



functions (155)	blacklist (29)	anonymous (1)	library (8)
<u>SearchPathA</u>	x	-	kernel32.dll
<u>MoveFileA</u>	x	-	kernel32.dll
<u>SetCurrentDirectoryA</u>	x	-	kernel32.dll
<u>SetFileAttributesA</u>	x	-	kernel32.dll
<u>CreateProcessA</u>	x	-	kernel32.dll
<u>RemoveDirectoryA</u>	x	-	kernel32.dll
<u>GetTempFileNameA</u>	x	-	kernel32.dll
<u>GetExitCodeProcess</u>	x	-	kernel32.dll
<u>WritePrivateProfileStringA</u>	x	-	kernel32.dll
<u>WriteFile</u>	x	-	kernel32.dll
<u>FindNextFileA</u>	x	-	kernel32.dll
<u>FindFirstFileA</u>	x	-	kernel32.dll
<u>DeleteFileA</u>	x	-	kernel32.dll
<u>CloseClipboard</u>	x	-	user32.dll
<u>SetClipboardData</u>	x	-	user32.dll
<u>EmptyClipboard</u>	x	-	user32.dll
<u>SystemParametersInfoA</u>	x	-	user32.dll
<u>OpenClipboard</u>	x	-	user32.dll
<u>ExitWindowsEx</u>	x	-	user32.dll
<u>SHGetPathFromIDListA</u>	x	-	shell32.dll
<u>SHBrowseForFolderA</u>	x	-	shell32.dll
<u>SHGetFileInfoA</u>	x	-	shell32.dll
<u>ShellExecuteA</u>	x	-	shell32.dll

Based on the imports, this sample shows potentially:

- Anti-debugging capabilities (EmptyClipboard, GetTickCount)
- Parsing through files and folders (FindFirstFileA, FindNextFileA, SearchPath, CreateFileA)
- Evasive behaviors/artifact destruction (DeleteFile, RemoveDirectory)
- File writing (CreateFileA, WriteFile, MoveFile)
- Registry interactions (RegCreateKey, RegDeleteKey, RegEnumKey, RegOpenKey, RegSetValue, RegQueryValue)

Looking at the strings tab, we see a lot of the same references to the API calls, especially if sorting for blacklist items to show first:

file settings about

	encoding (2)	size (bytes)	file-offset	blacklist (33)	hint (99)	value (3464)
indicators (45)	ascii	9	0x0000654A	x	function	WriteFile
virustotal (error)	ascii	18	0x000065EC	x	function	GetExitCodeProcess
dos-header (64 bytes)	ascii	13	0x00006A58	x	function	ExitWindowsEx
dos-stub (144 bytes)	ascii	14	0x00006BE2	x	function	CloseClipboard
rich-header (Visual Studio)	ascii	16	0x00006BF4	x	function	SetClipboardData
file-header (Oct.2008)	ascii	14	0x00006C08	x	function	EmptyClipboard
optional-header (GUI)	ascii	13	0x00006C1A	x	function	OpenClipboard
directories (3)	ascii	26	0x00006DDA	x	function	SHGetSpecialFolderLocation
sections (virtualized)	ascii	10	0x000064F0	x	-	DeleteFile
libraries (8) *	ascii	13	0x000064FE	x	-	FindFirstFile
functions (155)	ascii	12	0x00006510	x	-	FindNextFile
exports (n/a)	ascii	25	0x00006572	x	-	WritePrivateProfileString
tls-callbacks (n/a)	ascii	10	0x00006696	x	-	SearchPath
.NET (n/a)	ascii	8	0x000066CC	x	-	MoveFile
resources (6) *	ascii	19	0x000066D8	x	-	SetCurrentDirectory
strings (3464)	ascii	17	0x0000672A	x	-	SetFileAttributes
debug (n/a)	ascii	13	0x00006860	x	-	CreateProcess
manifest (asInvoker)	ascii	15	0x00006872	x	-	RemoveDirectory
version (n/a)	ascii	15	0x00006886	x	-	GetTempFileName
overlay (Nullsoft)	ascii	20	0x00006AA8	x	-	SystemParametersInfo
	ascii	15	0x00006D76	x	-	SHFileOperation
	ascii	12	0x00006D8A	x	-	ShellExecute
	ascii	13	0x00006D9A	x	-	SHGetFileInfo

Outside of those, nothing is proving to be too conclusive here.

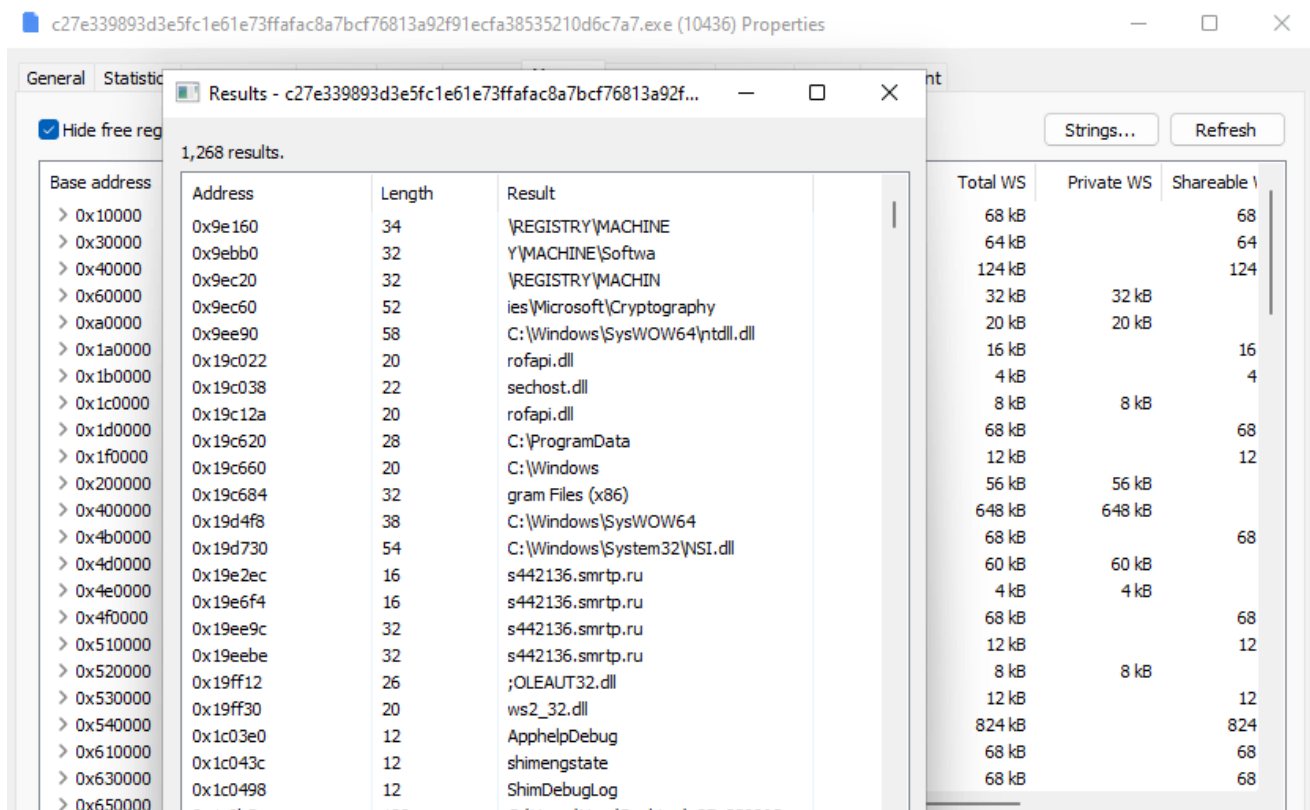
I pushed the file over to REMnux to give a stab at it with capa. Capa gave the following output:

```
remnux@remnux:~$ capa lokibot
loading : 100%| | 579/579 [00:00<00:00, 1591.19 rules/s]
matching: 100%| | 83/83 [00:06<00:00, 13.00 functions/s]
WARNING:capa:-----
WARNING:capa: This sample appears to be an installer.
WARNING:capa:
WARNING:capa: capa cannot handle installers well. This means the results may be misleading or incomplete
. You should try to understand the install mechanism and analyze created files with capa.
WARNING:capa:
WARNING:capa: Use -v or -vv if you really want to see the capabilities identified by capa.
WARNING:capa:-----
```

In order to analyze further with capa, I'll need to dump the actual malware executable once it starts running during the dynamic analysis stage and rebuild it. I'll have those details after the dynamic analysis section.

## DYNAMIC ANALYSIS

Prior to detonating the sample, I had Process Hacker, Process Monitor, and WireShark running to capture any events. I was able to capture the following data:



The memory strings in Process Hacker offered some IOCs. We see a domain (`s442136.smrtp[.]ru`) as well as some registry interactions.



Process Monitor offered the following:

2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Nichrome\User Data\Default\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Nichrome\User Data\Default\Web Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Nichrome\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Nichrome\Default\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Rock Melt\User Data\Default\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Rock Melt\User Data\Default\Web Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Rock Melt\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Rock Melt\Default\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Spark\User Data\Default\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Spark\User Data\Default\Web Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Spark\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Spark\Default\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Chromium\User Data\Default\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Chromium\User Data\Default\Web Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Chromium\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Chromium\Default\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Titan Browser\User Data\Default\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Titan Browser\User Data\Default\Web Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Titan Browser\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Titan Browser\Default\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Torch\User Data\Default\Login Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Torch\User Data\Default\Web Data	PATH NOT FOUND	Desired Access: R...
2:18:5...	c27e339893d3...	10436	CreateFile	C:\Users\User\AppData\Local\Torch\Login Data	PATH NOT FOUND	Desired Access: R...

We see a ton of "CreateFile" operations with browser file paths. It would be easy to be misled by the fact that the operation title is "CreateFile" and believe that the executable is attempting to generate files on the victim system. Reading Microsoft documentation offers some more context:

## CreateFileA function (fileapi.h)

Article • 10/13/2021 • 29 minutes to read

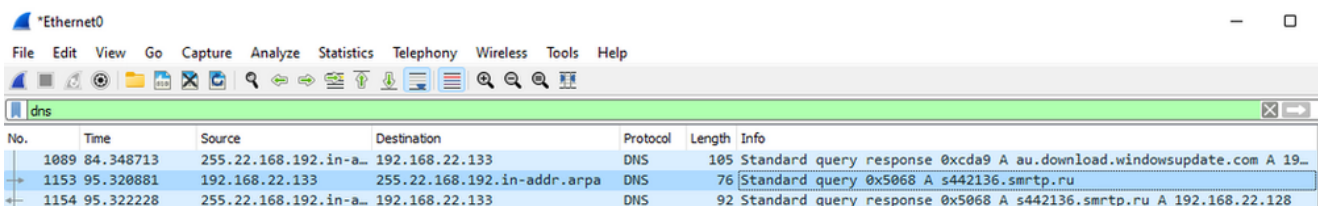
Is this page helpful?  

Creates or opens a file or I/O device. The most commonly used I/O devices are as follows: file, file stream, directory, physical disk, volume, console buffer, tape drive, communications resource, mailslot, and pipe. The function returns a handle that can be used to access the file or device for various types of I/O depending on the file or device and the flags and attributes specified.

To perform this operation as a transacted operation, which results in a handle that can be used for transacted I/O, use the [CreateFileTransacted](#) function.

Not only does this function allow for the creation of files, but also opening them. On the right side of the Process Monitor screenshot we see the value "path not found," meaning that the malware tried to open or access the browser file paths and they did not exist. Being that the malware is a password stealer, it is likely checking these file paths for saved credentials.

Seeing the WireShark output, we see information that corroborates earlier findings:



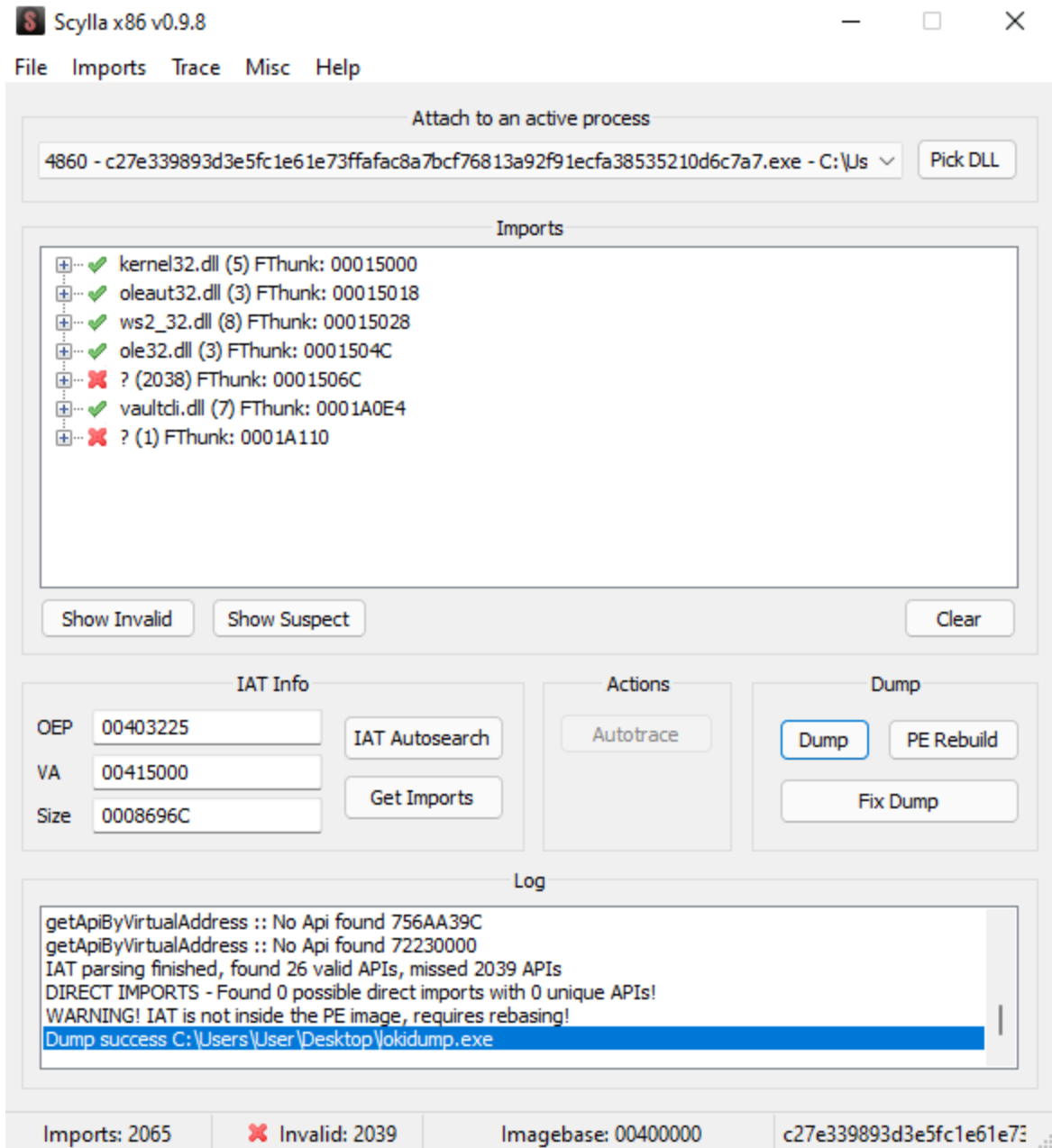
No.	Time	Source	Destination	Protocol	Length	Info
1089	84.348713	255.22.168.192.in-a...	192.168.22.133	DNS	105	Standard query response 0xcda9 A au.download.windowsupdate.com A 19...
1153	95.320881	192.168.22.133	255.22.168.192.in-addr.arpa	DNS	76	Standard query 0x5068 A s442136.smrtp.ru
1154	95.322228	255.22.168.192.in-a...	192.168.22.133	DNS	92	Standard query response 0x5068 A s442136.smrtp.ru A 192.168.22.128

## REBUILDING WITH SCYLLA

---

So like I mentioned early on, the initial file is an installer and not the actual child process that we've analyzed in the dynamic stage. I'll demonstrate how one could actually get a "tangible" version of the malware that's executing to then analyze it with tools like capa. The tool we'll be using is Scylla, and imports reconstructor developed by NtQuery and available on GitHub.

First, make sure the malware is already running. Next open Scylla x86 and attach it to the active process. Click "IAT autosearch," and then "get imports," followed by "dump." Name it something intuitive, and voila. This will reconstruct the executable.



I pushed this version back over to the REMnux box for analysis, and it worked fine with capa:







Dropped executable:

71e155ee000c0d1cbba18b92f0d512217afe195ba40f9326c60523cdfd3fa742

**Domains:**

s442136.smrtp[.]ru

REFERENCES:

[LokiBot Malware \(CISA\)](#)

[New Campaign Sees LokiBot Delivered Via Multiple Methods \(TrendMicro\)](#)

Scylla