An Analysis of Sidoh: WIZARD SPIDER's Exfiltration Tool

crowdstrike.com/blog/sidoh-wizard-spiders-mysterious-exfiltration-tool/

Alexander Hanel August 31, 2021

WIZARD SPIDER is an established, high-profile and sophisticated eCrime group, originally known for the creation and operation of the TrickBot banking Trojan. This Russia-based eCrime group originally began deploying TrickBot in 2016 to conduct financial fraud, but has since evolved into a highly capable group with a diverse and potent arsenal, including Ryuk, Conti and BazarLoader. Their toolset covers the entirety of the kill chain, from delivery to post-exploitation tools and big game hunting (BGH) ransomware, enabling WIZARD SPIDER to conduct a wide range of criminal activities against enterprise environments.

Sidoh (aka Ryuk Stealer) is a keyword-based exfiltration tool used by WIZARD SPIDER. Sidoh (as well as *Ryuk*) is the name of a character from the anime series Death Note. In the series, the character Sidoh has an item stolen by Ryuk. Since WIZARD SPIDER's tool is used for exfiltration, the name Sidoh is fitting.

WIZARD SPIDER's Sidoh has an aura of mystery due to its rarity and the keyword list it uses to determine what data is exfiltrated. As of this blog's publication, CrowdStrike Intelligence has observed 16 unique SHA256 hashes, with nine of them containing unique build times (date and time of compilation) and with the first build date of June 16, 2019, and the last build date of Jan. 18, 2020. Sidoh searches for specific file types with a fixed set of keywords. If a file matches Sidoh's criteria, it is exfiltrated via FTP to a hardcoded IP address.

The list of keywords suggests the adversary is searching for and targeting data related to government, military and financial sectors. It is unknown if WIZARD SPIDER was using Sidoh to steal files for espionage purposes or if they were stealing files for extortion purposes. Stealing files for espionage purposes is unusual for criminal

threat actors. However, GameOver Zeus3 was a previously observed criminal malware family that searched victim systems for files matching keywords related to foreign government officials, military documents, classified information and terrorism.

The keywords from the GameOver Zeus search queries resemble some of those found in Sidoh. In May 2019, OUTLAW SPIDER, the operators of RobbinHood ransomware, made headlines with ransoming the machines and exfiltrating data from the City of Baltimore (COB). This incident was one of the first instances observed by CrowdStrike Intelligence of data exportation to incentivize ransom payments. WIZARD SPIDER could have taken notice of the exfiltration and data leak extortion tactic and wrote Sidoh as a tool to determine the value of exfiltrated data in a set of limited test runs inspired by OUTLAW SPIDER targeting of COB. In August 2020, WIZARD SPIDER did add data exfiltration and data leak sites to the big game hunting list of tactics.

Two variants of Sidoh have been observed by CrowdStrike Intelligence. The first variant was observed in mid-June 2019 and the second was observed in mid-January 2020. Both versions are similar except for bug fixes and updated keyword lists. Upon initial analysis of the early versions of Sidoh, CrowdStike Intelligence dismissed the sample as being a variant of Ryuk. This initial assessment was based on a cursory analysis of Sidoh's WinMain function. Ryuk and Sidoh both contain functionality that reads a file passed as a command line argument, sleeps for 5,000 microseconds and then calls `DeleteFileW` to delete the file. Along with the deletion of the file, Sidoh's code was compiled in Visual Studio, contained the same API chains to accomplish specific functionality and the code even "felt" like Ryuk.

It was only upon deeper inspection that it was realized that the sample was not Ryuk, but a family of malware that looks to have borrowed code from Ryuk's source code. Even though both Sidoh and Ryuk are compiled using Visual Studio, they are not compiled using the same build chain. For example, a Ryuk binary compiled on June 26, 2019, contains an older toolset than a Sidoh binary compiled on July 8, 2019. The difference in the detected toolsets between Ryuk and Sidoh hint that the samples were compiled on different machines with different installed versions of Visual Studio. Table 1 shows the detected Visual Studio toolsets for the binaries.

Ryuk	Sidoh
Visual C++ 14.0.x 2015 (build 23918)	Visual C++ 14.0 2015 (build 24215)
Visual C++ 9.0 2008 SP1 (build 30729)	Visual C++ 14.0.x 2015 (build 24210)
Visual C++ 14.0.x 2015 (build 23907)	Visual C++ 14.0 2015 (build 24215)
	Visual C++ 14.0.x 2015 (build 23918)
Visual C++ 9.0 2008 SP1 (build 30729)	
Visual C++ 14.0.x 2015 (build 24123)	

Table 1. Ryuk and Sidoh detected toolsets

The exact intent of Sidoh is only known by the threat actors that operated it, but the keywords used by WIZARD SPIDER open up speculation of its targeted audience and usage of the exfiltrated data.

Technical Analysis

The following technical analysis is based on the newest version of Sidoh. Differences between the two variants are noted throughout the text when applicable.

Upon execution, Sidoh checks for the presence of a command line argument passed to it. If the argument is present, it is treated as a file path. Sidoh sleeps for 5,000 microseconds and then deletes the file if the argument is present. Sidoh lists all available drives on the host by calling `GetLogicalDrives`. For each available drive, it

calls `GetDriveTypeW` to ensure the drive type is not `DRIVE_CDROM`. If the drive type is not a CD-ROM, the drive path's contents are searched. After the drives have been searched, Sidoh attempts to search the contents of hosts that have IP addresses present within the compromised host's ARP entries. The ARP entries are retrieved by calling `GetIpNetTable`. For each entry, Sidoh attempts to mount the IP address from within the entry as a network drive, using Server Message Block (SMB), and then proceeds to search the drive.

To enumerate files on disk, Sidoh calls `FindFirstFileW` and `FindNextFileW`. Sidoh contains a deny list targeting specific file types of folder names. Table 2 shows a block list of folders and file names whose contents are ignored by Sidoh.

`Sample Music`	`Sample Pictures`	`$Recycle.Bin`
`Tor Browser`	`Package Cache`	`RyukReadMe.txt**`
`microsoft`	`UNIQUE_ID_DO_NOT_REMOVE**`	`PUBLIC`
`windows`	`PerfLogs`	`Windows`
`ProgramData`	`Firefox`	`Intel`
`Mozilla`	`Microsoft`	`$WINDOWS`
`Program Files`	`\\Users\\Public\\Pictures`	`MySQL`
`log`	`.dll`	`AhnLab`

Table 2. Sidoh folder name block list

Some of the names in Table 2 are not directories (e.g., `.dll`). The fields that end with ** are names that are artifacts from a Ryuk infection. For example, `RyukReadMe.txt` was the name for Ryuk ransom notes before WIZARD SPIDER switched the ransom note to HTML.

Sidoh does not attempt to exfiltrate files over 50MB, and earlier variants would not exfiltrate files over 20MB. If a file is over 50MB, it is ignored. Along with folders names, Sidoh has an allow list of file extensions. Earlier variants contained the file extension `.RYK`, which is appended to files after being encrypted by Ryuk. The presence of the string related to Ryuk suggests that Sidoh was either derived from the Ryuk source code or designed to be compatible with systems that have been encrypted by Ryuk. Table 3 contains the block list of file extensions for variant 2. The newer list is more extensive, likely with an end goal of speeding up time spent scanning the host for files.

`.exe`	`.sdi`	`.pem`	`.sys`	`.xpi`	`.rsm`	`.msc`
`.msi`	`.dic`	`.ibd`	`.ddf`	`.mp3`	`.wbverify`	`.ascx`
`.dll`	`.pyd`	`.db`	`.sqlite`	`cached`	`.tmp`	`.css`
`.lnk`	`.qml`	`.h`	`.ttf`	`ppt`	`.cat`	`.browser`
`.sdi`	`.js`	`.lib`	`.cdx`	`ppsm`	`.asp`	`.bmp`
`.wim`	`.png`	`.microsoft`	`.thmx`	`cached`	`.config`	`.url`
`.chm`	`.log`	`.txmx`	`.new`	`wtv`	`.inf`	`.search-ms`
`.vicache`	`.ini`	`.tif`	`.little`	`NTUSER`	`.din`	`.wmv`
`.chm`	`.lnk`	`.iobj`	`.cdx`	`.contact`	`.oem`	`.icml`

`.xml`	`.cab`	`.ipdb`	`.gif`	`.wbcat`	`.ps1`	`readme`

Table 3. Sidoh file extension name block list

If the filename contains an extension of `.cpp`, `.h` `.xls`, `.xlsx`, `.doc`, `.docx`, . `docb`, `.pdf` or `wallet.dat`, an attempt to automatically exfiltrate the file is made. Early variants also included `.txt` and `.gov` file extension but were removed. Further details about the attempt are described in the next section, *Exfiltration and Infrastructure*. If the filename does not contain one of the previously mentioned file extensions, the filename is then checked against the keyword list in Table 4. If the filename contains one of the keywords in the following table, the file is exfiltrated.

SECURITY	marketwired	fbi	Secret	war	victim	federal
N-CSR	10-Q	csi	scheme	suspect	court	bureau
10-SB	10Q	gun	tactical	cyber	hidden	government
EDGAR	8K	NATO	Engeneering	document	bribery	security
spy	fraud	Nato	explosive	treasonrestricted	contraband	unclassified
radar	hack	convictMilitary	drug	private	operation	concealed
censored	NSA	military	traitor	confident	undercover	newswire
agent	FBI	submarine	embeddedspy	important	clandestine	marketwired
newswire	CSI	Submarinesecret	radio	pass	investigation	Clearance

Table 4. Sidoh file name exfiltration keyword list

Early variants of Sidoh contained functionality to search the contents of `.docx` or `.xlsx` files. To search the contents of `.docx` or `.xlsx` files, Sidoh must first parse the file type. Modern Microsoft Office documents are stored in the Microsoft Open XML (MOX) format. This format primarily consists of a ZIP file with XML data. Before decompressing the ZIP file, Sidoh would import a Dynamic Link Library (DLL) named `libzip.dll`. It is derived from the open-source C library, libzip, which can be used for reading, writing and modifying ZIP files. This library is not installed on Windows by default. In order to execute properly, these early variants of Sidoh would need `libzip.dll` to be present within the working directory or within the standard DLL search order. The Sidoh executable and `libzip.dll` are likely written to disk by a dropper.

Once the libzip has been imported, it is used to decompress the document. Once decompressed, Sidoh reads the path `word/document.xml` for `.docx` files and `xl/worksheets/sheet*.xml` for `.xlsx`. The contents are read into a buffer and then searched for the keywords in Table 5. The bug existed in how Sidoh searched `.xlsx` for keywords. Data from rows and columns in Excel spreadsheets are not stored in `xl/worksheets/sheet*.xml` but are stored at the path `xl/SharedStrings.xml`. Therefore, the `.xlsx` file data that Sidoh reads and searches only contains the mapping of the rows and columns to fields in `SharedStrings.xml` but none of the text that would contain the keywords being searched for. The reading and searching of `.docx` or `.xlsx` was removed in the newest variant, likely because of the described error. Newer variants simply read the contents of the file and search for one of the keywords in Table 5.

personal	censored	traitor	checking	clandestine	seed	Olivia

securityN-CSR10-SBEDGAR	bribery	suspect	saving	illegal	personal	Noah
spy	contraband	cyber	routing	compromate	confident	Ava
radaragentnewswire	operation	document	finance	privacy	mail	William
marketwired	gun	embeddedspy	agreement	private	letter	Isabella
10-Q	attack	radio	SWIFT	contract	passport	James
10Q	military	war	BIC	concealed	victim**	Sophia
8K	tank	submarine	IBAN	backdoorundercover	court	Logan
fraud	convict	fbi	license	clandestine	id	Clearance
hack	scheme	restricted	Compilation	investigation	NATO	
NSA	tactical	secret	report	federal	Nato	
FBI	Engeneering	CSI	secret	bureau	scans	
defence	explosive	balance	confident	government	Emma	
treason	drug	statement	hidden	security	Liam	

Table 5. Sidoh file name exfiltration keyword list

The lists in Table 4 and Table 5 contain many words that are misspelled (e.g., Engeneering) or two words concatenated (e.g., treasonrestricted). These errors hint that this list was hastily imputed or never reviewed. Further details about the attempts to upload the files are described next.

Exfiltration and Infrastructure

Sidoh uploads documents that match the search criteria to a command-and-control (C2) IP address via FTP with a hard-coded username and password of `anonymous`. If the FTP server is not available, Sidoh typically contains a backup IP address that it can use. One variant of Sidoh did not contain a backup IP address. Once connected, the files are uploaded to a hard-coded directory with four random digits prefixing the original filename. If the connections fail, Sidoh sleeps for a random amount of time (between 0 and 125 seconds) and then tries to connect again. Sidoh does not keep an active connection to the FTP server but continuously logs into the FTP server.

If during the exfiltration process Sidoh fails to connect to the C2 after three attempts, it scans the filename or contents of files with an extension of `.cpp`, `.xls`, `.xlsx`, `.doc`, `.docx`, `.docb` or `.pdf` for the targeted keywords to determine if additional effort should be made to exfiltrate those files. If there is a match, Sidoh makes another three connection attempts. After each failed connection, Sidoh will sleep for a random amount of time (between 0 and 125 seconds) between each attempt. If all attempts to connect to the C2 fail, Sidoh discards the current file being uploaded and continues to search for more files to exfiltrate. Table 6 contains all known Sidoh IP addresses and observed folder paths for uploading exfiltrated files.

IP Address	IP Address	Upload Path	Build Time
`109.236.92[.]162`	N/A	`/upload/files/a7`	`2019-06-16 21:20:14`
`109.236.92[.]162`	`185.254.121[.]157`	N/A	`2019-06-21 08:30:19`
`185.254.121[.]157`	`109.236.92[.]162`	`/upload/files/military2`	`2019-06-22 03:37:21`

185.254.121[.]157	109.236.92[.]162	/upload/files/3	2019-07-08 17:21:37
185.254.121[.]157	109.236.92[.]162	/upload/files/sharpsec	2019-07-11 22:26:03
185.254.121[.]157	109.236.92[.]162	upload/files/a71	2019-07-17 12:12:37
66.42.76[.]46	N/A	/files_server/a8-5	2019-08-18 19:45:35
66.42.108[.]141	45.76.1[.]57	/test50/fx3-92	2020-01-18 23:44:40
66.42.108[.]141	45.76.1[.]57	/test50/fx3-92	2020-01-19 00:11:32

Table 6. Sidoh IP addresses, upload paths and build times

Due to reallocation, the IP addresses in Table 6 should not be used for blocking.

SHA256	Build Time
df6847bbf7e75ded54028081f5f27abb199562409aee6e20f99abcda5b48fb51	2019-06-16 21:20:14
f07079472f1cb0247f530001f02d6189443146a719d986bca750ee9b1139e84f	2019-06-21 08:30:19
a1ce52437252001b56c9ccd2d2da46240dc38db8074a5ed39a396e8c8e387fc2	2019-06-22 03:37:21
e6762cb7d09cd90d5469e3c3bfc3b47979cd67aa06c06e893015a87b0348c32c	2019-07-08 17:21:37
6f06e5a8bdf983ec73177ef63ea053d391b46915a7dd1fbd0ddea5c70471f593	2019-07-11 22:26:03
cc4a0b4080844e20fb9535679f7b09a3e2449729ce1815d1e5a64272b0225465	2019-07-17 12:12:37
c64269a64b64b20108df89c4f1a415936c9d9923f8761d0667aa8492aa057acb	2019-08-18 19:45:35
a8c4703fab7d2548701523b4c215d7cb57d337cc243046647bda18d4e6690853	2020-01-18 23:44:40
5794ce98af725b29ae32280909a725812a89fd4ecdbbf7f121b83f031526a967	2020-01-19 00:11:32

Table 7. Sidoh SHA256 Hashes and Build Time

Conclusion

WIZARD SPIDER is one of the most sophisticated groups tracked by CrowdStrike. Their threat arsenal ranges from banking trojans to spam bots to ransomware — with all of these tools designed with an end result of getting money from their victims. Some of these tools have been short-lived, but the diversity in tooling used by WIZARD SPIDER demonstrates their desire to use new strategies to monetize their attacks. It is unknown if Sidoh is one such strategy that experimented with monetizing victims by stealing potentially sensitive or proprietary data, or if Sidoh was used by WIZARD SPIDER to steal data from specific victims at the request of a third party.

There is even the possibility that the original Hermes source code (which Ryuk was derived from) was modified to be Sidoh, and the threat actors added the Ryuk file artifact references as a false flag. But the likely and simplest solution is Sidoh was used in rare instances to automatically steal data from compromised hosts by WIZARD SPIDER.

As of this publication date, CrowdStrike has not observed Sidoh within the telemetry of our customers. If your company was a victim of Sidoh and you have further details you'd be willing to share, please reach out to CrowdStrike Intelligence.

Additional Resources

- *For more intel about WIZARD SPIDER, visit the CrowdStrike Adversary Universe.*
- *To find out how to incorporate intelligence on threat actors into your security strategy, visit the Falcon X™ Threat Intelligence page.*
- *Learn about the powerful, cloud-native CrowdStrike Falcon® platform by visiting the product webpage.*
- *Get a full-featured free trial of CrowdStrike Falcon Prevent™ to see for yourself how true next-gen AV performs against today's most sophisticated threats.*