RESEARCH

Multiple Chinese Threat Groups Exploiting CVE-2018-0798 Equation Editor Vulnerability Since Late 2018

July 3, 2019 | Anomali Threat Research Team

During Anomali Threat Researcher's tracking of the "Royal Road" Rich Text Format (RTF) weaponizer, commonly used by multiple Chinese threat actors to exploit CVE-2017-11882 and CVE-2018-0802, it was discovered that multiple Chinese threat groups updated their weaponizer to exploit the Microsoft Equation Editor (EE) vulnerability CVE-2018-0798 late 2018. We believe the groups moved to use CVE-2018-0798 instead of the other Microsoft Equation Editor Remote Code Execution (RCE) vulnerabilities because the former is more reliable as it works on all known versions of Equation Editor.

The analyzed RTF files share the same object dimension (objw2180\objh300) used to track the RTF weaponizer in our previous report, however, the sample was not exploiting CVE-2017-11882 or CVE-2018-0802. After further analysis, it was discovered that the RTF files were exploiting the CVE-2018-0798 vulnerability in Microsoft's Equation Editor (EQNEDT32). CVE-2018-0798 does not appear to be a commonly exploited In The Wild (ITW) even though it is more reliable compared to other well-known RCE counterparts This is mainly because CVE-2018-0798 works with all EE versions while the counterparts are limited to specific versions. CVE-2017-11882 is only exploitable on an unpatched version prior to its fix, and CVE-2018-0802 is only exploitable on the version released to fix CVE-2017-11882. In contrast, a threat actor utilizing CVE-2018-0798 has a higher chance of success because it is not limited by version.

Anomali Researchers were able to identify multiple samples of malicious RTF documents ITW using the same exploit for CVE-2018-0798. Some of the analyzed samples have a creation date of November 19, 2017 (five days after a patch was released for CVE-2017-11882), however, that date appears to be incorrect because the dropped payloads had a recent compilation timestamps in 2019. The earliest use of the exploit ITW we were able to identify and confirm is a sample (e228045ef57fb8cc1226b62ada7eee9b) dating back to October 2018 (VirusTotal submission of 2018-10-29) with the RTF creation time 2018-10-23.

Multiple samples analyzed by Anomali researchers that we associate with CVE-2018-0798 were also mentioned in previous instances by other researchers in the security community. We believe that some of these were misattributed to CVE-2017-11882 or CVE-2018-0802 when they actually appear to be CVE-2018-0798.

Vulnerability and Exploit Analysis

CVE-2018-0798 is an RCE vulnerability, a stack buffer overflow that can be exploited by a threat actor to perform stack corruption. The vulnerable subroutine is located at the relative virtual address 0x43f6c (sub_443f6c), shown in Figure 1 below. This routine is called by EQNEDT32 when parsing Matrix type records. To note, CVE-2017-11882 and CVE-2018-0802 are vulnerabilities that take place when parsing Font records. Part of the Matrix record object is copied to a stack buffer without proper bound checks. This allows the threat actor to overflow the stack buffer, change the stored return address, and take control of the instruction pointer. Due to the age of this binary, it was compiled and linked in the early 2000s, it does not use any modern protections against stack overflows that would have made exploitation much harder.

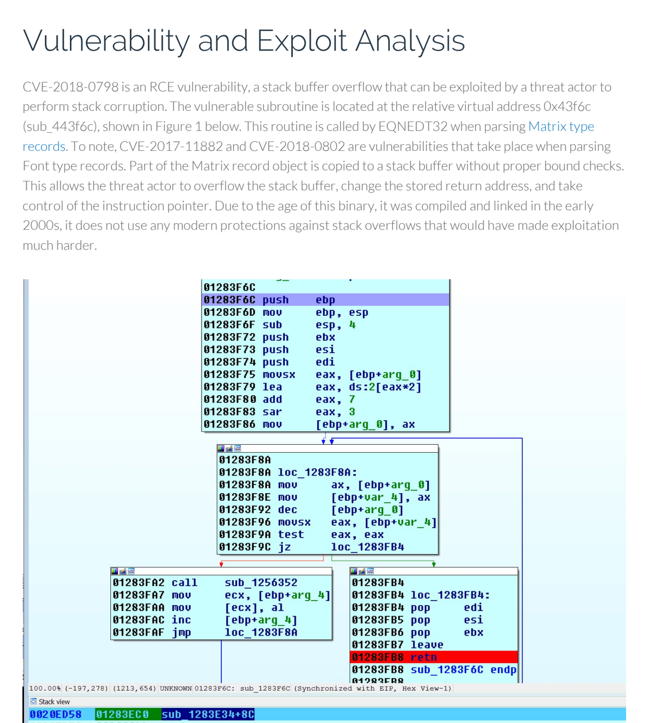

Figure 1 - The vulnerable function before the exploit. The saved return address marked in red is manipulated. Instruction at 0x1283fea copies a byte from the equation object to a stack buffer and return from the call.

The write primitive is used to fill the stack with padding 0x60s and 0x61s until the location of the stored return address and the lower two bytes of the stored return address are changed to 0x0bfb, as depicted in Figure 2. Changing those bytes allows hijackig the control flow when the return address is popped off the stack and into the instruction pointer (EIP). The instruction pointer is then redirected to the return instruction (retn) and the return instruction pops the next value off the stack, located at 0x20ed94 in Figure 2, and into the EIP. This forces the original function argument to be taken as a return address. The argument points to the heap where the equation object has been stored.

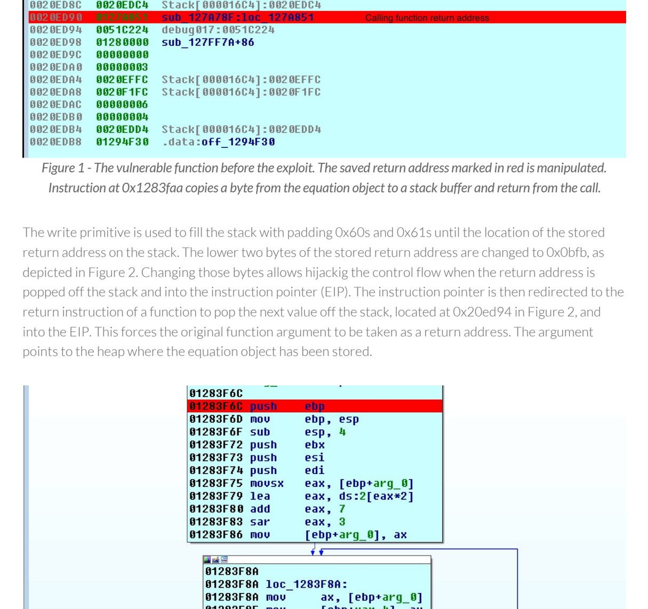

Figure 2 - The vulnerable function after stack corruption showing calling function return address last two bytes overwritten with x0bfb.

EIP lands on a Null sled until it reaches to the shellcode shown below. The shellcode pops the next value on the stack using this value the location of the final shellcode is computed.

```
debug017:0051C24D pop     eax
debug017:0051C24E jmp     short loc_51C256
---------------------------------------------------------------
debug017:0051C256 add     eax, offset byte_1BD3C
debug017:0051C25B mov     eax, [eax]
debug017:0051C25D mov     eax, [eax+14h]
debug017:0051C260 add     eax, 4Dh
debug017:0051C263 jmp     eax
```

The final shellcode in sample (264cee1c1854698ef0eb3a141912db40) is shown below. It resolves the address of WinExec and executes the PowerShell command: **powershell.exe Copy-Item "c:\target\Flag.dat" -Destination "C:\pwn"**

```
debug017:00523280 jmp     short sub_523276
---------------------------------------------------------------
debug017:00523276 push    'Acor'
debug017:0052327B push    'PteG'
debug017:00523280 call    Sub_getprocaddr
debug017:00523285 push    eax
debug017:00523286 push    'cex'
debug017:0052328B push    'EniW'
debug017:00523290 call    Sub_getprocaddr
debug017:00523295 push    0
debug017:00523297 xor     edx, edx
debug017:00523299 push    offset unk_226E77
debug017:0052329E push    'pl:C'
debug017:005232A3 push    '" no'
debug017:005232A8 push    'itan'
debug017:005232AD push    'itse'
debug017:005232B2 push    'D- "'
debug017:005232B7 push    'tad.'
debug017:005232BC push    'galF'
debug017:005232C1 push    '\teg'
debug017:005232C6 push    'rat\'
debug017:005232CB push    ':c" '
debug017:005232D0 push    'metI'
debug017:005232D5 push    '-ypo'
debug017:005232DA push    'C ex'
debug017:005232DF push    'e.ll'
debug017:005232E4 push    'ehsr'
debug017:005232E9 push    'ewop'
debug017:005232EE mov     ecx, esp
debug017:005232F0 push    edx
debug017:005232F1 push    ecx
debug017:005232F2 call    eax "winexec"
debug017:005232F4 pop     edi
debug017:005232F5 pop     esi
debug017:005232F6 pop     ebx
debug017:005232F7 add     esp, 40h
debug017:005232FA cmp     ebp, esp
debug017:005232FC call    near ptr unk_523305
debug017:00523301 mov     esp, ebp
debug017:00523303 pop     ebp
debug017:00523304 retn
```

As previously mentioned this exploit works on all known versions of Microsoft Equation Editor.

```
rule RTF_Equation_Editor_CVE_2018_0798

{
Meta:

author        = "Anomali"
    tlp           = "GREEN"
    version       = "1.0"
    date          = "2019-05-10"
    hash          = "264cee1c1854698ef0eb3a141912db40"
    description   = "Detects Malicious RTFs exploiting CVE-2018-0798"

strings:

$S1= "4460606060606060606061616161616161616161616161616161fb0b"
$RTF= "{\\rt"

condition:

$RTF at 0 and $S1
}
```

Figure 3 -Yara rule to detect Malicious RTF exploiting CVE-2018-0798

Threat Actors and Exploit Usage

Most of the collected Samples were attributed to the following Chinese Cyber Espionage threat actor:

- Conimes
- KeyBoy
- Emissary Panda
- Rancor
- Temp.Trident

However,Beginning on 25 June 2019, we started observing multiple commodity campaigns (Mostly dropping AsyncRAT) using the updated RTF weaponizer with the same exploit (CVE-2018-0798). As observed previously with CVE-2017-11882 and CVE-2018-0802, the weaponizer was used exclusively by Chinese cyber espionage actors for approximately one year (December 2017 through December 2018), after which cybercrime actors began to incorporate it in their malicious activity. This indicates that the weaponizer author is now selling to a wider group of actors.

Examples of social engineering lures and malicious document content used with CVE-2018-0798 in cyberespionage attacks are shown in Table 1.

Table 1 - Documents exploiting CVE-2018-0798

MD5	Lure Topic	File Name
019debaee6fdf9a9f872277563f0d9ee	A Mongolian language-lure themed around government policy.	APP.doc
21d0f19abd15d65aa755e89e55157ae7	Labeled "Ministry of Defence" for Mongolia. Themed around Russian President Vladimir Putin making a statement on United States' missiles.	File name is unavailable
2ef069d0e3bb636d2d969d3e6a4d5039	Pertains to be a report from the Mongolian Embassy in Japan regarding news about North Korea.	TM 30.17.doc
853136f00e87a1ab3e2fc3acb309573e	A Mongolian-language lure that contains a table with apparent details of people including email, name, and phone number.	Цэргийн багийн 8 ээлж ашиглагдах утасны дугаарын жагсаалт.doc (List of telephone numbers to be used in the 8th Military Team.doc)
ac0eac22ce12eac9ee15ca03646ed70c	Contains an image with Russian text titled about "Commonwealth of Independent States Anti-terrorist Centre".	doc.rtf
6930bd66a11e30dee1ef4f57287b1318	Titled "Social Security Reform Note". Discusses demographics and social security reform in Brazil.	Sosyal Güvenlik Reformu-Not-3.doc
8f1ab1f96b8322c9e02d87a431a98823	Titled "Foreign Office of Vietnam". Guidance on granting, extending, modifying and supplementing diplomatic passports, official passports and diplomatic note for visa application.	02_2019_TT-BNG.doc
b3f8abe274cb6a5926bd5c3fc2168997	In the Vietnamese language that appears to talk about the health of former Member of the Central Party Committee VIII, IX Nguyen Phuc Thanh.	Giay moi hoi nghi.doc
f0424ed16b435f0c7c802f3a17cbd9de	In the Vietnamese language that contains instructions for employees before taking a blood test.	PV Báo Quốc Phòng xin phỏng vấn anh.doc
7b9d386280da1b840f1b32b85ce74278	Lure in the Russian language that is a letter to rector of Russian university.	Unavailable
0764ecc46463fb10952d54515c73e6fc	Mongolian lure on topic of training and the United Nations.	uuganaa-test.doc
d648c374439cf5fe9df8dc59eb472067	Vietnamese lure themed on the current Vietnamese Prime Minister Nguyễn Xuân Phúc	TB -VPCP.doc
a94db3001c0c3fa3cf40bc7fdf9d21b7	Mongolian lure on topic of the Mongolian prime minister visiting Japan.	Medee Bolor 20181217.doc
6614a8776692c982ad766d23b2a5ea29	Russian lure linking to Russian news about NATO troops leaving Afghanistan.	Program on applied security studies.rtf
84fca27bc75f40194c95534b07838d6c	Vietnamese Police-themed lure.	Giấy tặng quà.doc

Sample Documents:

fc47442f175ff7e312a4aa4f5c8745b8

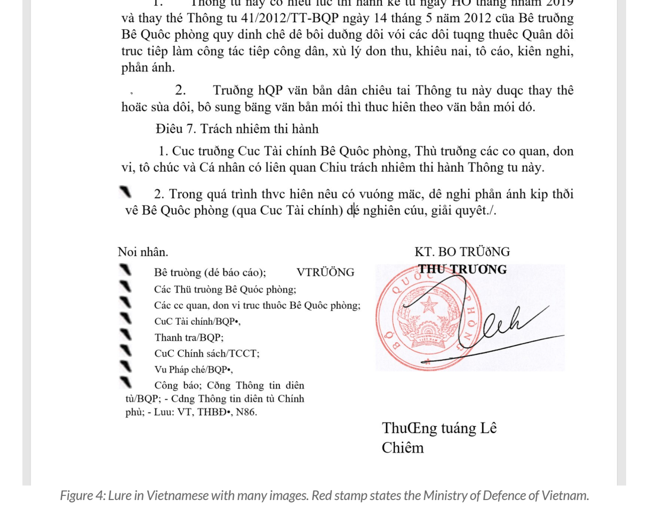

Figure 4: Lure in Vietnamese with many images. Red stamp states the Ministry of Defence of Vietnam.

40cfeb699d239652dd4a79c18b1c7366

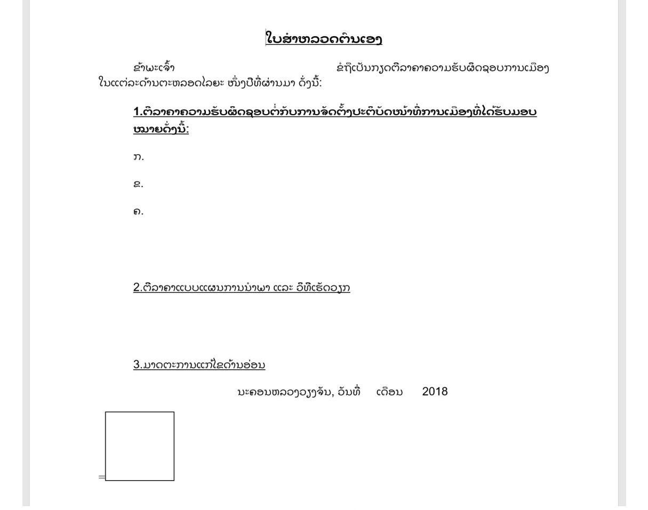

Figure 5: Lure in Lao language.

1690766e844034b3c2ab4f853bd59df7

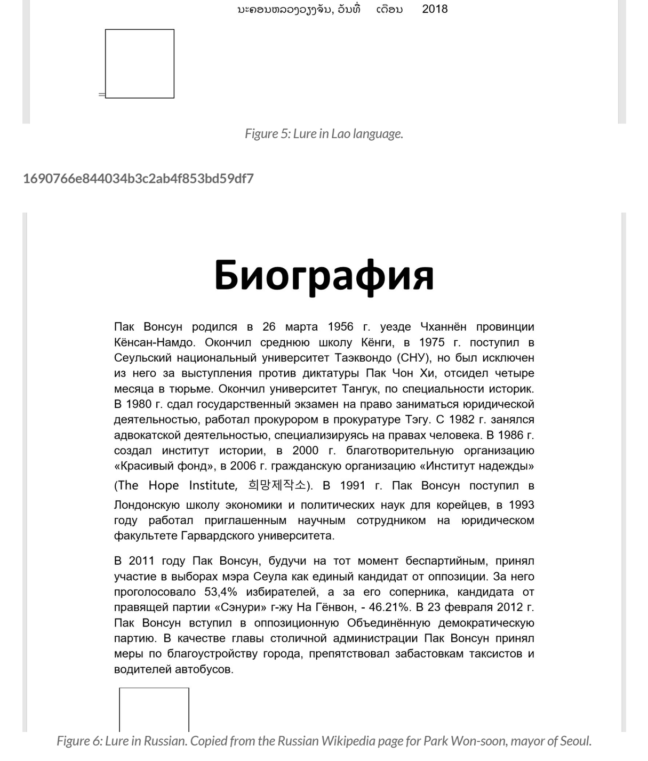

Figure 6: Lure in Russian. Copied from the Russian Wikipedia page for Park Won-soon, mayor of Seoul.

Exploitation Methods and payload Analysis:

Anomali Threat Researchers identified multiple exploitation techniques using CVE-2018-0798 to drop malicious payloads. Some of the observed techniques identified being used to exploit the vulnerability are as follows:

OLE package objects and DLL Sideloading

Sample MD5: fc47442f175ff7e312a4aa4f5c8745b8 (Goblin Panda)

The malicious RTF document contains OLE Package objects. On execution (user opening the attachment) the document drops OLE package as "8.t" in the %TEMP% directory. The 8.t file is a dropper and it is encrypted using XOR cipher with encryption key "0xFC". Upon decrypting and executing, it drops two additional files "wsc_proxy.exe" (legitimate Avast executable) and a malicious DLL "wsc.dll" in the %TEMP% folder. The dropper then creates a scheduled task to run the executable "wsc_proxy.exe" for every five minutes as a persistence mechanism.

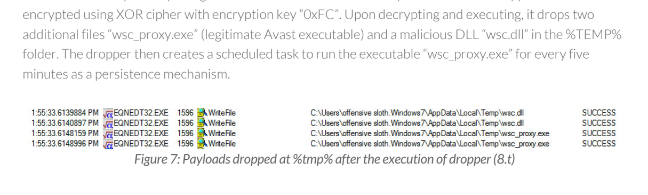

Figure 7: Payloads dropped at %tmp% after the execution of dropper (8.t)

Schedule task command:
"schtasks /create /sc MINUTE /tn "Avast Antivirus" /tr "C:\Users\Username\AppData\Local\Temp\wsc_proxy.exe" /mo 5 /f"

The benign executable "wsc_proxy.exe" gets executed by the scheduled task "Avast Antivirus," and using DLL sideloading the malicious payload "wsc.dll" gets started. The malware attempts to communicate via HTTP to the C2 at vxcvxsdvx.dynamic-dns[.]net over port 2113/TCP.

Payload MD5: 9AD1D8A92734A534B9180FB8A6821856
C2: vxcvsdvx.dynamic-dns[.]net
IP: 185.216.35[.]11 (known Goblin panda C&C)
URL: vxcvsdvx.dynamic-dns[.]net/image/logo.png
OLE package objects and VBScript Execution

Sample MD5: b3f8abe274cb6a5926bd5c3fc2168997 (Rancor Group)

The malicious RTF drops embedded OLE package to "8.t" into the %TEMP% directory after the malicious document is opened. The file 8.t is a malicious executable dropper and encrypted via XOR cipher using the key "0xFC". On execution it drops two files "ChromeApp.ps1" and "ChromeApp.vbs" in the directory "C:\Windows\tracing\". It then creates a scheduled task named "ChromeApp" to execute the Visual Basic Script (VBScript). The VBScript calls the PowerShell script and it beacons out to C2 "185.234.73[.]4" using HTTP to send the victim User ID and receiving further instructions to execute.

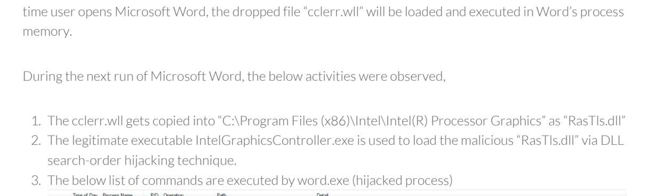

Figure 8: Scheduled task creation to start the malicious payload

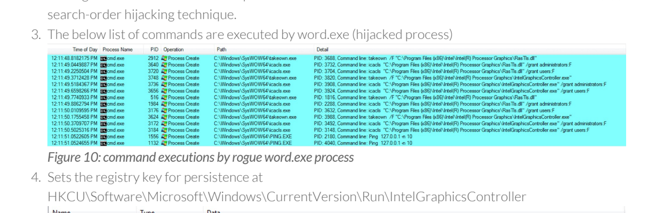

Figure 9: Malicious payloads dropped after the execution of dropper (8.t)

C2 IP: 185.234.73.4
URL: : http:185.234.73[.]4/CApp.php?name=NzI4QTRENTYtMEY0Ny0yQzY3LTY3QzEtQjg0MzNBOUU1Rjgw:VUk=

Dropping '.wll' file in Microsoft Word 'startup' folder

Sample MD5: 019debaee6fdf9a9f872277563f0d9ee

The RTF document drops embedded OLE package as "8.t" in the %TEMP% directory whe the file is opened. The shellcode decrypt "8.t" and save it as "%APPDATA%\Microsoft\Word\STARTUP\cclerr.wll". The next time user opens Microsoft Word, the dropped file "cclerr.wll" will be loaded and executed in Word's process memory.

During the next run of Microsoft Word, the below activities were observed,

1. The cclerr.wll gets copied into "C:\Program Files (x86)\Intel\Intel(R) Processor Graphics" as "RasTls.dll"
2. The legitimate executable IntelGraphicsController.exe is used to load the malicious "RasTls.dll" via DLL search-order hijacking technique.
3. The below list of commands are executed by word.exe (hijacked process)

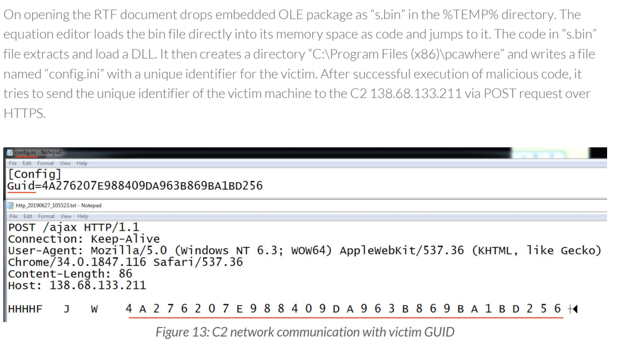

Figure 10: command executions by rogue word.exe process

4. Sets the registry key for persistence at HKCU\Software\Microsoft\Windows\CurrentVersion\Run\IntelGraphicsController

Figure 11: Windows Autorun key set for persistence.

5. *Drops two batch files in the %TEMP% folder named as UnIB490.bat & UnIB4A0.bat*
6. *The batch files are used to clean up the word document and ".wll" file.*

```
Ping 127.0.0.1 -n 10
del "C:\Users\admin\AppData\Roaming\Microsoft\Word\STARTUP\cclerr.wll" /q /f
del %0 /q /f
```

Figure 12: Batch script for clearing traces of malicious activities.

Payload MD5: B72448AF5F58E70C225AB6525126CF8B
C2: 217.69.8[.]255

Sample MD5: 6930bd66a11e30dee1ef4f57287b1318 (Emissary Panda)

On opening the RTF document drops embedded OLE package as "s.bin" in the %TEMP% directory. The equation editor loads the bin file directly into its memory space as code and jumps to it. The code in "s.bin" file extracts and load a DLL. It then creates a directory "C:\Program Files (x86)\pcawhere" and writes a file named "config.ini" with a unique identifier for the victim. After successful execution of malicious code, it tries to send the unique identifier of the victim machine to the C2 138.68.133.211 via POST request over HTTPS.

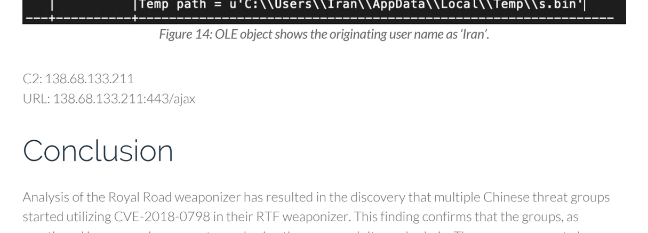

Figure 13: C2 network communication with victim GUID

The OLE object had an interesting source path "C:\\Users\\Iran\\Desktop\\s.bin"

```
File: '6930bd66a11e30dee1ef4f57287b1318.rtf' - size: 291090 bytes
---+----------+-------------------------------------------------------------
id |index     |OLE Object
---+----------+-------------------------------------------------------------
0  |00010670h |format_id: 2 (Embedded)
   |          |class name: 'Package'
   |          |data size: 97354
   |          |OLE Package object:
   |          |Filename: u's.bin'
   |          |Source path: u'C:\\Users\\Iran\\Desktop\\s.bin'
   |          |Temp path = u'C:\\Users\\Iran\\AppData\\Local\\Temp\\s.bin'
---+----------+-------------------------------------------------------------
```

Figure 14: OLE object shows the originating user name as 'Iran'.

C2: 138.68.133.211
URL: 138.68.133.211:443/ajax

Conclusion

Analysis of the Royal Road weaponizer has resulted in the discovery that multiple Chinese threat groups started utilizing CVE-2018-0798 in their RTF weaponizer. This finding confirms that the groups, as mentioned in our previous report, are sharing the same exploit supply chain. The groups appear to have been using the Microsoft vulnerability exploit exclusively for approximately six months before it began appearing in commodity-malware campaigns. This may indicate that the Chinese groups sold the exploit after using it in their malicious campaigns.

These findings also suggest that the threat groups have robust exploit developing capabilities because CVE-2018-0798 is not widely reported on and it is typically not incorporated into publicly available weaponizers.

Threatstream enterprise users can read a more detailed analysis here.

IOCs

File Hashes (MD5):
e228045ef57fb8cc1226b62ada7eee9b
019debaee6fdf9a9f872277563f0d9ee
0764ecc46463fb10952d54515c73e6fc
0827f48e883f5a59f1c4bf70c98dc42a
0e8d3ae263fae7775ccc744a5c0c4dc1
10348b56b0e3466f9f9fa62bda081c98
109d51899c832287d7ce1f70b5bd885d
1690766e844034b3c2ab4f853bd59df7
21d0f19abd15d65aa755e89e55157ae7
264cee1c1854698ef0eb3a141912db40
2868447eebdf897bdd6b7ce2a18f4609
29027a6d2a38a9a954c1e1315439baf9
2ef069d0e3bb636d2d969d3e6a4d5039
31283ad09bc7cf618c32a1c893163891
36796fabb76eb946d211a2fcf5820929
40cfeb699d239652dd4a79c18b1c7366
4642e8712c8ada8d56bd36416abb4808
47353a86ea58df3714870e5755056d97
4eb14eb23d50b4c7ee768038172f9794
51c35cb62a0ad294979b0645e5aa4376
5271a5ddf476af87c6f833638375c72f
595e30b0c794f47fd768b24ae9caf210
5982ba16356ee8118e4cdbe54d182b11
600e14e4b0035c6f0c6a344d87b6c27f
6614a8776692c982ad766d23b2a5ea29
67682e25939dce4406f55b6c0c741c0e
6930bd66a11e30dee1ef4f57287b1318
6bdc73a2fc8506d9e842fc7b7a4123db
6d2e6a61eede06fa9d633ce151208831
7b9d386280da1b840f1b32b85ce74278
827c7048c269645ce36546c01c01f93f
8408641cfbcdb53e1e6802f07ea32f11
84fca27bc75f40194c95534b07838d6c
853136f00e87a1ab3e2fc3acb309573e
8621ff472360600ec2a6f7d61a66eeb8
8f1ab1f96b8322c9e02d87a431a98823
923d60f3e63c95021f9e99f943fcfbbc
a02712c6cefb532e7928a781fe8d8592
a37df9b230c9d05210613b3c2916328f
a497426d0f65877947e92a14b8a086af
a5a4046989fa0f99c2076aec3ea0ab2a
a94db3001c0c3fa3cf40bc7fdf9d21b7
a99efd6b4b69c55774a16ae157cd20b9
ac0eac22ce12eac9ee15ca03646ed70c
af7f59b2b197d454ab8c8a7b0bc371a2
b2bce665c9bcdf0d3d04dc7ce5e30f79
b3f8abe274cb6a5926bd5c3fc2168997
b72448af5f58e70c225ab6525126cf8b
b82e0ac46f6b812c83a3954038814cce
bb7aba40c6fc76291fd1cf2c4c558e9f
bcbea5b25356d768fd826e0376268ff5
c65b73dde66184bae6ead97afd1b4c4b
d648c374439cf5fe9df8dc59eb472067
e004daf8e09b56940d6ca6e51974498b
e137b95f6149a8639f6d18e286a0a55f
f0424ed16b435f0c7c802f3a17cbd9de
f1824bd902251314a4fd5506caced48b
f1dcf1b2376360c9f0c23f1fb9f4355a
f333194c19730d6f82ab858210327051
f34514118eb4689560cd6c0c654f26d9
fc47442f175ff7e312a4aa4f5c8745b8

Network IOCs:
185.234.73[.]4
138.68.133[.]211
Vvcxvsdvx.dynamic-dns[.]net
loge.otzo[.]com

About the Author
Anomali Threat Research Team