APT28 and Upcoming Elections: Evidence of Possible Interference (Part II)

blog.yoroi.company/research/apt28-and-upcoming-elections-possible-interference-signals-part-ii/

April 18, 2019

Introduction

The uncertain attribution of the Ukrainian themed malicious document discussed in our past article “*APT28 and Upcoming Elections: Possible Interference Signals*”, led us to a review of Sofacy’s phishing techniques to confirm or deny the possible involvement of Russian state-sponsored actors in the election interference. We ended up in an old fake Hotel reservation request form, containing dummy interactive text boxes used to lure the victims to enable the macro code execution.

We analyzed this sample two years ago and we linked it to a Sofacy attack operation discovered by FE researchers in the mid of 2017, which hit several hotels in European and Middle Eastern countries.

Technical Analysis

Sha256	a4a455db9f297e2b9fe99d63c9d31e827efb2cda65be445625fa64f4fce7f797
Threat	APT28 GAMEFISH
Brief Description	GAMEFISH document dropper (reference sample, 2017)
Ssdeep	1536:009J0E4v13p/gL7Jj4P9bvzKGXpIiUvh23oKRO/HhcKmFoR:fb4v13pYL7J49bvr5Iias32Jc5FoR

The macro code inside the 2017 document is password protected, just like the last suspicious document we analyzed to investigate a possible Ukraine elections interference by Russian groups. After its opening, the reference sample decodes the extracted Base64 content using a custom *“DecodeBase64”* function:

Figure 1: Custom Base64 decryption routine

The decoded content is actually a DLL file which is written into *“%AppData%\user.dat”.* After that, it will be executed through an **ASR bypass technique** (Attack Surface Reduction) allowing attackers to run new child process within the Office environment. This is the same publicly available exploit previously found into the Ukrainian sample (more details in the next section).

Figure 2: Technique used to bypass Microsoft ASR protection
In this reference sample, the *“user.dat”*’s purpose is to create two new artifacts and to set persistence through *“HKCU\Environment->UserInitMprLogonScript”*. The created files are:

- %AppData%\mrset.bat
- %AppData%\mvtband.dat

Figure 3: Persistence setting and artifacts creation by *“user.dat”* file
The *“mrset.bat”* file is a short bash file, designed to check the *“mvtband.dat”* existence and to run it through *“rundll32.exe”* system utility.

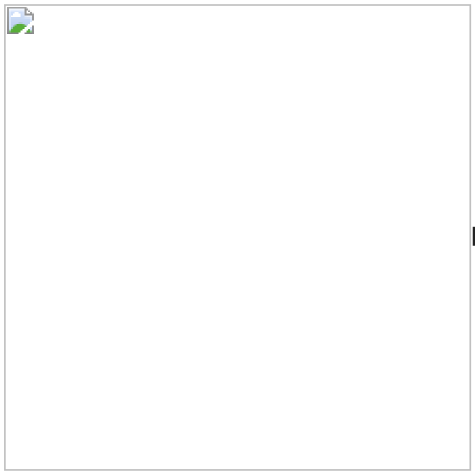
Figure 4: *“mrset.bat”* file code

Finally, the *“mvtband.dat”* file, which actually is a Delphi DLL library, is a well-known malware named “GAMEFISH” (f9fd3f1d8da4ffd6a494228b934549d09e3c59d1). Russian groups were used to use it in recon-phases to steal information from victim machine and to implant new payloads.

Figure 5: Information retrieved by mvtband.dll

Comparison with Ukrainian Elections Sample

Sha256	8a35b6ecdf43f42dbf1e77235d6017faa70d9c68930bdc891d984a89d895c1e7
Threat	Document dropper
Brief Description	Dropper of APT28 sample (Ukraine elections, 2019)
Ssdeep	12288:hRd2KFJ7uq9U2Gaz6L2qJnlIzeTLC7m0HmhVyqZPY/q7rB:/RFwqK21VolI6TL0m0GhM6LF

Despite some differences between the "Hospitality campaign" vector and the Ukraine elections one, both use similar TTP related to the APT28 group. The link between Hospitality malware and the "FancyBear" actor has been already sifted by Info-Sec community. So, we can exploit the similarities between it and the Ukrainian elections sample to link it to Russian hacker groups.

Both documents under analysis use protected macro code. All the code inside the macro is not obfuscated in any way: Hospitality document surprisingly contains code comments too. Moreover, the main macro function name is *“Execute”* for both documents and the ASR trick used to create new processes from the Office work-space is substantially the same.

Figure 6. The Ukraine elections macro on the left; Hospitality's one on the right.
In both cases the real payload is encoded in Base64 and it is stored into an Office hidden section: the first sample uses a document property, the second one employs an XML resource.

The next stages are different: the Ukraine sample deploys some Powershell obfuscated scripts, which at the end carry an Empire stager, allowing the attackers to directly interact with the victim machine; the reference sample, instead, implants the GAMEFISH malware which automatically exfiltrates victim information while waiting for new payloads to install.

Conclusion

Finally, the attribution of the Ukraine elections sample (highlighted in our previous report) can be confirmed due to the strong similarities with the first stage of the Sofacy's Hospitality malware, because:

- Both use password protection.
- Both have the same function name.
- Both have the same macro code structure.
- Both embeds the real payload in a hidden document section.
- The ASR trick is implemented using the same instructions.

The presence of these similarities between the droppers indicates, with high probability, the attacker is the same and consequentially suggests APT28 is reusing some 2017 tricks and code snippets which, despite their simplicity, make their attacks effective.

Indicators of Compromise

Hash:
- b40cbf38284e6a1b9157002ad564e40fad2d85ba36437cf95c3b6326ad142520
- 353aa1f03b36ee51138b61ef1f91f75de01850d73d619bbe5a0050594eba660d
- 58b223f74992f371cab8f1df7c03b9b66f2ea9e3c9e22122898a9be62a05c0b4
- 51eaf3b30c1ea932843cb9f5b6fb41804976d94a53a507ccb292b8392276cfd6
- 8c47961181d9929333628af20bdd750021e925f40065374e6b876e3b8afbba57
- e259df89e065c4162b273ebb18b75ea153f9bafe30a8c6610204ccf5e3f4ebcd

Yara Rules

```
rule APT28_office_document_dropper_GAMEFISH {
    meta:
         description = "Yara Rule for office_document dropper (2017)"
         author = "ZLAB Yoroi-Cybaze"
         last_updated = "2019-04-16"
         tlp = "white"
         category = "informational"
    strings:
         $a = "word\\vbaProject.binPK"
         $b = {E3 5D B8 1E 9C C7 11 F4 1E}
         $c = {36 B7 DD E9 6F 33 4b D7 E7 7F}
    condition:
         all of them
}

import "pe"
rule APT28_user_dll {
    meta:
         description = "Yara Rule for user_dll (2017)"
         author = "ZLAB Yoroi-Cybaze"
         last_updated = "2019-04-16"
         tlp = "white"
         category = "informational"
    strings:
         $a = "MZ"
         $b = "GetEnvironmentVariable"
         $c = {49 73 50 72 6F 63 65 73 73 6F 72}
    condition:
         all of them and pe.number_of_sections == 5
}

rule APT28_mrset_bat {
    meta:
         description = "Yara Rule for mrset_bat_file (2017)"
         author = "ZLAB Yoroi-Cybaze"
         last_updated = "2019-04-16"
         tlp = "white"
         category = "informational"
    strings:
         $a = "inst_pck"
         $b = "mvtband.dat"
    condition:
         all of them
}

import "pe"
rule APT28_mvtband_dat_dll {
    meta:
         description = "Yara Rule for mvtband_dat_dll (2017)"
         author = "ZLAB Yoroi-Cybaze"
         last_updated = "2019-04-16"
         tlp = "white"
         category = "informational"
    strings:
         $a = "DGMNOEP"
         $b = {C7 45 94 0A 25 73 30 8D 45 94} // two significant instructions

    condition:
         all of them and pe.sections[2].raw_data_size == 0 and pe.version_info["OriginalFilename"]
contains "mvtband"
}
```