Study of the Spyder modular backdoor for targeted attacks

Published: 2021-03-04 · Archived: 2026-04-05 19:54:50 UTC

04.03.2021

Real-time threat news | Hot news | All the news | Virus alerts

Download PDF

March 4, 2021

In December 2020, the Doctor Web virus laboratory was contacted by a telecommunications company based in Central Asia after its employees discovered suspicious files on their corporate network. During the examination, our analysts extracted and studied a malicious sample, which turned out to be one of the backdoors used by the hacker group known as Winnti.

We already came across the malware Winnti uses when we studied the **ShadowPad** backdoor samples that we found in the compromised network of a state institution in Kyrgyzstan. In addition, earlier in the same network, we found another specialized backdoor called **PlugX**, which has many intersections with ShadowPad in the code and network infrastructure. A separate material was devoted to the comparative analysis of both families.

In this study, we analyze the uncovered malicious module, explore its algorithms and features, and define its connection with other well-known tools of the Winnti APT group.

Main features

On the infected device, the malicious module was located in the system directory C:\Windows\System32 as oci.dll. Thus, the module was prepared for launch by the MSDTC (Microsoft Distributed Transaction Coordinator) system service using the DLL Hijacking method. According to our data, the file got to the computers in May 2020, but the method of initial infection remains unknown. The Event Log contained records of the creation of services designed to start and stop MSDTC, as well as for the backdoor execution.

```
 Log Name:      System
Source:        Service Control Manager
Date:          23.11.2020 5:45:17
Event ID:      7045
Task Category: None
Level:         Information
Keywords:      Classic
User:          <redacted>
Computer:      <redacted>
Description:
A service was installed in the system.
```

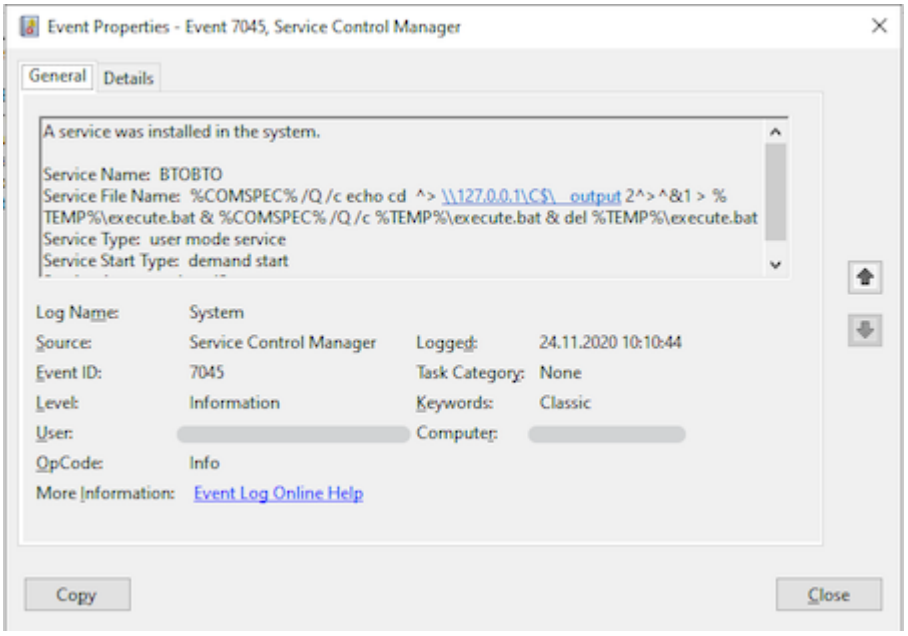
```
Service Name:  IIJVXRUMDIKZTTLAMONQ
Service File Name:  net start msdtc
Service Type:  user mode service
Service Start Type:  demand start
Service Account:  LocalSystem
```

```
 Log Name:      System
Source:        Service Control Manager
Date:          23.11.2020 5:42:20
Event ID:      7045
Task Category: None
Level:         Information
Keywords:      Classic
User:          <redacted>
Computer:      <redacted>
Description:
A service was installed in the system.

Service Name:  AVNUXWSHUNXUGGAUXBRE
Service File Name:  net stop msdtc
Service Type:  user mode service
Service Start Type:  demand start
Service Account:  LocalSystem
```

We also found traces of other services running that had random names. Their files were located in directories like C:\Windows\Temp\<random1>\<random2>>, where random1 and random2 are strings of random length and random Latin characters. At the time of the study, these services' executable files were missing.

An interesting find was a service that indicates the use of a smbexec.py utility for remote code execution from the Impacket set. The attackers used this tool to establish remote access to the command shell in a semi-interactive mode.

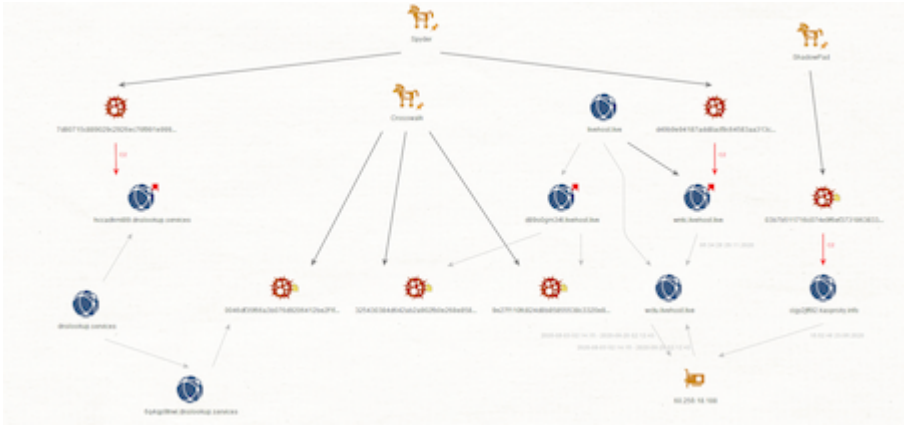


The studied malicious sample was added to the Dr.Web virus database as BackDoor.Spyder.1. In one of the discovered Spyder samples, the debug logging functions and messages remained. Messages used when communicating with the C&C server contained the string "Spyder".

```
10064C6C 5B 53 70 79 64 65 72+aSpyderRegister db '[Spyder] register mod %d error = %u.',0Ah,0
10064C6C 5D 20 72 65 67 69 73+                                      ; DATA XREF: sub_10042110+A0↑o
10064C92 00 00                              align 4
10064C94                            ; const char aSpyderClientMo[]
10064C94 5B 53 70 79 64 65 72+aSpyderClientMo db '[Spyder] client module init error = %d.',0
10064C94 5D 20 63 6C 69 65 6E+                                      ; DATA XREF: sub_10042110+4D↑o
10064CBC                            ; const char aSpyderSetCaFor[]
10064CBC 5B 53 70 79 64 65 72+aSpyderSetCaFor db '[Spyder] set ca for client id=%u error=%d',0
10064CBC 5D 20 73 65 74 20 63+                                      ; DATA XREF: sub_10042600+1C7↑o
10064CE6 00 00                              align 4
10064CE8                            ; const char aSpyderAllocCli_0[]
10064CE8 5B 53 70 79 64 65 72+aSpyderAllocCli_0 db '[Spyder] ALLOC client uid = %u.',0
10064CE8 5D 20 41 4C 4C 4F 43+                                      ; DATA XREF: sub_10042600+118↑o
10064D08                            ; const char aSpyderAllocCli[]
10064D08 5B 53 70 79 64 65 72+aSpyderAllocCli db '[Spyder] alloc client error = %d.',0
10064D08 5D 20 61 6C 6C 6F 63+                                      ; DATA XREF: sub_10042600+102↑o
10064D2A 00 00                              align 4
10064D2C                            ; const char aSpyderServerAd[]
10064D2C 5B 73 70 79 64 65 72+aSpyderServerAd db '[spyder] server address already exists in conf list.',0
10064D2C 5D 20 73 65 72 76 65+                                      ; DATA XREF: sub_10042600+95↑o
10064D61 00 00 00                           align 4
10064D64                            ; const char aSpyderProxySet[]
10064D64 5B 53 70 79 64 65 72+aSpyderProxySet db '[Spyder] proxy setting exists, srv=%s',0
```

The backdoor is notable for a number of interesting features. First, oci.dll contains the main PE module, but with missing file signatures. Erasing the header signatures was presumably done to obstruct the backdoor detection in the device's memory. Secondly, the payload itself does not carry malicious functionality, but serves to load and coordinate additional plug-ins received from the C&C server. With these plug-ins, the backdoor performs its main tasks. Therefore, this family has a modular structure, just like the other backdoor families used by **Winnti** — the previously mentioned **ShadowPad** and **PlugX**.

Analysis of **Spyder**'s network infrastructure revealed a link to other Winnti attacks. In particular, the infrastructure used by the **Crosswalk** and **ShadowPad** backdoors described in the Positive Technologies study corresponds with some of the **Spyder** samples. The graph below clearly shows the identified intersections.



For a detailed description of BackDoor.Spyder.1 and how it works, see the PDF-version of the study or the Doctor Web Virus Library.

Conclusion

The analyzed sample of **BackDoor.Spyder.1** is notable primarily because its code does not perform direct malicious functions. Its main tasks are to covertly operate within the infected system and establish communication with the control server and then wait for operator commands. At the same time, it has a modular structure that allows the operator to scale its capabilities, providing any functionality depending on the needs of the attackers. The plug-ins make the considered sample similar to **ShadowPad** and **PlugX**, which, together with the intersections in their network infrastructures, allows us to conclude that it is used by **Winnti**.

Indicators of compromise.
