

The return of the BOM

By GReAT

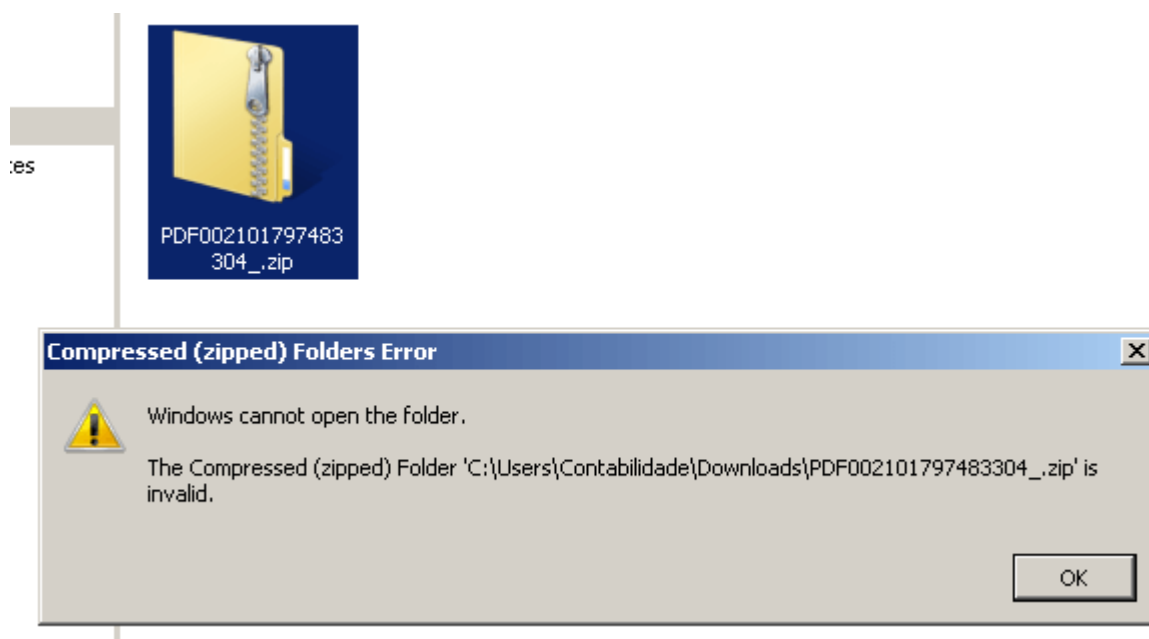
Published: 2019-03-28 · Archived: 2026-04-05 15:50:52 UTC

There's nothing new in Brazilian cybercriminals trying out new ways to stay under the radar. It's just that this time around the bad guys have started using a method that was reported in the wild years ago.

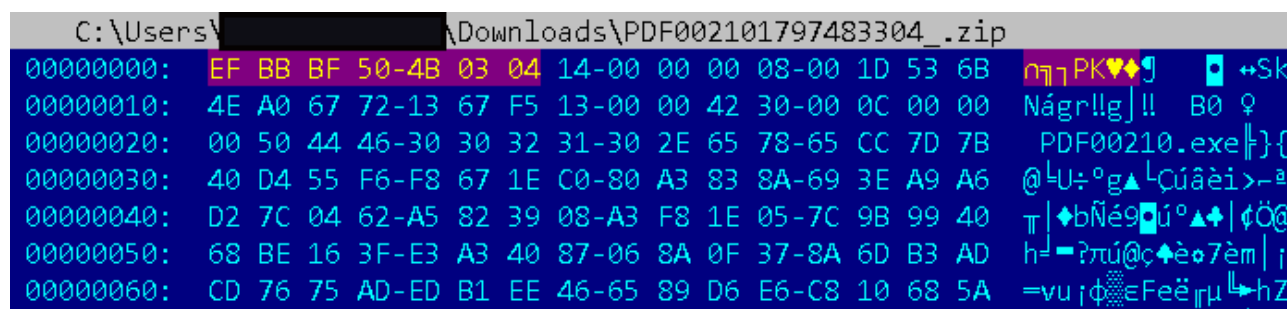
Russian gangs used this technique to distribute malware capable of modifying the hosts file on Windows systems. [Published by McAfee in 2013](#), the UTF-8 BOM (Byte Order Mark) additional bytes helped these malicious crews avoid detection.

Since these campaigns depended on spear phishing to increase the victim count, the challenge was to fool email scanners and use a seemingly corrupted file that lands in the victim's inbox.

The first indicator appears when the user tries to open the ZIP file with the default file explorer and sees the following error:



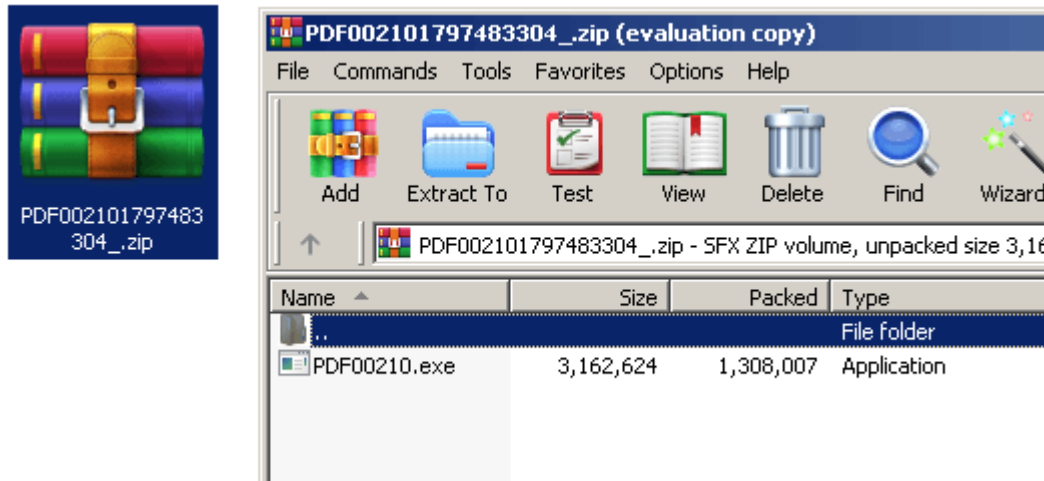
The error message suggests the file is corrupt, but when we check its contents we see something strange in there.



Zip header prefixed by UTF-8 BOM

Instead of having the normal ZIP header starting with the “PK” signature (0x504B), we have three extra bytes (0xEFBBBF) that represent the Byte Order Mark (BOM) usually found within UTF-8 text files. Some tools will not recognize this file as being a ZIP archive format, but will instead recognize it as an UTF-8 text file and fail to extract the malicious payload.

However, utilities such as WinRAR and 7-Zip ignore this data and extract the content correctly. Once the user extracts the file with any of these utilities they can execute it and infect the system.

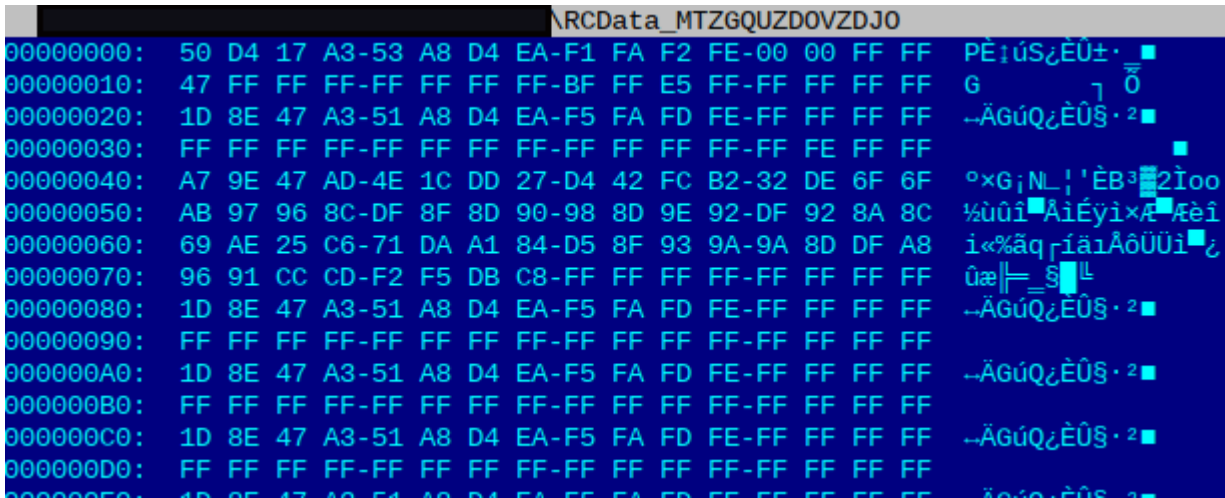


The file is successfully extracted by WinRAR

The malicious executable acts as a loader for the main payload which is embedded in the resource section.

RCDATA	16	0	DVCLAL
RCDATA	> 732160	1046	MTZGQUZDOVZDJO
RCDATA	1812	0	PACKAGEINFO
RCDATA	1172	0	TLOGINDIALOG
RCDATA	1186	1033	TMSAOCLOSE
RCDATA	1186	1033	TMSAODOWN
RCDATA	1270	1033	TMSAODOWNLAST

Resource table showing the resource containing the encrypted data



Encrypted DLL stored in resource section

The content stored inside the resource, encrypted with a XOR-based algorithm, is commonly seen in different malware samples from Brazil. The decrypted resource is a DLL that will load and execute the exported function “BICDAT”.

```

dll_image = (int)load_code(&dw0rd_516F84, dw0rd_516F88, dw0rd_516F88 >> 31);
HIDWORD(v19) = &savedregs;
LODWORD(v19) = &loc_476801;
v18 = __readfsdword(0);
__writefsdword(0, (unsigned int)&v18);
if ( !dll_image )
    Sysutils::Abort();
p_BICDAT = (int (__cdecl *) (_DWORD))get_func_addr((int)&dll_image, "BICDAT");
if ( !p_BICDAT )
    Sysutils::Abort();
v17 = a6;
v16 = System::__linkproc__ LStrToPChar(a5);
v15 = System::__linkproc__ LStrToPChar(a4);
System::__linkproc__ LStrToPChar(v25);
System::__linkproc__ LStrToPChar(v26);
args = System::__linkproc__ LStrToPChar(v27);
p_BICDAT(args);
__writefsdword(0, v14);
if ( dll_image )
    unload_image(&dll_image);
}
    
```

Code used to load the extracted DLL and execute the exported function BICDAT

This library will then download a second stage payload which is a password-protected ZIP file and encrypted with the same function as the embedded payload. After extracting all the files, the loader will then launch the main executable.

```
download_file_0(v21, v63);
unknown_libname_907(*(Forms::TApplication **)off_4A2754[0]);
Sysutils::GetEnvironmentVariable((const int)&str_TEMP[1], &v58);
System::_linkproc__ LStrCatN(&v59, 3, v12, &str__28[1], dword_4A4488);
v13 = (_BYTE *)System::_linkproc__ LStrToPChar(v59);
unknown_libname_70((int)&v60, v13);
v22 = v60;
Sysutils::GetEnvironmentVariable((const int)&str_TEMP[1], &v55);
System::_linkproc__ LStrCatN(&v56, 3, v14, &str__28[1], dword_4A4488);
v15 = (_BYTE *)System::_linkproc__ LStrToPChar(v56);
unknown_libname_70((int)&v57, v15);
decrypt_file(v57, v22);
unknown_libname_907(*(Forms::TApplication **)off_4A2754[0]);
unknown_libname_907(*(Forms::TApplication **)off_4A2754[0]);
Sleep(0x280Du);
Sysutils::GetEnvironmentVariable((const int)&str_TEMP[1], &v52);
System::_linkproc__ LStrCatN(&v53, 3, v16, &str__28[1], dword_4A4488);
v17 = (_BYTE *)System::_linkproc__ LStrToPChar(v53);
unknown_libname_70((int)&v54, v17);
v23 = v54;
get_programfiles_dir(&v51);
System::_linkproc__ LStrCat(&v51, dword_4A4494);
extract_zip(v23, v51, &str__[1]);
Sleep(0x2F79u);
unknown_libname_907(*(Forms::TApplication **)off_4A2754[0]);
get_programfiles_dir(&v49);
System::_linkproc__ LStrCat(&v49, dword_4A4494);
read_ini(v49, &v50);
Sleep(0x1039u);
get_programfiles_dir(&v47);
System::_linkproc__ LStrCat(&v47, dword_4A4494);
sub_498188(v47, &v48);
Sleep(0x870u);
get_programfiles_dir(&v46);
System::_linkproc__ LStrCat(&v46, dword_4A4494);
run_executable(v46);
unknown_libname_907(*(Forms::TApplication **)off_4A2754[0]);
Sleep(0x1528u);
Sysutils::GetEnvironmentVariable((const int)&str_TEMP[1], &v44);
System::_linkproc__ LStrCatN(&v45, 3, v18, &str__28[1], dword_4A4488);
lpFileName = (const CHAR *)System::_linkproc__ LStrToPChar(v45);
DeleteFileA(lpFileName);
Sleep(0x1528u);
```

Code executed by BICDAT function

```

D... p _TKaliputcus_Timer8Tim... call decrypt_str; Banco Safra
D... p _TKaliputcus_RtcPCusto... call decrypt_str; IniciarrecorteSelect
D... p _TKaliputcus_RtcPCusto... call decrypt_str; handleDomouseCONtrolaPosicao
D... p _TKaliputcus_RtcPCusto... call decrypt_str; MENSAGEMDO_
D... p _TKaliputcus_RtcPCusto... call decrypt_str; TRAVASITE |Sicoob
D... p _TKaliputcus_RtcPCusto... call decrypt_str; TRAVASITE |Banco do Nordeste
D... p _TKaliputcus_RtcPCusto... call decrypt_str; TRAVASITE |Sicredi
D... p _TKaliputcus_RtcPCusto... call decrypt_str; TRAVASITE |Safra
D... p _TKaliputcus_RtcPCusto... call decrypt_str; TRAVASITE |DESCO
D... p _TKaliputcus_RtcPCusto... call decrypt_str; TRAVASITE |C3F
D... p _TKaliputcus_RtcPCusto... call decrypt_str; TRAVASITE |AMARELO
D... p _TKaliputcus_RtcPCusto... call decrypt_str; BLOCK24HSAPP
D... p _TKaliputcus_RtcPCusto... call decrypt_str; TRAVASITE |ITA
D... p _TKaliputcus_RtcPCusto... call decrypt_str; TRAVASITE |Santander
D... p _TKaliputcus_RtcPCusto... call decrypt_str; Monitor_tipo_ON
D... p _TKaliputcus_RtcPCusto... call decrypt_str; Monitor ligado! bata mover mouse!
D... p _TKaliputcus_RtcPCusto... call decrypt_str; Monitor_tipo_OFF
D... p _TKaliputcus_RtcPCusto... call decrypt_str; Monitor desligado!
D... p _TKaliputcus_RtcPCusto... call decrypt_str; CapespecialWIN7
D... p _TKaliputcus_RtcPCusto... call decrypt_str; Windows 8 e Windows 10
D... p _TKaliputcus_RtcPCusto... call decrypt_str; Windows 8
D... p _TKaliputcus_RtcPCusto... call decrypt_str; Windows 10
D... p _TKaliputcus_RtcPCusto... call decrypt_str; PEDEJANELAS
D... p _TKaliputcus_RtcPCusto... call decrypt_str; ===== | JANELAS REMOTAS |=====
D... p _TKaliputcus_RtcPCusto... call decrypt_str; ===== | FIM |=====
D... p _TKaliputcus_RtcPCusto... call decrypt_str; DELEGADO
D... p _TKaliputcus_RtcPCusto... call decrypt_str; DELETAKL
D... p _TKaliputcus_RtcPCusto... call decrypt_str; Affinity1
D... p _TKaliputcus_RtcPCusto... call decrypt_str; Affinity0
D... p _TKaliputcus_RtcPCusto... call decrypt_str; MINIMIZATDSJANELAS
D... n _TKaliputcus_RtcPCusto... call decrypt_str; SFMPREFONI INFON

```

Strings related to Banking RAT malware

The final payload that's delivered is a variant of a [Banking RAT malware](#), which is currently widespread in Brazil and Chile.

Kaspersky Lab products can extract and analyze compressed ZIP files containing the Byte Order Mark without any problem.

Indicators of compromise

- 087b2d745bc21cb1ab7feb6d3284637d
- 3f910715141a5bb01e082d7b940b3552
- 60ce805287c359d58e9afd90c308fcc8
- c029b69a370e1f7b3145669f6e9399e5

Source: <https://securelist.com/the-return-of-the-bom/90065/>