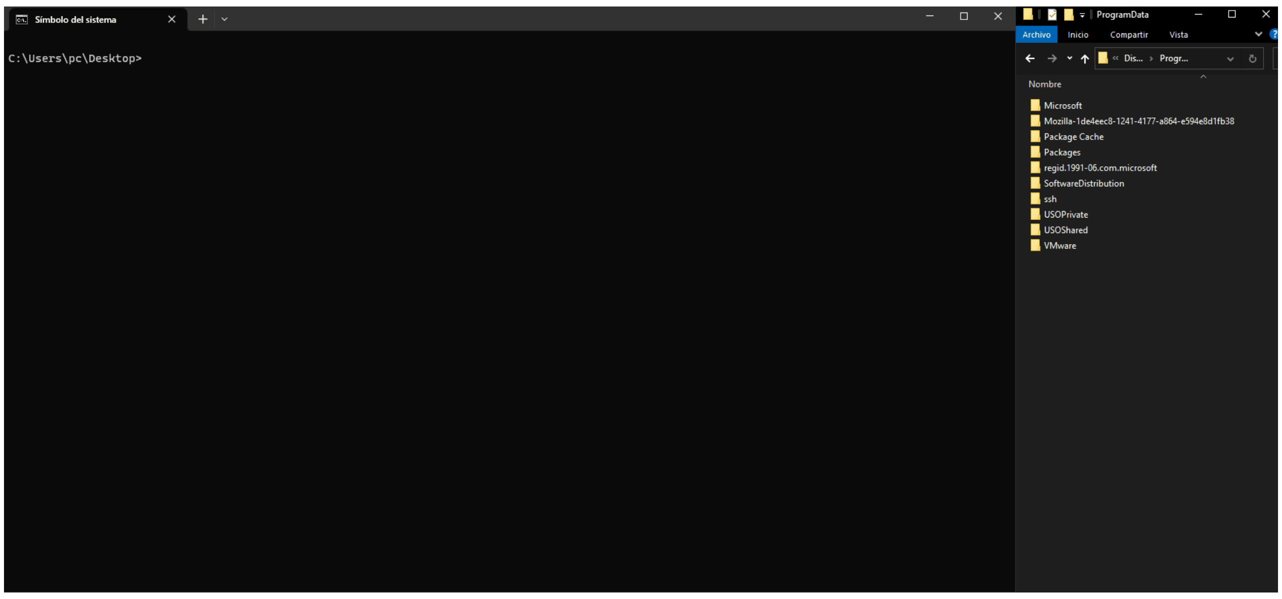
- RpcClient.exe: that triggers the RPC call.
- SprintCSP.dll: which can be placed to exploit the DLL Hijacking. This PoC runs a `whoami` command and writes the output to `C:\ProgramData\whoamiall.txt`. If you want to expand the functionality of this PoC you can edit the `DoStuff()` function at main.c.

The provided exploit should work by default and has been tested on **Windows 10**, ** Windows 11**, **Windows Server 2019** and **Windows Server 2022**. **In order to make it work, the `#define` macro at storsvc_c.c must be changed so the exploit is adapted to the target machine's operative system.**

After triggering the exploit it is necessary to **stop** or **reboot** the service, which SprintCSP.dll already does.

Steps

1. Find writable SYSTEM path with `reg query "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment" -v Path`
2. Copy SprintCSP.dll to the writable path
3. Execute RpcClient.exe
4. Check `C:\ProgramData\whoamiall.txt`



```
C:\Users\pc\Desktop>
```

ProgramData

Nombre

Microsoft
Mozilla-1de4eec8-1241-4177-a864-e594e8d1fb38
Package Cache
Packages
regid.1991-06.com.microsoft
SoftwareDistribution
ssh
USOPrivate
USOShared
VMware

References

- Fuzzing Windows RPC with RpcView

- CdpSvcLPE
- CDPSvc DLL Hijacking - From LOCAL SERVICE to SYSTEM

---

www blackarrow.net twitter @BlackArrowSec linkedin @BlackArrowSec