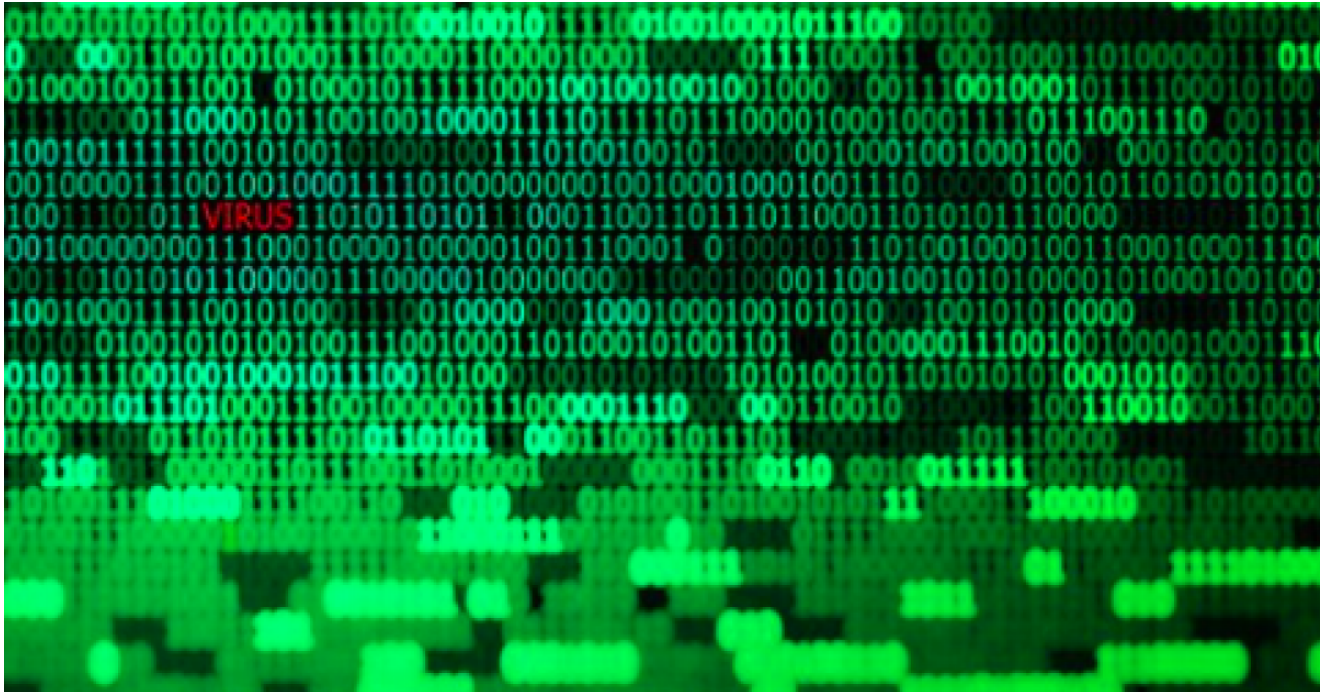


GuLoader: A Popular New VB6 Downloader that Abuses Cloud Services

 proofpoint.com/us/threat-insight/post/guloder-popular-new-vb6-downloader-abuses-cloud-services

March 5, 2020





[Blog](#)

[Threat Insight](#)

GuLoader: A Popular New VB6 Downloader that Abuses Cloud Services



March 05, 2020 Proofpoint Threat Research Team

Proofpoint researchers have observed a new downloader in the wild that we and other researchers are calling “GuLoader.” Our researchers first observed GuLoader in late December 2019 being used to deliver Parallax RAT, which itself had recently been released. While we regularly observe new loaders, GuLoader has gained popularity quickly and is in active use by multiple threat actors. GuLoader is a downloader, written partly in VB6, which typically stores its encrypted payloads on Google Drive or Microsoft OneDrive (underscoring that threat actors continue to adopt the cloud just like legitimate businesses are).

GuLoader is a portable executable (PE) file that is often observed embedded in a container file such as an .iso or .rar file. We have also observed it being downloaded directly from various cloud hosting platforms. GuLoader is used predominantly to download remote access Trojans (RATs) and information stealers such as Agent Tesla/Origin Logger, FormBook, NanoCore RAT, Netwire RAT, Remcos RAT, Ave Maria/Warzone RAT and Parallax RAT.

Analysis

The GuLoader executable is a Visual Basic 6 wrapper which decrypts (XORing with a DWORD, 4-byte key) some shellcode containing the main functionality.

```
1 0x00000000 (06) 81ec00020000 sub esp, 0x200
2 0x00000006 (01) 55 push ebp
3 0x00000007 (02) 89e5 mov ebp, esp
4 0x00000009 (05) e800000000 call 0xe
5 0x0000000e (01) 58 pop eax
6 0x0000000f (03) 83e80e sub eax, 0xe
7 0x00000012 (03) 894544 mov dword ptr [ebp + 0x44], eax
8 0x00000015 (05) e840240000 call 0x245a
9 0x0000001a (06) 64a130000000 mov eax, dword ptr fs:[0x30]
10 0x00000020 (02) 85c0 test eax, eax
11 0x00000022 (03) 8b400c mov eax, dword ptr [eax + 0xc]
12 0x00000025 (03) 8b4014 mov eax, dword ptr [eax + 0x14]
13 0x00000028 (02) 8b00 mov eax, dword ptr [eax]
14 0x0000002a (03) 8b5828 mov ebx, dword ptr [eax + 0x28]
15 0x0000002d (07) 817b0c33003200 cmp dword ptr [ebx + 0xc], 0x320033
16 0x00000034 (02) 75f2 jne 0x28
17 0x00000036 (05) 66837b102e cmp word ptr [ebx + 0x10], 0x2e
```

The loader uses sophisticated injection techniques to make analysis difficult. For example, it

1. spawns a child process copy of itself (in suspended state)
2. maps the image of a system DLL (typically "msvbvm60.dll" or "mstsc.exe") over the child at 0x400000 (instead of a normal high load address)
3. injects the unpacking code into the child
4. modifies a register within the context of the suspended child thread to redirect execution into the injected code
5. resumes the child
6. the child overwrites the system DLL image at 0x400000 with the unpacked code

The downloaded files consist of 64 hex digits followed by a PE executable encoded with XOR, where the XOR key is stored in the shellcode.

```
461 00001cc0 ff ff 53 48 43 72 65 61 74 65 44 69 72 65 63 74 |..SHCreateDirect|
462 00001cd0 6f 72 79 45 78 57 00 e8 2f f9 ff ff 53 68 65 6c |oryExW./...Shel|
463 00001ce0 6c 45 78 65 63 75 74 65 57 00 e8 7e f4 ff ff bd |lExecuteW...|
464 00001cf0 cb a2 1d 68 43 f5 9c a7 0c 99 af f4 27 8e d0 91 |...hC.....'|...|
465 00001d00 4e d4 42 3c c6 6b c1 7c 90 0f 91 c8 aa 04 f6 66 |N.B<.k.|.....f|
466 00001d10 d1 4a 23 10 4a 9d e6 50 13 41 b6 9c 2d 36 d7 3a |.J#.J..P.A..-6.:|
467 00001d20 55 7d 49 e4 11 14 0b 24 96 b8 db 71 b0 ad fc 0e |U)I....$.q....|
468 00001d30 1c f3 6e b9 94 46 ed f8 5d ea 00 45 34 df 21 9e |...n..F...E4.!.|
469 00001d40 9f 25 4f 8d 17 78 12 89 e1 60 e2 19 fb 55 03 73 |.%.x...U.s|
470 00001d50 22 57 74 61 9b ee 37 5d 64 92 07 ed 7e 87 28 47 |"Wta..7)d...~.(G|
471 00001d60 a6 ce 9a 35 1e 21 18 31 e7 09 2c 7e 01 fe 4d 1b |...S!.1...~.M.|
472 00001d70 29 00 7b 0a a1 97 3e 05 6a 3b 0d 52 85 30 2e ef |).{...>.j;.R.O..|
473 00001d80 f0 76 a0 de 68 c9 63 da 32 6d 33 26 08 62 54 c4 |.v..h.c.2m3&.bT.|
474 00001d90 73 a8 c5 b2 ec 3f 44 ae b5 e3 58 fa 8b d8 79 98 |s....?D...X...y.|
475 00001da0 f7 1f eb 86 6f 72 69 82 38 16 39 cf 52 0b 5a 6c |....ori.8.9.R.Zl|
476 00001db0 7a 51 cc 5b f2 a4 8f 56 bb 8c 5e a3 d6 81 7f 40 |zQ.[...V..^....@|
477 00001dc0 fd 83 f1 2f 75 1a 70 2b 3f be 84 77 59 b3 a5 15 |.../u.p+?.wY...|
478 00001dd0 80 f9 16 03 3d 4c 95 ff 06 34 65 4b dc 29 86 e9 |....=L...4eK)...|
479 00001de0 48 2c f8 93 c0 c3 ba d3 89 67 8a 20 60 5c ab bd |H,.....g. `\.|
480 00001df0 cb a2 1d 68 43 f5 9c a7 0c 99 af f4 27 8e d0 91 |...hC.....'|...|
481 00001e00 4e d4 42 3c c6 6b c1 7c 90 0f 91 c8 aa 04 f6 66 |N.B<.k.|.....f|
482 00001e10 d1 4a 23 10 4a 9d e6 50 13 41 b6 9c 2d 36 d7 3a |.J#.J..P.A..-6.:|
483 00001e20 55 7d 49 e4 11 14 0b 24 96 b8 db 71 b0 ad fc 0e |U)I....$.q....|
484 00001e30 1c f3 6e b9 94 46 ed f8 5d ea 00 45 34 df 21 9e |...n..F...E4.!.|
485 00001e40 9f 25 4f 8d 17 78 12 89 e1 60 e2 19 fb 55 03 73 |.%.x...U.s|
486 00001e50 22 57 74 61 9b ee 37 5d 64 92 07 ed 7e 87 28 47 |"Wta..7)d...~.(G|
487 00001e60 a6 ce 9a 35 1e 21 18 31 e7 09 2c 7e 01 fe 4d 1b |...S!.1...~.M.|
488 00001e70 29 00 7b 0a a1 97 3e 05 6a 3b 0d 52 85 30 2e ef |).{...>.j;.R.O..|
489 00001e80 f0 76 a0 de 68 c9 63 da 32 6d 33 26 08 62 54 c4 |.v..h.c.2m3&.bT.|
490 00001e90 73 a8 c5 b2 ec 3f 44 ae b5 e3 58 fa 8b d8 79 98 |s....?D...X...y.|
491 00001ea0 f7 1f eb 86 6f 72 69 82 38 16 39 cf 52 0b 5a 6c |....ori.8.9.R.Zl|
492 00001eb0 7a 51 cc 5b f2 a4 8f 56 bb 8c 5e a3 d6 81 7f 40 |zQ.[...V..^....@|
493 00001ec0 fd 83 f1 2f 75 1a 70 2b 3f be 84 77 59 b3 a5 15 |.../u.p+?.wY...|
494 00001ed0 80 f9 16 03 3d 4c 95 ff 06 34 65 4b dc 29 86 e9 |....=L...4eK)...|
495 00001ee0 48 2c f8 93 c0 c3 ba d3 89 67 8a 20 60 5c ab bd |H,.....g. `\.|
496 00001ef0 cb a2 1d 68 43 f5 9c a7 0c 99 af f4 27 8e d0 91 |...hC.....'|...|
497 00001f00 4e d4 42 3c c6 6b c1 7c 90 0f 91 c8 aa 04 f6 66 |N.B<.k.|.....f|
498 00001f10 d1 4a 23 10 4a 9d e6 50 13 41 b6 9c 2d 36 d7 3a |.J#.J..P.A..-6.:|
499 00001f20 55 7d 49 e4 11 14 0b 24 96 b8 db 71 b0 ad fc 0e |U)I....$.q....|
500 00001f30 1c f3 6e b9 94 46 ed f8 5d ea 00 45 34 df 21 9e |...n..F...E4.!.|
501 00001f40 9f 25 4f 8d 17 78 12 89 e1 60 e2 19 fb 55 03 73 |.%.x...U.s|
502 00001f50 22 57 74 61 9b ee 37 5d 64 92 07 d9 d0 55 89 e5 |"Wta..7)d...U..|
503 00001f60 8b 45 08 39 c0 8b 5d 0c d9 d0 50 53 53 e8 34 01 |.E.9...PSS.4.|
```

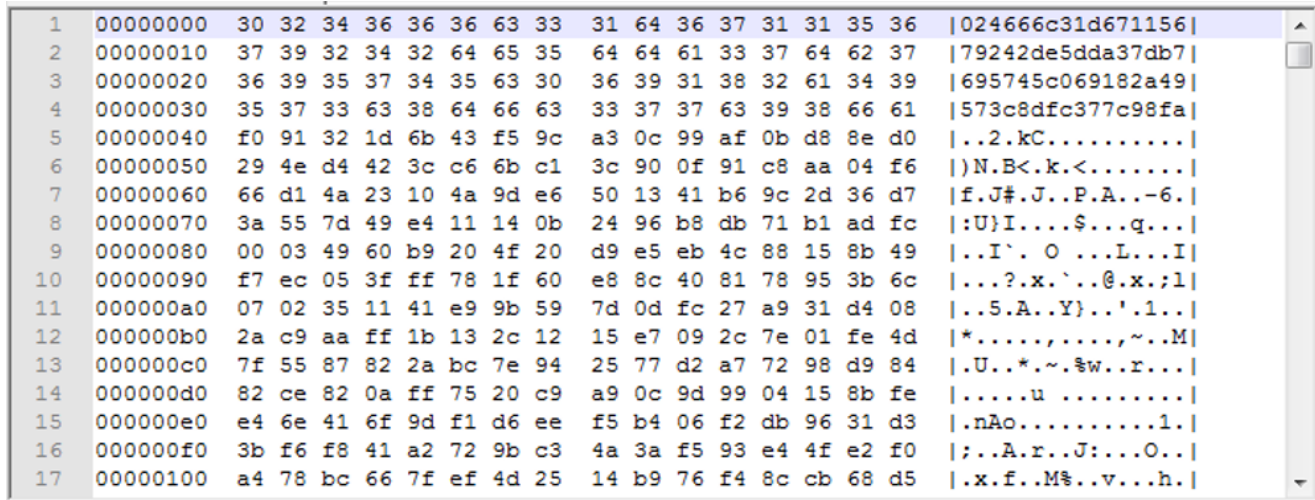
Payload Encoding

The payload URI paths (other than Google Drive or OneDrive ones) and downloaded filename frequently have the form "<something>_encrypted_XXXXXX.bin" where "XXXXXXX" are hexadecimal digits.

The downloaded payloads consist of

* 64 lower-case hex digits

* the XORed PE binary



1	00000000	30 32 34 36 36 36 63 33	31 64 36 37 31 31 35 36	024666c31d671156
2	00000010	37 39 32 34 32 64 65 35	64 64 61 33 37 64 62 37	79242de5dda37db7
3	00000020	36 39 35 37 34 35 63 30	36 39 31 38 32 61 34 39	695745c069182a49
4	00000030	35 37 33 63 38 64 66 63	33 37 37 63 39 38 66 61	573c8dfc377c98fa
5	00000040	f0 91 32 1d 6b 43 f5 9c	a3 0c 99 af 0b d8 8e d0	..2.kC.....
6	00000050	29 4e d4 42 3c c6 6b c1	3c 90 0f 91 c8 aa 04 f6)N.B<.k.<.....
7	00000060	66 d1 4a 23 10 4a 9d e6	50 13 41 b6 9c 2d 36 d7	f.J#.J..P.A..-6.
8	00000070	3a 55 7d 49 e4 11 14 0b	24 96 b8 db 71 b1 ad fc	:U}I...\$....q...
9	00000080	00 03 49 60 b9 20 4f 20	d9 e5 eb 4c 88 15 8b 49	..I`. O ...L...I
10	00000090	f7 ec 05 3f ff 78 1f 60	e8 8c 40 81 78 95 3b 6c	...?.x.`...@.x.;l
11	000000a0	07 02 35 11 41 e9 9b 59	7d 0d fc 27 a9 31 d4 08	..5.A..Y}...'..1..
12	000000b0	2a c9 aa ff 1b 13 2c 12	15 e7 09 2c 7e 01 fe 4d	*.....,.....~..M
13	000000c0	7f 55 87 82 2a bc 7e 94	25 77 d2 a7 72 98 d9 84	.U..*..~.~w..r...
14	000000d0	82 ce 82 0a ff 75 20 c9	a9 0c 9d 99 04 15 8b feu
15	000000e0	e4 6e 41 6f 9d f1 d6 ee	f5 b4 06 f2 db 96 31 d3	.nAo.....1..
16	000000f0	3b f6 f8 41 a2 72 9b c3	4a 3a f5 93 e4 4f e2 f0	;..A.r..J:...O..
17	00000100	a4 78 bc 66 7f ef 4d 25	14 b9 76 f4 8c cb 68 d5	.x.f..M%.~.v...h.

The XOR key was fixed at 96 bytes in early versions of the loader

Later versions have longer keys, typically 512-768 bytes long, usually consisting of a 256-byte key repeated to give the required length. The key is stored completely in the decoded shellcode.

IOCs

Parallax Sample - 2019-12-23

SHA256: e8f8cc178425c55c03c76d0a2a11918371bba8f2d6f400752ca1cea5e663da2e

URLs: [hxxps://drive.google\[.\]com/uc?export=download&id=1dtIMCyoZUPBepc-AtEdirGENZBpWesAi](https://drive.google.com/uc?export=download&id=1dtIMCyoZUPBepc-AtEdirGENZBpWesAi)

C2: 185.140.53[.]134:7776

Remcos Sample - 2020-02-20

SHA256: 26f7bfe041a3d8a2b620d0ed2af4e2ef54b004202ec479362939b9154b1c8758

URLs: [hxxps://drive.google\[.\]com/uc?export=download&id=1N8gVOM5p8Ubm1HwolChxHidT7YoN29EE](https://drive.google.com/uc?export=download&id=1N8gVOM5p8Ubm1HwolChxHidT7YoN29EE)

C2: droptop1[.]com:2500

C2: droptop2[.]com:2500

C2: droptop3[.]com:2500

C2: droptop4[.]com:2500

C2: droptop5[.]com:2500

C2: droptop6[.]com:2500

C2: droptop7[.]com:2500

C2: droptop8[.]com:2500

C2: droptop9[.]com:2500

C2: droptop10[.]com:2500

Subscribe to the Proofpoint Blog