Fancy Bear Droppings: Malware Analysis Report of DROPPER

Matthías Ragnarsson

February 27, 2017

1 Executive Summary

This report delineates the analysis of DROPPER, a DLL file extracted from Bulletin.doc. It is surmised that this file's purpose is to be a packed container, a dropper, for a malicious payload: nshwmpfs.dll. The file is furthermore identified to be a part of the toolset of Sofacy, a.k.a. Fancy Bear or APT28, an advanced persistent threat group believed to be based in Russia. A quick scan of the payload indicates a proxy service to coneal communication with a C2 server.

2 Dropping the Payload

The DLL file we're analyzing is the following:

Name	DROPPER
SHA256	231aea51cbf29c0877119ccc742823454f080852601500c3300a0d7ec85cb64a
Size	123532
Compile Date	2016-08-13 01:26:26

I performed basic dynamic analysis on it by appending `.dll` to the file name and opening `cmd.exe` and running

```
> rundll32.exe DROPPER.dll,#1
```

while capturing activity with Procmon and CaptureBAT. Viewing the CaptureBAT log and filtering for `rundll32.exe`, we see that the process dropped the following payload:

```
C:\Users\rev_eng\AppData\Local\nshwmpfs.dll
```

Name	nshwmpfs.dll
SHA256	73db52c0d4e31a00030b47b4f0fa7125000b19c6c9d462c3d0ce0f9d68f04e4c
Size	33280
Compile Date	2016-08-10 09:38:01

From the compile times, we can see that DROPPER was compiled later, indicating that it may have been made to fit the payload specifically. A quick scan of nshwmpfs.dll yields one URL: `https://%snetwork.proxy.ht`, indicating that this is a proxy service that may be used to conceal communication with the C2 server.

3 Indicators of Compromise of DROPPER

3.1 Matched Sofacy APT Group

A comment made by Venom23 on an existing Virustotal report of the sample informs that the THOR APT Scanner matched the sample to the Sofacy_Malware_Sednit_Dec16_A rule in the Sofacy Monitoring ruleset. Sofacy, a.k.a. Fancy Bear is an APT group believed to be sponsored by the Russian government.

3.2 High antivirus detection ratio

On the Virustotal report, the detection ratio is 34 / 57, which is fairly high.

3.3 Suspicious imports

A number of suspicious imports were listed by Peframe. Notable ones are the GetProcAddress and LoadLibrary which are indicative of packing, ShellExecute indicative of running shell interpreter commands (or a resulting payload), and IsDebuggerPresent which can be used for anti-debugging.

3.4 High entropy .data section

Virustotal reveals the .data section of the file has entropy of 7.91. Scanning with the Peframe flags the section as suspicious. Moreover, closer examination of it with radare2 reveals the entropy of individual blocks of the section to be nearly constant for most of the lower addresses of the section.

3.5 Execution and termination prior to entrypoint

While attempting to make the file drop manually (an effort I abandoned), the executable terminated in the x64dbg debugger before the entrypoint could be reached. I could not pinpoint the cause, but possibilities are (non TLS Callback) exploitation of the PE format to execute code before the entry point or the less likely debugger vulnerability exploitation.

4 Detection signatures for DROPPER

Unfortunately, I got lost in trying to manually drop the file from DROPPER since my basic dynamic analysis did not run properly in the beginning. I only found out it had not run properly on Sunday, but by then I had too little time to more closely examine the file for effective signatures. However, the signature

of the THOR APT Scanner mentioned earlier might be useful, also to detect future samples.