"Cracking Open the Malware Piñata" Series: Intro to Dynamic Analysis with RedLineStealer

atomicmatryoshka.com/post/cracking-open-the-malware-piñata-series-intro-to-dynamic-analysis-with-redlinestealer

z3r0day_504 January 2, 2022

- z3r0day_504
-
-
  - Jan 2
  -
  - 6 min read

Dynamic analysis involves running a binary and observing its behavior in a controlled environment. This can be of significant benefit because some capabilities of malware come to life only at runtime, meaning that the indicators and behaviors would not be observed if only analyzed statically.

In this iteration of my latest series, we dive into RedLineStealer. At the time of writing this blog post, MalwareBazaar shows RedLineStealer as the second most prevalent malware family in the last 14 days. The variant we're analyzing is relatively noisy, as you'll see in the content below.

If you're interested in viewing my previous posts in this series, please check them out here:

"Cracking Open the Malware Piñata" Series: Analysis Environment Setup

"Cracking Open the Malware Piñata" Series: Intro to Static Analysis with Kazy Trojan

Additionally, if you'd like to get right to the counter-hacking, scroll down to the "Pushing the Big Red Button" headling. :)

BEFORE THE BOOM

Prior to double-clicking on that badness-laden executable, there are a couple of things that need to be done to prepare the environment.

Network Configuration

First and foremost, we need to check the networking between virtual machines. Verify that the network adapters are set to host only. From there, go into REMnux and identify its IP.

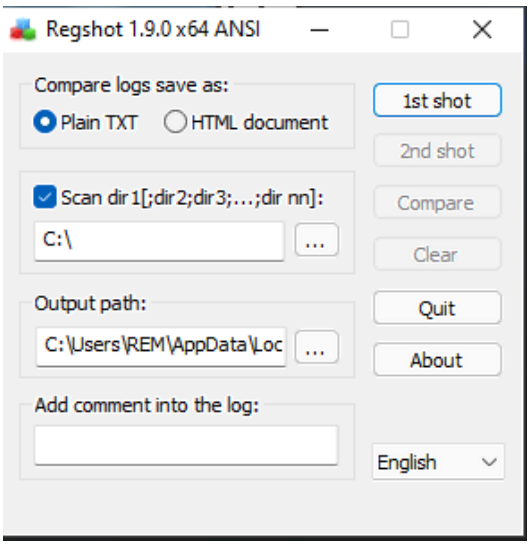
Making note of it, we return to the Windows VM and take the following steps:

1. Access the network adapter settings in Windows
2. Change the Windows IP to be on the same subnet as the REMnux IP
3. Input the REMnux IP as the default gateway and preferred DNS server in the Windows settings
4. Close all dialog boxes
5. Ping the REMnux VM from the Windows VM to make sure the connection is functional

This ensures that any time the Windows VM attempts to establish a network connection, the connection will be routed to the REMnux VM. This will be useful when it comes time to observing what the malware calls out to when detonating.

Registry Snapshot

Next, let's grab a snapshot of the registry. RegShot is a tool originally developed in 1999 by TiANWEi but contributed to by Maddes, XhmikosR, tulipfan, and Belogorokhov Youri amongst a large cohort of others over the course of two decades. The RegShot utility enables to user to grab "snapshots" of the registry prior to and after a specific event. The utility also has a "compare" feature, which highlights the differences between the "before" and "after" shots.



To grab our "before" shot, it's as simple as:

1. Double-click on the RegShot executable
2. Click on "1st Shot," then "Shot and Save"
3. Once complete, name your file and where to save it

VM Snapshot

After completing the registry baseline, it's a good time to take a snapshot of the VM. This will allow you to revert the Windows VM to this state after detonating the malware. I recommend taking the VM snapshot after taking the initial registry snapshot to avoid having to repeat the process again.

To take a VM snapshot in VMWare:

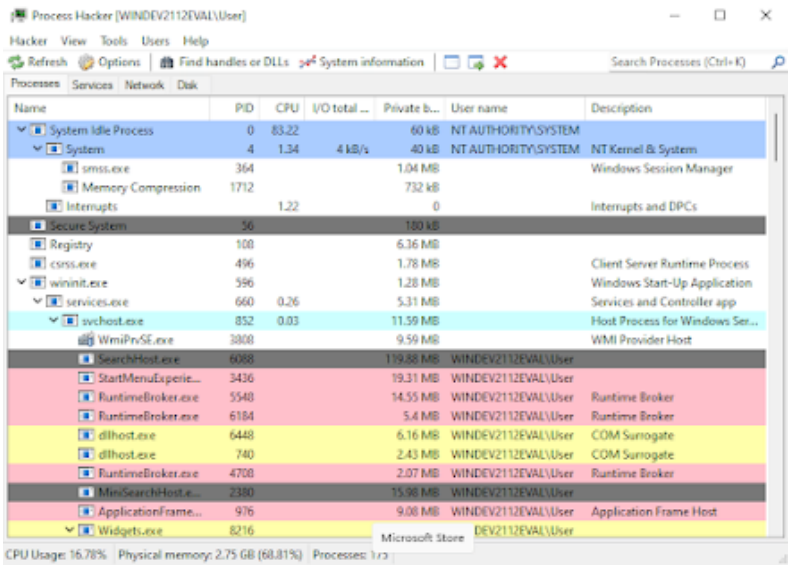
1. Go to the toolbar at the top of the window and click on "VM"
2. Hover over "Snapshot"
3. Click "Take Snapshot..."
4. Name it something intuitive and click "Take Snapshot"

PREPARE THE WATCH

The environment is now configured and baselines have been captured. At this stage, we start spinning up the tools that will actively be running when the malware is executed. It's important to have these tools active prior to initiating detonation so that we capture *all* events that take place.

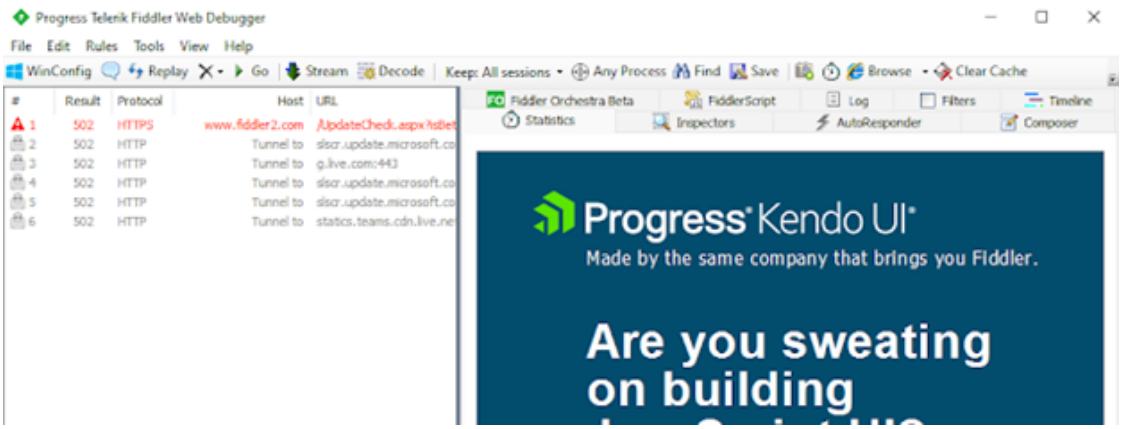
Process Hacker

Process Hacker is one of the first tools I spin up at this point. Developed by Wen Jia Liu with help from XhimkosR and a plethora of additional contributors, it is a "task manager on steroids." The GUI is intuitive with highlighting showing which processes are spawning and terminating, modifiable columns, and additional details visible with just the hover of a mouse.



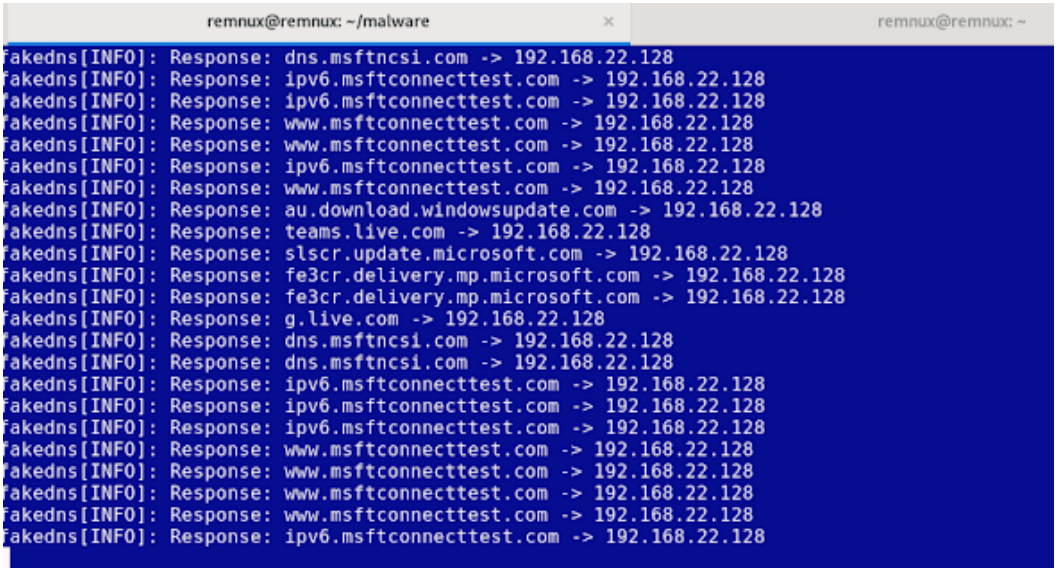
Fiddler Classic

Fiddler allows you to observe and inspect HTTP/S network requests. Developed by Telerik, this tool is one of the several tools available to observe the networking capabilities of the malware once detonated.



FakeDNS

Possible overkill, but I normally have this *and* Fiddler running concurrently. FakeDNS is a command-line tool available on REMnux and developed by Verisign's iDefense group. It captures all DNS requests sent to the device its running on and lists them in the terminal in real-time.

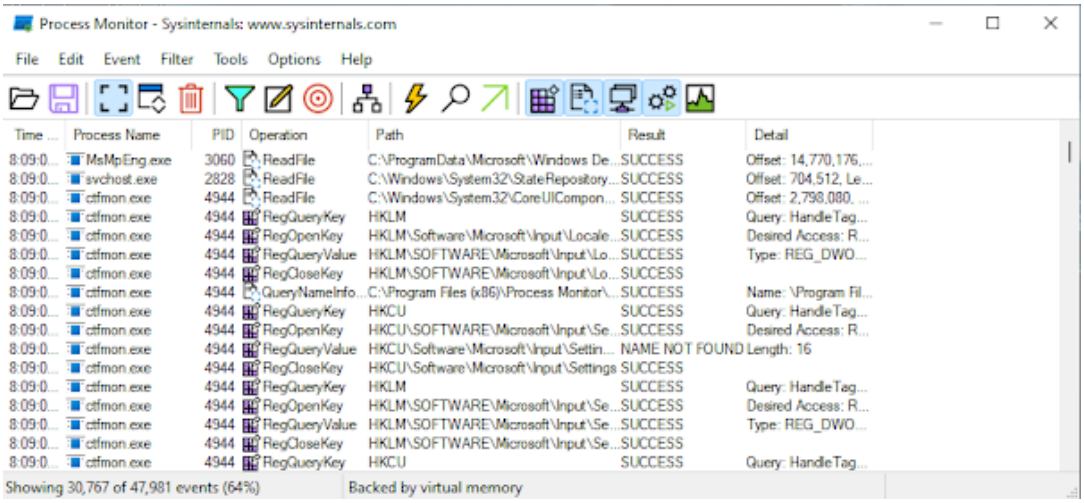


WireShark

While I won't be using it in this specific scenario, Wireshark is an incredibly useful packet capture tool. Packet inspection is a very useful technique, especially when characterizing communications between a piece of malware and the infrastructure it's "calling out" to. Dissecting this information can better help paint the picture of what information the malware is sending or receiving.

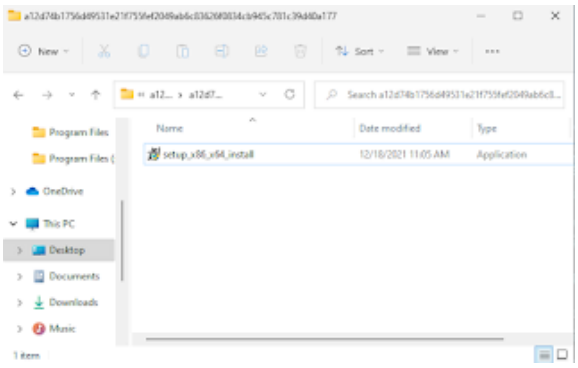
Process Monitor

Developed by Mark Russinovich as part of Windows Sysinternals, Process Monitor captures file, process, registry, and networking events all in real-time. In dynamic analysis, this is magnificent in that in catches a lot of artifacts that may no longer be present by the time an analyst goes digging for evidence. It is also very effective at painting a timeline of events.

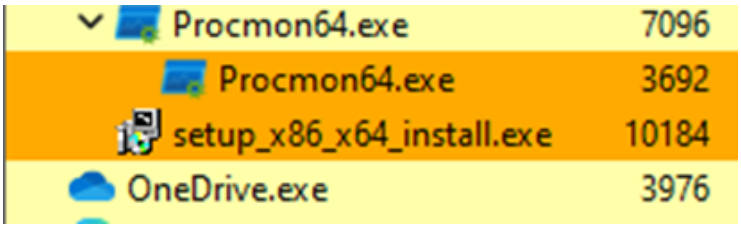


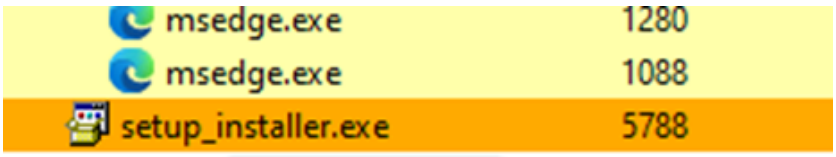
PUSHING THE BIG RED BUTTON

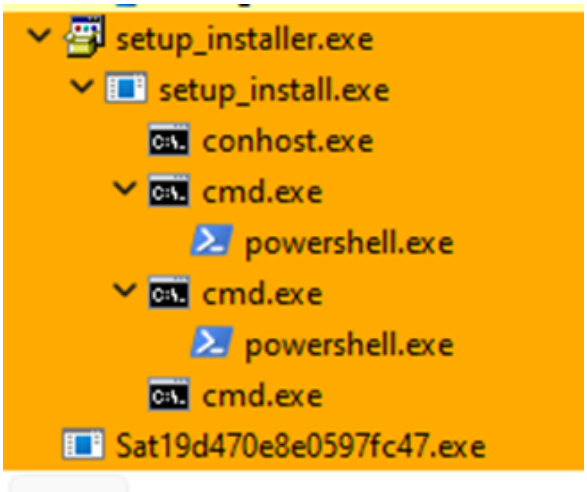
Now we're locked and loaded to let the badness run rampant. Here we'll start our analysis of RedLine Stealer with a sample I've pulled from MalwareBazaar. If you're not familiar, you can check them out here. IOCs identified through this process will be summarized at the bottom of the post.



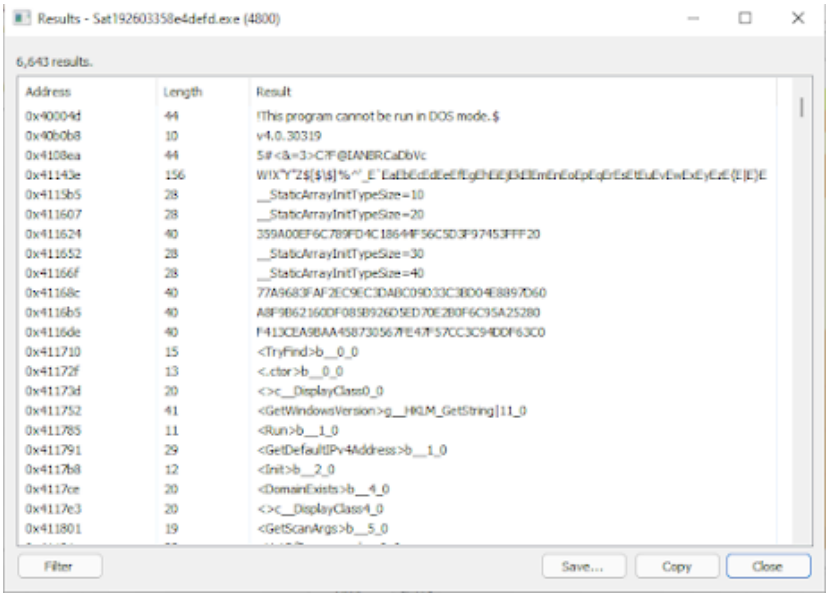
Right out of the gate, here's the evolution of Process Hacker after initial execution:







setup_x86_x64_install.exe spawns setup_installer.exe, which in turn spawns several instances of Sat19d470e8e0597fc47.exe. Process Hacker allows you to take a look at the memory strings during runtime. This can be done by right-clicking on the process of interest, clicking "Properties," navigating to the "Memory" tab, and clicking on "Strings..." Below are screenshots of what this looked like for Sat19d470e8e0597fc47.exe.



Address	Length	Result
0x412e13	40	B14D74C51EAE4F88FBF39B8BD07DA392799FCAAF
0x412e3c	40	78F285852D43939E0FBD786C5592189AF986E88F
0x412e65	40	3DB6DAD76E13B54DC03AF1C6092C40388E57FBBF
0x412e98	34	BCRYPT_INIT_AUTH_MODE_INFO_VERSION
0x412ed4	14	TryInitNordVPN
0x412ee3	14	TryInitOpenVPN

Results - Sat192603358e4defd.exe (4800)

977 results.

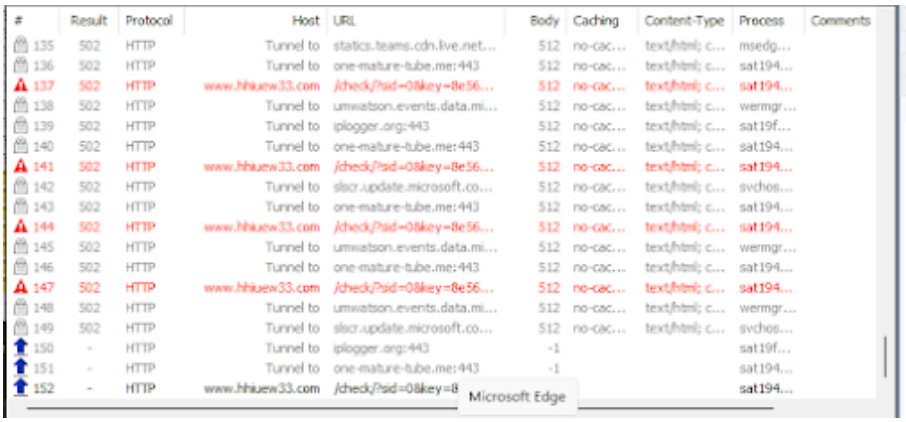
Address	Length	Result
0x2cccd84	74	http://tempuri.org/Entity/Id4Respo...
0x2ccd678	52	http://tempuri.org/Entity/
0x2ccd704	44	http://tempuri.org/Ent
0x2ccd748	58	http://tempuri.org/Entity/Id5
0x2ccd7ac	52	http://tempuri.org/Entity/
0x2ccd85c	44	http://tempuri.org/Ent
0x2ccd8a0	74	http://tempuri.org/Entity/Id5Respo...
0x2cce178	52	http://tempuri.org/Entity/
0x2cce204	44	http://tempuri.org/Ent
0x2cce248	58	http://tempuri.org/Entity/Id6
0x2cce2ac	52	http://tempuri.org/Entity/
0x2cce35c	44	http://tempuri.org/Ent
0x2cce3a0	74	http://tempuri.org/Entity/Id6Respo...
0x2cced38	52	http://tempuri.org/Entity/
0x2ccedc4	44	http://tempuri.org/Ent
0x2ccee08	58	http://tempuri.org/Entity/Id7

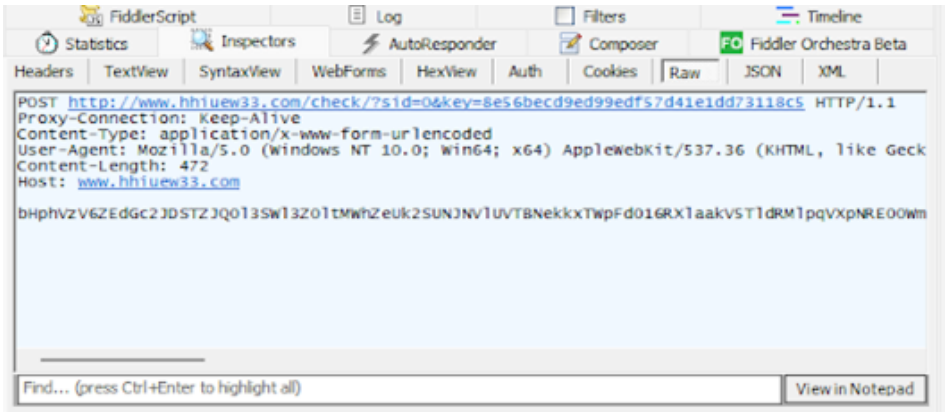
We see references to "tempuri[.]org" as well as references to "TryInitNordVPN" and "TryInitOpenVPN." Based on cursory research, the last two references don't appear to be existing open-source libraries/APIs, so these could've been developed by the malware author.

Taking a look over in fakeDNS, we have quite a few attempts to contact domains, aside from general Windows noise:

```
fakedns[INFO]: Response: www.msftconnecttest.com -> 192.168.22.128
fakedns[INFO]: Response: www.msftconnecttest.com -> 192.168.22.128
fakedns[INFO]: Response: wdcp.microsoft.com -> 192.168.22.128
fakedns[INFO]: Response: wdcpalt.microsoft.com -> 192.168.22.128
fakedns[INFO]: Response: 184.246.69.159.in-addr.arpa -> 192.168.22.128
fakedns[INFO]: Response: 45.30.193.212.in-addr.arpa -> 192.168.22.128
fakedns[INFO]: Response: dns.msftncsi.com -> 192.168.22.128
fakedns[INFO]: Response: dns.msftncsi.com -> 192.168.22.128
fakedns[INFO]: Response: dns.msftncsi.com -> 192.168.22.128
fakedns[INFO]: Response: one-mature-tube.me -> 192.168.22.128
fakedns[INFO]: Response: cloudjah.com -> 192.168.22.128
fakedns[INFO]: Response: tlu.dl.delivery.mp.microsoft.com -> 192.168.22.128
fakedns[INFO]: Response: 57.225.144.45.in-addr.arpa -> 192.168.22.128
fakedns[INFO]: Response: kelenxz.xyz -> 192.168.22.128
fakedns[INFO]: Response: ad-postback.biz -> 192.168.22.128
fakedns[INFO]: Response: www.listincode.com -> 192.168.22.128
fakedns[INFO]: Response: ip-api.com -> 192.168.22.128
fakedns[INFO]: Response: iplogger.org -> 192.168.22.128
fakedns[INFO]: Response: gp.gamebuy768.com -> 192.168.22.128
fakedns[INFO]: Response: www.hhiuew33.com -> 192.168.22.128
fakedns[INFO]: Response: umwatson.events.data.microsoft.com -> 192.168.22.128
fakedns[INFO]: Response: 168.69.108.65.in-addr.arpa -> 192.168.22.128
fakedns[INFO]: Response: teams.live.com -> 192.168.22.128
```

Over the course of time, the malware consistently attempted to call out to hhieuw33[.]com. We were able to take a closer look at that communication in Fiddler:





In the URL, the malware is attempting to pass some information in the parameters named "sid" and "key." Additionally, the content of the POST request is encoded in multiple layers, with the outermost layer being base64.

Moving to the RegShot comparison, output:

```
Files added: 171
C:\Program Files (x86)\Process Monitor\Logfile.CSV
C:\Program Files (x86)\FarLabUninstaller\unins000.dat
C:\Program Files (x86)\FarLabUninstaller\unins000.exe
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_@.cmd_edd8c564b2e33d8f5d03143c8eaf05d6e819754_e6c52f83_cab_872cedee-6936-4b
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_@.cmd_edd8c564b2e33d8f5d03143c8eaf05d6e819754_e6c52f83_cab_872cedee-6936-4b
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_@.cmd_edd8c564b2e33d8f5d03143c8eaf05d6e819754_e6c52f83_cab_872cedee-6936-4b
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_@.cmd_edd8c564b2e33d8f5d03143c8eaf05d6e819754_e6c52f83_cab_872cedee-6936-4b
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_@.cmd_edd8c564b2e33d8f5d03143c8eaf05d6e819754_e6c52f83_cab_872cedee-6936-4b
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_@.cmd_edd8c564b2e33d8f5d03143c8eaf05d6e819754_e6c52f83_cab_872cedee-6936-4b
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_@.cmd_edd8c564b2e33d8f5d03143c8eaf05d6e819754_e6c52f83_cab_872cedee-6936-4b
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_Sat1903c7a137841_c7f3a667854f329fb45264ead98727b1ad860de_7acd47b4_cab_a0eaa
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_Sat1903c7a137841_c7f3a667854f329fb45264ead98727b1ad860de_7acd47b4_cab_a0eaa
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_Sat1903c7a137841_c7f3a667854f329fb45264ead98727b1ad860de_7acd47b4_cab_a0eaa
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_Sat1903c7a137841_c7f3a667854f329fb45264ead98727b1ad860de_7acd47b4_cab_a0eaa
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_Sat1903c7a137841_c7f3a667854f329fb45264ead98727b1ad860de_7acd47b4_cab_a0eaa
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_Sat1903c7a137841_c7f3a667854f329fb45264ead98727b1ad860de_7acd47b4_cab_a0eaa
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_Sat1903c7a137841_c7f3a667854f329fb45264ead98727b1ad860de_7acd47b4_cab_a0eaa
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_Sat194837533b1da_427e4c8bbbcd7d84be70e2d2afa9f675bfe51810_17168784_cab_dc9b
```

~res-x64 - Notepad

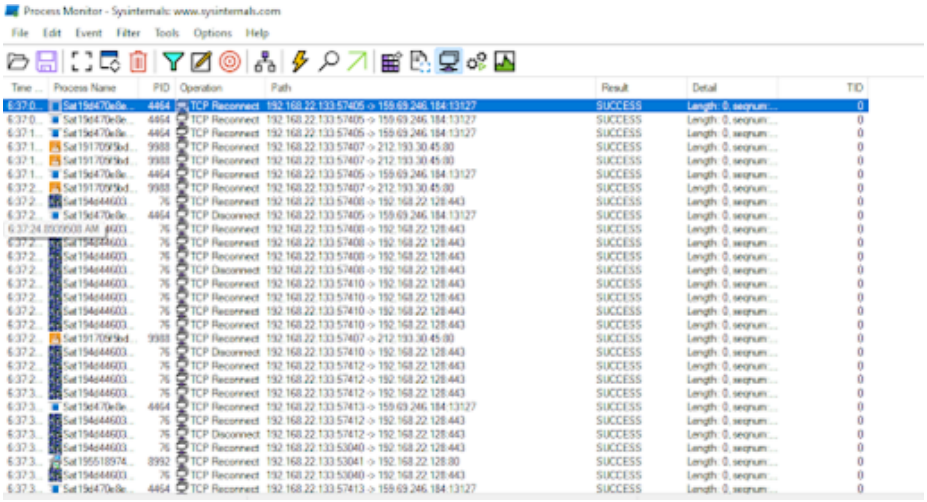
File Edit Format View Help

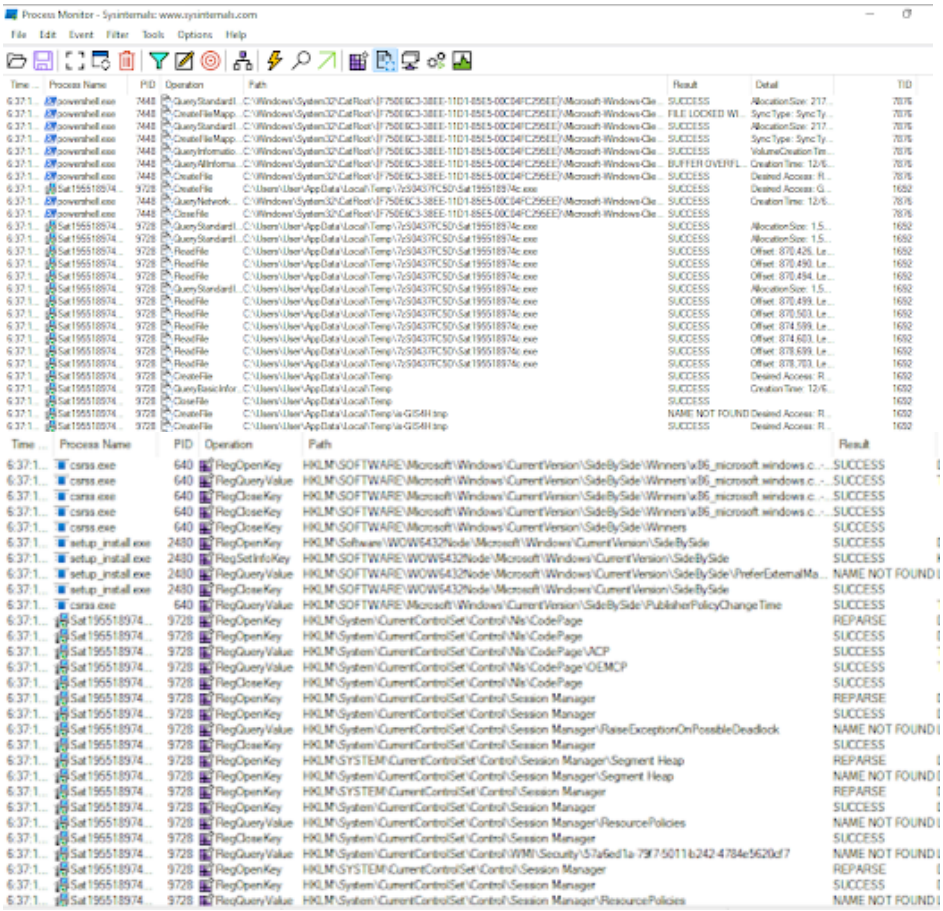
```
HKU\S-1-5-21-1516808570-3660347512-3657706960-1001\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\

---------------------------------
Keys added: 50
---------------------------------
HKLM\SOFTWARE\Microsoft\SystemCertificates\AuthRoot\Certificates\02FAF3E291435468607857694DF5E45B68851868
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASAPI32
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASMANCS
HKLM\SOFTWARE\Microsoft\Tracing\Sat19f1c04426464e86_RASAPI32
HKLM\SOFTWARE\Microsoft\Tracing\Sat19f1c04426464e86_RASMANCS
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\10084
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\240
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\4156
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\4364
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\8084
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\9592
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\9780
HKLM\SOFTWARE\WOW6432Node\Microsoft\IdentityCRL\ClockData
HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall\FarLabUninstaller.exe_is1
HKLM\SOFTWARE\WOW6432Node\Microsoft\SystemCertificates\AuthRoot\Certificates\02FAF3E291435468607857694DF5E45B68851868
HKLM\SYSTEM\ControlSet001\Control\Class\{3a1380f4-708f-49de-b2ef-04d25eb009d5}
HKLM\SYSTEM\ControlSet001\Services\PROCMON24
HKLM\SYSTEM\ControlSet001\Services\PROCMON24\Instances
```

```
---------------------------------
Values added: 152
---------------------------------
HKLM\SOFTWARE\Microsoft\FTH\CheckPointTime: 0x0D32BA0E
HKLM\SOFTWARE\Microsoft\SystemCertificates\AuthRoot\Certificates\02FAF3E291435468607857694DF5E45B68851868\Blob:  19 00 00 00 01 00 00 00 1
 10 06 0A 2B 06 01 04 01 82 37 3C 01 01 03 02 00 C0 09 00 00 00 01 00 00 00 54 00 00 00 30 52 06 08 2B 06 01 05 05 07 03 02 06 08 2B 06 01
30 6F 31 0B 30 09 06 03 55 04 06 13 02 53 45 31 14 30 12 06 03 55 04 0A 13 0B 41 64 64 54 72 75 73 74 20 41 42 31 26 30 24 06 03 55 04 0B
A D0 38 72 D8 14 A8 45 C4 5D 2A 7D B7 B4 D6 C4 EE AC CD 13 44 B7 C9 2B DD 43 00 25 FA 61 B9 69 6A 58 23 11 B7 A7 33 8F 56 75 59 F5 CD 29 D
 D9 84 79 81 D9 1E 5B 14 07 23 36 65 8F B0 D8 77 B8 AC 41 6C 47 60 83 51 B0 F9 32 3D E7 FC F6 26 13 C7 80 16 A5 8F 5A FC 87 CF 78 79 89 21
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASAPI32\EnableFileTracing: 0x00000000
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASAPI32\EnableAutoFileTracing: 0x00000000
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASAPI32\EnableConsoleTracing: 0x00000000
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASAPI32\FileTracingMask: 0xFFFF0000
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASAPI32\ConsoleTracingMask: 0xFFFF0000
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASAPI32\MaxFileSize: 0x00100000
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASAPI32\FileDirectory: "%windir%\tracing"
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASMANCS\EnableFileTracing: 0x00000000
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASMANCS\EnableAutoFileTracing: 0x00000000
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASMANCS\EnableConsoleTracing: 0x00000000
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASMANCS\FileTracingMask: 0xFFFF0000
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASMANCS\ConsoleTracingMask: 0xFFFF0000
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASMANCS\MaxFileSize: 0x00100000
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASMANCS\FileDirectory: "%windir%\tracing"
```

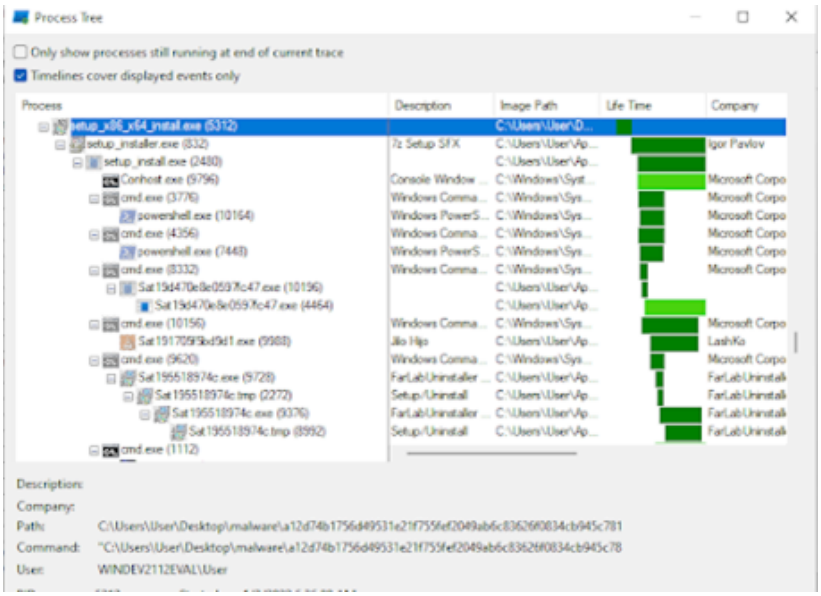
RegShot does an awesome job tracking everything related to the registry, and even goes as far as tracking file and folder events. We can see several that stand out in relation to the activity we've already seen, such as the keys which contain the malware executable names.

In a similar but more robust "all encompassing" ability, Process Monitor grabs a plethora of data as well:

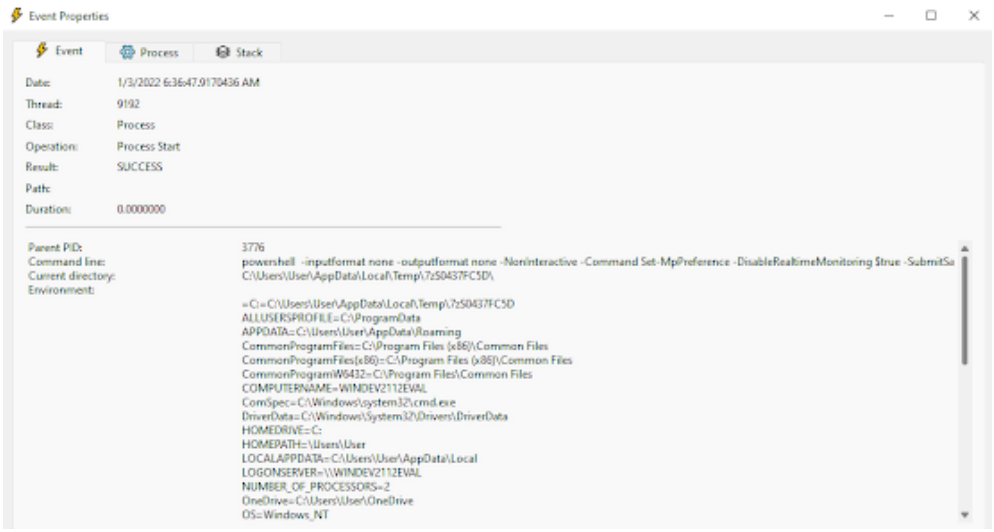




Using some of the pre-built filters located on the toolbar makes it easier to parse through some of the data. This information could be further whittled down if necessary, but during the initial glance I prefer to keep as much digestible information as necessary. Process Monitor has a "process tree" feature, pictured below:

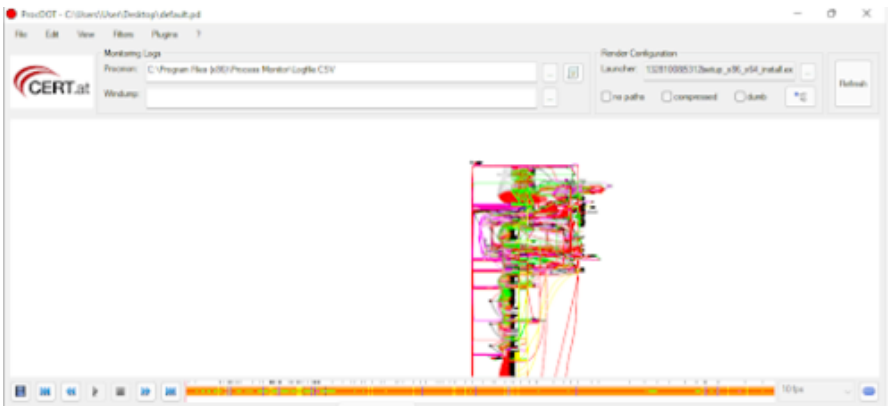


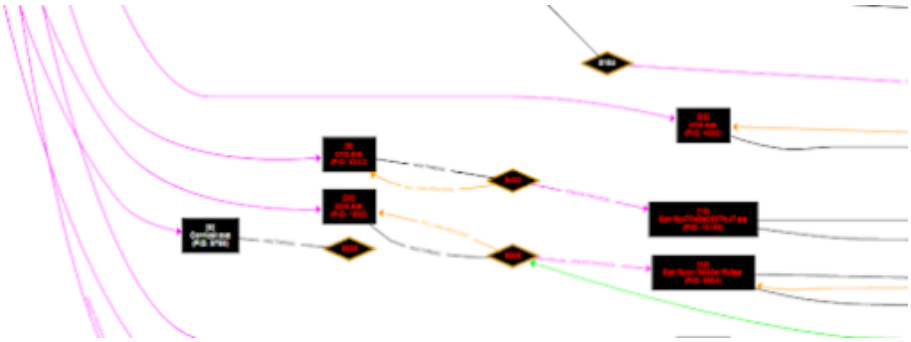
In this image, it can be observed that the processes associated with the malware spawn cmd.exe, which then spawns powershell.exe. There's an event in the Process Monitor data showing PowerShell with a "Process Start" operation after spawning from cmd.exe. This caught my interest because it was a unique operation in comparison with the bulk of PowerShell events. Further analysis resulted in catching this:



For those unfamiliar, Set-MpPreference is a PowerShell cmdlet associated with Windows Defender scans. The malware authors developed the specimen such that it would leverage PowerShell to disable Defender monitoring on the directory where the malware resides. In layman's terms, they're using PowerShell to tell Defender "nothing to see here" when it comes to the folder where all of the badness is. This is what could be considered an evasive maneuver.

A useful tool related to Process Monitor is ProcDOT. Developed by Christian Wojner, it provides correlation and visualization of Process Monitor and PCAP data. If there's a ton of data, the visual can seem convoluted from a zoomed out view, but it helps in showing how processes, file writes, and other events are related.





Without being able to connect back to its "mothership," the malware doesn't do much more than what's already been displayed. In order to further analyze a specimen of this nature, being reliant on a network connection, it would be useful to utilize dispensable hardware, a public network, and a VPN solution, with the intent being to do a full system restoration after detonation. This method is labor intensive and risky and not recommended for folks with minimal experience.

SO...WHAT IS REDLINESTEALER?

RedLineStealer is a credential stealer that targets web browsers. Access to the tool is available on the forums for several hundred dollars. The website HaveIBeenPwned recently added almost half a million entries related to RedLineStealer credential theft. Check it out here if you have concerns regarding your creds and whether or not they've been compromised in a malicious campaign.

Indicators of Compromise (IOCs) for RedLineStealer

File name: setup_x86_x64_install.exe

File hash: a12d74b1756d49531e21f755fef2049ab6c83626f0834cb945c781c39d40a177

File name: Sat19d470e8e0597fc47.exe (or similarly named matching the same alphanumeric pattern)

File path: C:\Users\User\AppData\Local\Temp\7zS4441B019\Sat19d470e8e0597fc47.exe

File hash: BC118B7708D56B93707A9BB025D3BF62D723B7932435A08299F59249C1C37DBE

File name: @.cmd

File path: C:\Users\User\AppData\Local\Temp\IXP000.TMP\@.cmd

File hash: 286227287F1FA79D5D5D909C2F457FC4D0AEFA6BE9E940F9A1F214D113FF88B4

File name: Sat195518974c.exe

File path: C:\Users\User\AppData\Local\Temp\7zS0437FC5D\Sat195518974c.exe

File hash: 13357A53F4C23BD8AC44790AA1DB3233614C981DED62949559F63E841354276A

File name: IXP000.TMP

File path: C:\Users\User\AppData\Local\Temp\IXP000.TMP

Directory: C:\Program Files (x86)\FarLabUninstaller*

Domains associated:

www.hhiuew33[.]com

gp.gamebuy768[.]com

one-mature-tube[.]com

cloudjah[.]com

kelenxz[.]xyz

ad-postback[.]biz

IPs associated:

212.193.30[.]45

159.69.246[.]184

Registry keys:

HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASAPI32
HKLM\SOFTWARE\Microsoft\Tracing\Sat194d446031aec9ca_RASMANCS
HKLM\SOFTWARE\Microsoft\Tracing\Sat19f1c04426464e86_RASAPI32
HKLM\SOFTWARE\Microsoft\Tracing\Sat19f1c04426464e86_RASMANCS

HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall\FarLabUninstaller.exe_is1

TOOLS AND REFERENCES

- Process Monitor
- ProcDOT
- WireShark
- Fiddler
- Process Hacker
- FakeDNS
- RegShot