Tracking OceanLotus' new Downloader, KerrDown

unit42.paloaltonetworks.com/tracking-oceanlotus-new-downloader-kerrdown

By Vicky Ray and Kaoru Hayashi February 1, 2019

OceanLotus (AKA APT32) is a threat actor group known to be one of the most sophisticated threat actors originating out of south east Asia. Multiple attack campaigns have been reported by number of security organizations in the last couple of years, documenting the tools and tactics used by the threat actor. While OceanLotus' targets are global, their operations are mostly active within the APAC region which encompasses targeting private sectors across multiple industries, foreign governments, activists, and dissidents connected to Vietnam.

This blog will cover a new custom downloader malware family we've named "KerrDown" which OceanLotus have been actively using since at least early 2018. We also show how the jaccard-index algorithm was used to quickly find similarities between the new KerrDown malware family within our datasets. This method has proven to be very useful to extract similarities from large sample datasets and connecting attack campaigns together. Given the large number of "KerrDown" samples found, we were also able to discern possible patterns in OceanLotus' working hours and days of a week which is discussed in the later sections of this blog.

We identified two methods to deliver the KerrDown downloader to targets. One is using the Microsoft Office Document with a malicious macro and the other is RAR archive which contains a legitimate program with DLL side-loading. For RAR archive files, the file names used to trick targets are all in Vietnamese as shown in Figure 11. Our analysis shows that the primary targets of the ongoing campaign discussed in this blog are either in Vietnam or Vietnamese speaking individuals.

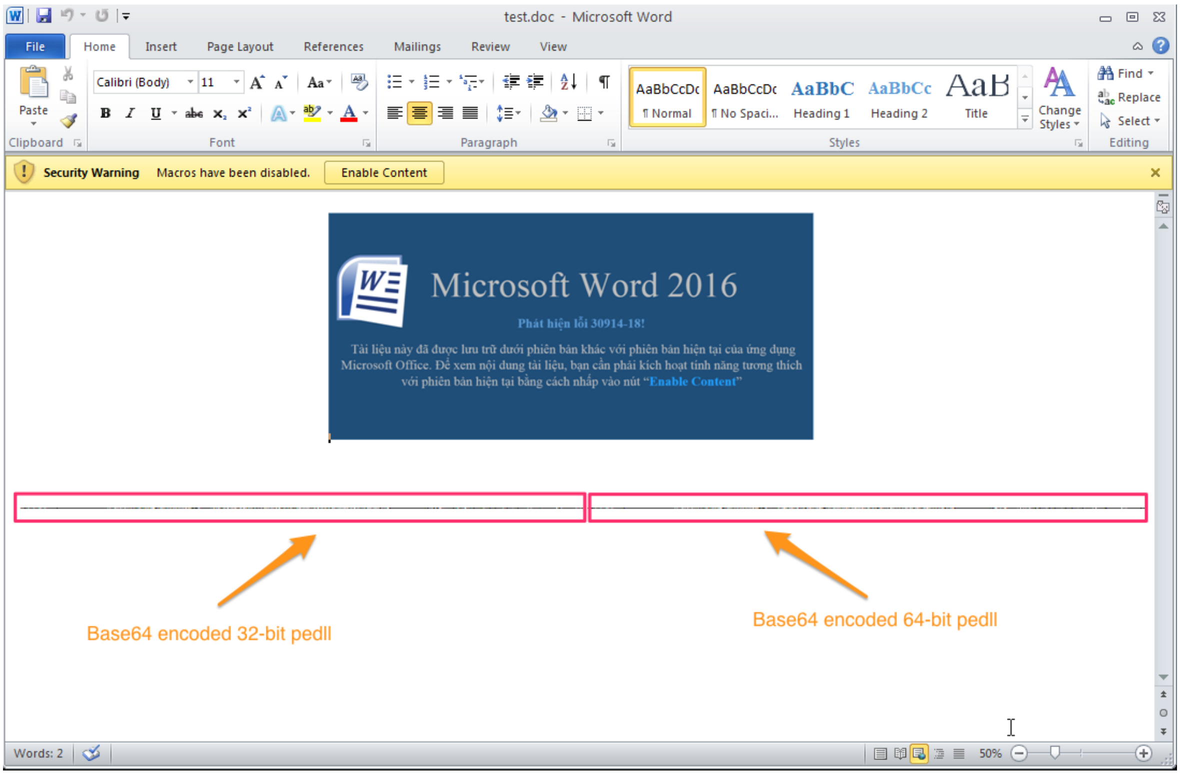
Malicious Document

Our analysis began with an active mime document, something we've seen OceanLotus use before but this time involving a new payload, KerrDown. The lure hash is

(SHA256:89e19df797481ae2d2c895bcf030fe19e581976d2aef90c89bd6b3408579bfc3)

Figure 1 below shows a snapshot of the lure file. Once the victim opens the lure document, which includes an image file with a message in Vietnamese which that asks the victim to enable macros to view the contents of the file. At first glance the document may look like there is no other content other than the notification to enable macros. However, a closer look reveals two different base64 blobs inserted in the page in separate tables and the font size has been changed to 1 which may deceive victims to overlook the content. Another reason for this technique may be that many automated tools are able to detect the presence of an embedded binary within the streams of such files and this technique may allow them to go undetected.

Delivery Document Analysis



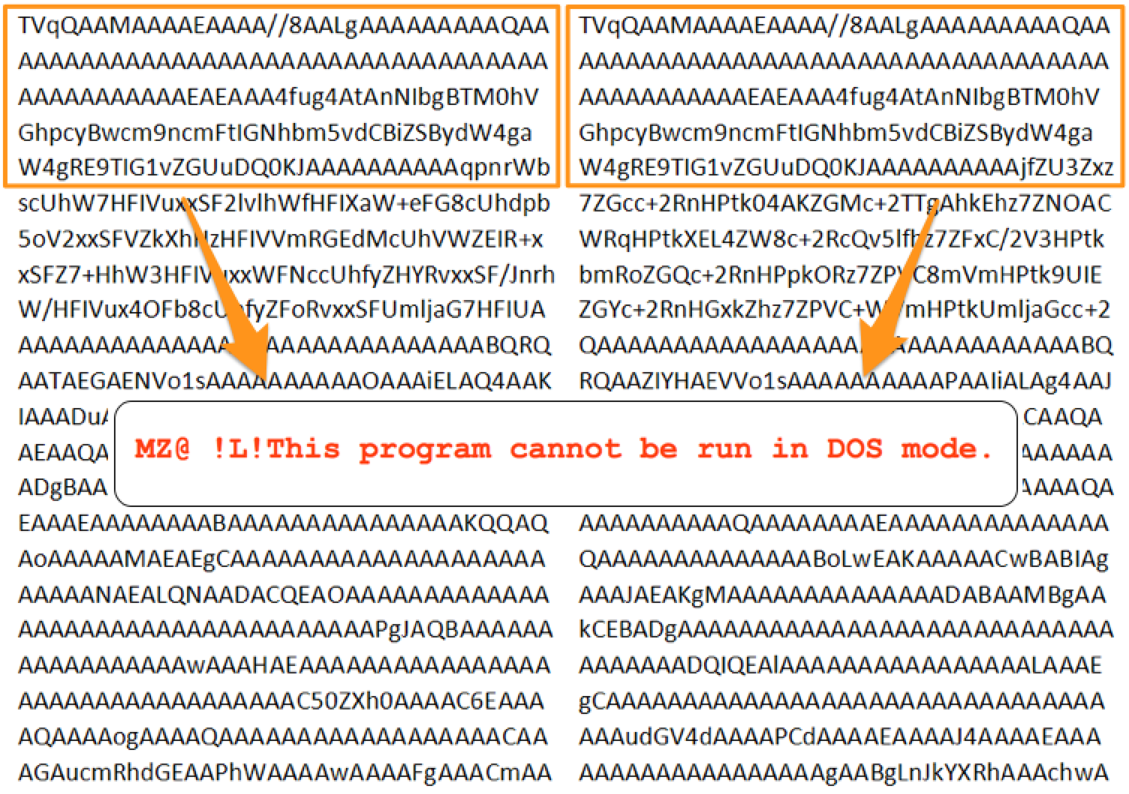
Microsoft Word 2016

Phát hiện lỗi 30914-18!

Tài liệu này đã được lưu trữ dưới phiên bản khác với phiên bản hiện tại của ứng dụng Microsoft Office. Để xem nội dung tài liệu, bạn cần phải kích hoạt tính năng tương thích với phiên bản hiện tại bằng cách nhấp vào nút "Enable Content"

Figure 1: Lure document

Once we increase the font size, the base64 blobs are visible in two different tables. Once decoded you can see the MZ header of the PE DLL at the beginning of each table, as shown in Figure 2.



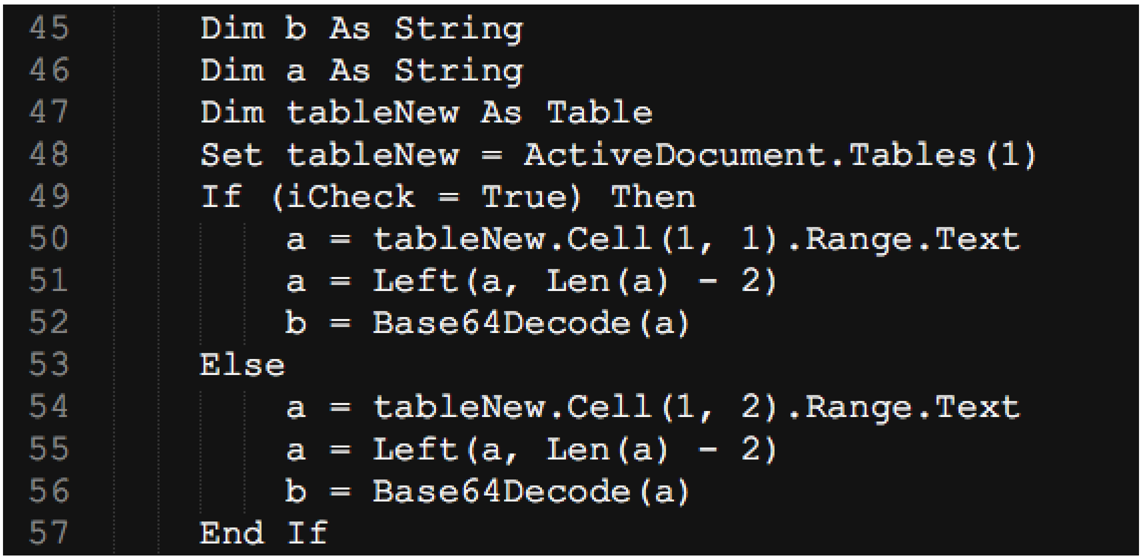
```
TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAEAEAAA4fug4AtAnNIbgBTM0hV
GhpcyBwcm9ncmFtIGNhbm5vdCBiZSBydW4ga
W4gRE9TIG1vZGUuDQ0KJAAAAAAAAAAqpnrWb
scUhW7HFIVuxxSF2lvlhWfHFIXaW+eFG8cUhdpb
5oV2xxSFVZkXhlzHFIVVmRGEdMcUhVWZEIR+x
xSFZ7+HhW3HFIVuxxWFNccUhfyZHYRvxxSF/Jnrh
W/HFIVux4OFb8cUhfyZFoRvxxSFUmljaG7HFIUA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAABQRQ
AATAEGAENVo1sAAAAAAAAAAOAAAiELAQ4AAK
IAAADu
AEAAQA
ADgBAA
EAAAEAAAAAAAABAAAAAAAAAAAAAAAKQQAQ
AoAAAAMAEAEgCAAAAAAAAAAAAAAAAAAAA
AAAAANAEALQNAADACQEAOAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAPgJAQBAAAAAA
AAAAAAAAAAwAAAHAEAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAC50ZXh0AAAAC6EAAA
AQAAAAogAAAAQAAAAAAAAAAAAAAAAAACAA
AGAucmRhdGEAAPhWAAAAwAAAAFgAAACmAA
```

```
TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAEAEAAA4fug4AtAnNIbgBTM0hV
GhpcyBwcm9ncmFtIGNhbm5vdCBiZSBydW4ga
W4gRE9TIG1vZGUuDQ0KJAAAAAAAAAAjfZU3Zxz
7ZGcc+2RnHPtk04AKZGMc+2TTgAhkEhz7ZNOAC
WRqHPtkXEL4ZW8c+2RcQv5lfhz7ZFxC/2V3HPtk
bmRoZGQc+2RnHPpkORz7ZPVC8mVmHPtk9UIE
ZGYc+2RnHGxkZhz7ZPVC+WVmHPtkUmljaGcc+2
QAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABQ
RQAAZIYHAEVVo1sAAAAAAAAAAPAAIiALAg4AAJ
CAAQA
AAAAAA
AAAAQA
AAAAAAAAAAQAAAAAAAAEAAAAAAAAAAAAAA
QAAAAAAAAAAAAAABoLwEAKAAAAACwBABIAg
AAAJAEAKgMAAAAAAAAAAAAAADABAAMBgAA
kCEBADgAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAADQIQEAlAAAAAAAAAAAAAAAALAAAE
gCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAudGV4dAAAAPCdAAAAEAAAAJ4AAAAEAAA
AAAAAAAAAAAAAAAAgAABgLnJkYXRhAAAchwA
```

MZ@ !L!This program cannot be run in DOS mode.

Figure 2: Base64 encoded pedll files embedded as text in the document.

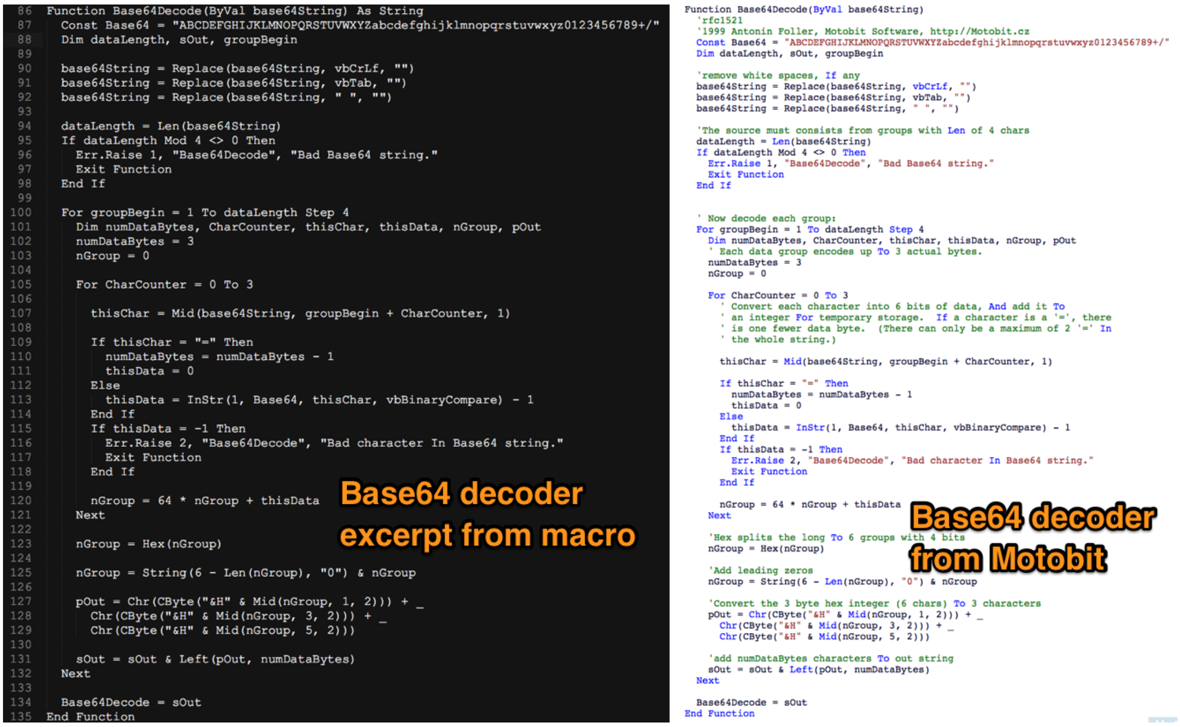
Figure 3 shows a code excerpt from the embedded macro that checks which base64 blob should be decoded based on the iCheck variable, a Boolean value which is set to true if the victim system is running on a 64-bit system and false on a 32-bit system. If the system is found to be 64-bit, the base64 encoded blob on the left is decoded otherwise the base64 encoded blob on the right is decoded.



```
Dim b As String
Dim a As String
Dim tableNew As Table
Set tableNew = ActiveDocument.Tables(1)
If (iCheck = True) Then
    a = tableNew.Cell(1, 1).Range.Text
    a = Left(a, Len(a) - 2)
    b = Base64Decode(a)
Else
    a = tableNew.Cell(1, 2).Range.Text
    a = Left(a, Len(a) - 2)
    b = Base64Decode(a)
End If
```

Figure 3: Base64 blob selection based on system check

We also noticed that the actors reused the VBS decode function published by Motobit. Figure 4 shows the comparison between the base64 function used in the macro code and the VBS base64 decoder function published by Motobit.



```
Function Base64Decode(ByVal base64String) As String
  Const Base64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
  Dim dataLength, sOut, groupBegin

  base64String = Replace(base64String, vbCrLf, "")
  base64String = Replace(base64String, vbTab, "")
  base64String = Replace(base64String, " ", "")

  dataLength = Len(base64String)
  If dataLength Mod 4 <> 0 Then
    Err.Raise 1, "Base64Decode", "Bad Base64 string."
    Exit Function
  End If

  For groupBegin = 1 To dataLength Step 4
    Dim numDataBytes, CharCounter, thisChar, thisData, nGroup, pOut
    numDataBytes = 3
    nGroup = 0

    For CharCounter = 0 To 3

      thisChar = Mid(base64String, groupBegin + CharCounter, 1)

      If thisChar = "=" Then
        numDataBytes = numDataBytes - 1
        thisData = 0
      Else
        thisData = InStr(1, Base64, thisChar, vbBinaryCompare) - 1
      End If
      If thisData = -1 Then
        Err.Raise 2, "Base64Decode", "Bad character In Base64 string."
        Exit Function
      End If

      nGroup = 64 * nGroup + thisData
    Next

    nGroup = Hex(nGroup)

    nGroup = String(6 - Len(nGroup), "0") & nGroup

    pOut = Chr(CByte("&H" & Mid(nGroup, 1, 2))) + _
      Chr(CByte("&H" & Mid(nGroup, 3, 2))) + _
      Chr(CByte("&H" & Mid(nGroup, 5, 2)))

    sOut = sOut & Left(pOut, numDataBytes)
  Next

  Base64Decode = sOut
End Function
```

Base64 decoder excerpt from macro

```
Function Base64Decode(ByVal base64String)
  'rfc1521
  '1999 Antonin Foller, Motobit Software, http://Motobit.cz
  Const Base64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
  Dim dataLength, sOut, groupBegin

  'remove white spaces, If any
  base64String = Replace(base64String, vbCrLf, "")
  base64String = Replace(base64String, vbTab, "")
  base64String = Replace(base64String, " ", "")

  'The source must consists from groups with Len of 4 chars
  dataLength = Len(base64String)
  If dataLength Mod 4 <> 0 Then
    Err.Raise 1, "Base64Decode", "Bad Base64 string."
    Exit Function
  End If


  ' Now decode each group:
  For groupBegin = 1 To dataLength Step 4
    Dim numDataBytes, CharCounter, thisChar, thisData, nGroup, pOut
    ' Each data group encodes up To 3 actual bytes.
    numDataBytes = 3
    nGroup = 0

    For CharCounter = 0 To 3
      ' Convert each character into 6 bits of data, And add it To
      ' an integer For temporary storage.  If a character is a '=', there
      ' is one fewer data byte.  (There can only be a maximum of 2 '=' In
      ' the whole string.)

      thisChar = Mid(base64String, groupBegin + CharCounter, 1)

      If thisChar = "=" Then
        numDataBytes = numDataBytes - 1
        thisData = 0
      Else
        thisData = InStr(1, Base64, thisChar, vbBinaryCompare) - 1
      End If
      If thisData = -1 Then
        Err.Raise 2, "Base64Decode", "Bad character In Base64 string."
        Exit Function
      End If

      nGroup = 64 * nGroup + thisData
    Next

    'Hex splits the long To 6 groups with 4 bits
    nGroup = Hex(nGroup)

    'Add leading zeros
    nGroup = String(6 - Len(nGroup), "0") & nGroup

    'Convert the 3 byte hex integer (6 chars) To 3 characters
    pOut = Chr(CByte("&H" & Mid(nGroup, 1, 2))) + _
      Chr(CByte("&H" & Mid(nGroup, 3, 2))) + _
      Chr(CByte("&H" & Mid(nGroup, 5, 2)))

    'add numDataBytes characters To out string
    sOut = sOut & Left(pOut, numDataBytes)
  Next

  Base64Decode = sOut
End Function
```

Base64 decoder from Motobit

Figure 4: Base64 decoder comparison

Similarity Analysis of KerrDown Samples using Jaccard-Index

Once we decoded and extracted both DLL files from the document we used a similarity analysis algorithm using the Jaccard index to check the binaries against known set of malware families used by OceanLotus which yielded no matches with any previous OceanLotus malware families. However, we were able to find multiple other samples in our datasets using the imphash value of the KerrDown samples and the accompanying C2 domains. Given the high number of samples found, we again used a similarity analysis algorithm using Jaccard Index to extract similarities between all the samples found. At this stage we were not sure if the DLL files were a backdoor or had any other functionality. Hence, we included a few other known OceanLotus malware family samples used no earlier than 2017 to our similarity test, and in most cases samples which were final payloads dropped in victim machines. One of the main objectives was to quickly discern if KerrDown could have been variants of the known malware families we have been tracking or was OceanLotus employing a new malware family in their playbooks and in the recent campaigns. Plotting the Jaccard index results using networkx we can quickly visualize the similarities extracted. As you can see from Figure 5, there is a thick cluster of samples at the top right of the networkx graph which did not have any similarities with the other known OceanLotus malware family samples. Therefore, this observation was helpful for us to understand that the samples we were looking into were likely a new malware family being employed by the OceanLotus group at the time of analysis, which we have now named KerrDown.

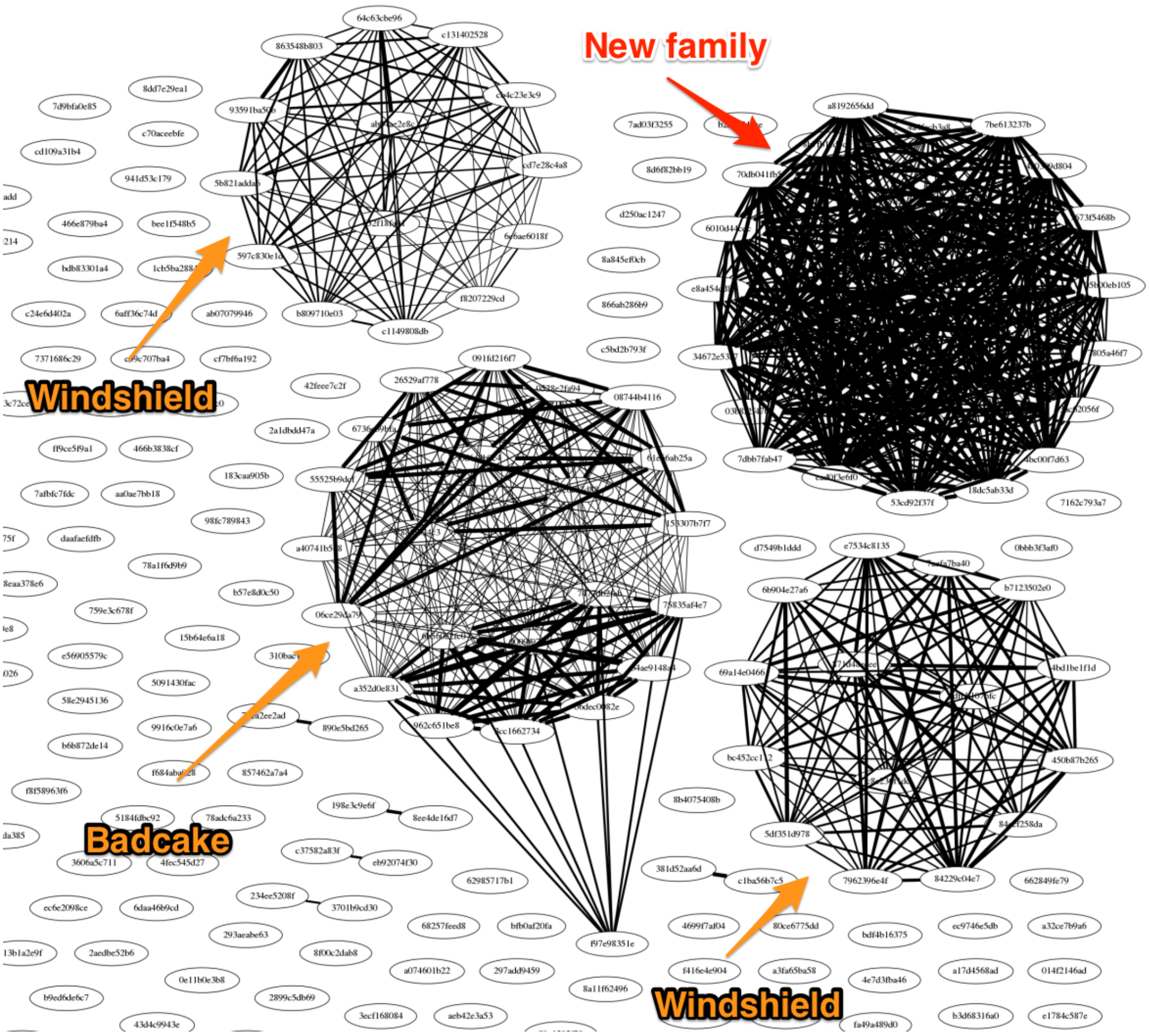


Figure 5: Similarity analysis using Jaccard Index

KerrDown to Cobalt Strike Beacon

As discussed in the delivery document analysis above, depending on the OS architecture either of the embedded KerrDown DLLs will be dropped in the victim machine. The DLL is dropped in the directory location 'Users\Administrator\AppData\Roaming\' as 'main_background.png'. The DLL retrieves the payload from the URL, decrypts it by using DES algorithm and execute it in the memory. Therefore, it is observed that only the KerrDown DLL downloader is saved in the system and the payload directly gets executed in the memory without being written in the system. Table 1 shows the URL the downloader will attempt to download the payload from depending on the OS architecture of the victim machine.

OS Architecture	URL	User Agent
32 bit	https://syn.servebbs[.]com/kuss32.gif	Mozilla/5.0 (Windows NT 10.0; Win32; x32; rv:60.0)
64 bit	https://syn.servebbs[.]com/kuss64.gif	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:60.0)

Table 1 : Payload DLL selection based on architecture

The link to the final payload of KerrDown was still active during the time of analysis and hence we were able to download a copy which turned out to be a variant of Cobalt Strike Beacon. Cybereason also published previously on OceanLotus using Cobalt Strike in their campaigns and it is interesting to see the use of a new downloader malware family being used to still deliver the final payload of Cobalt Strike. As we can see in this case, the purpose of the malware is to download and execute the Cobalt Strike Beacon payload in memory. Though Cobalt Strike is a commercial penetration testing tool, various threat actors are known to have used it in their campaigns.

RAR Archives with KerrDown

While investigating KerrDown we found multiple RAR files containing a variant of the malware. We haven't yet identified the delivery method or targets of this variant. The attacker changed the downloader code by adding more stages and hiding each stage by compression and encryption. They also changed the way to execute the malicious code from an Office macro to the DLL side-loading technique through a legitimate program.

The RAR archive

(SHA256:040abac56542a2e0f384adf37c8f95b2b6e6ce3a0ff969e3c1d572e6b4053ff3)

has the Vietnamese file name 'Don khieu nai.rar' which translates to 'Complaint letter' in English. The archive contains a legitimate older version of Microsoft Word (Microsoft Word 2007) executable file that is named 'Noi dung chi tiet don khieu nai gui cong ty.exe' which translates to 'Learn more about how to use your company' in English. The attacker used the DLL side loading technique to load a malicious DLL by the older version of Microsoft Word. When opening the executable file in the archive, it loads the malicious DLL in the same directory. The DLL executes multi-stage shellcodes and each shellcode employs various technique to hide the next stage. The overall installation steps are below:

1. The Microsoft Word exe loads wwlib.dll in the same directory and executes 'FMain' function of the DLL.
2. The DLL decodes base64 encoded shellcode in the body and executes it.
3. The shellcode decompresses the second shellcode which is compressed with the open source compression code UCL and execute it.
4. The second shellcode decrypts the third shellcode with AES.
5. The third shellcode retrieves the shellcode from the following remote location and executes it: https://cortanasyn[.]com/Avcv
6. The fourth shellcode loads the embedded Cobalt Strike Beacon DLL in memory and executes it.

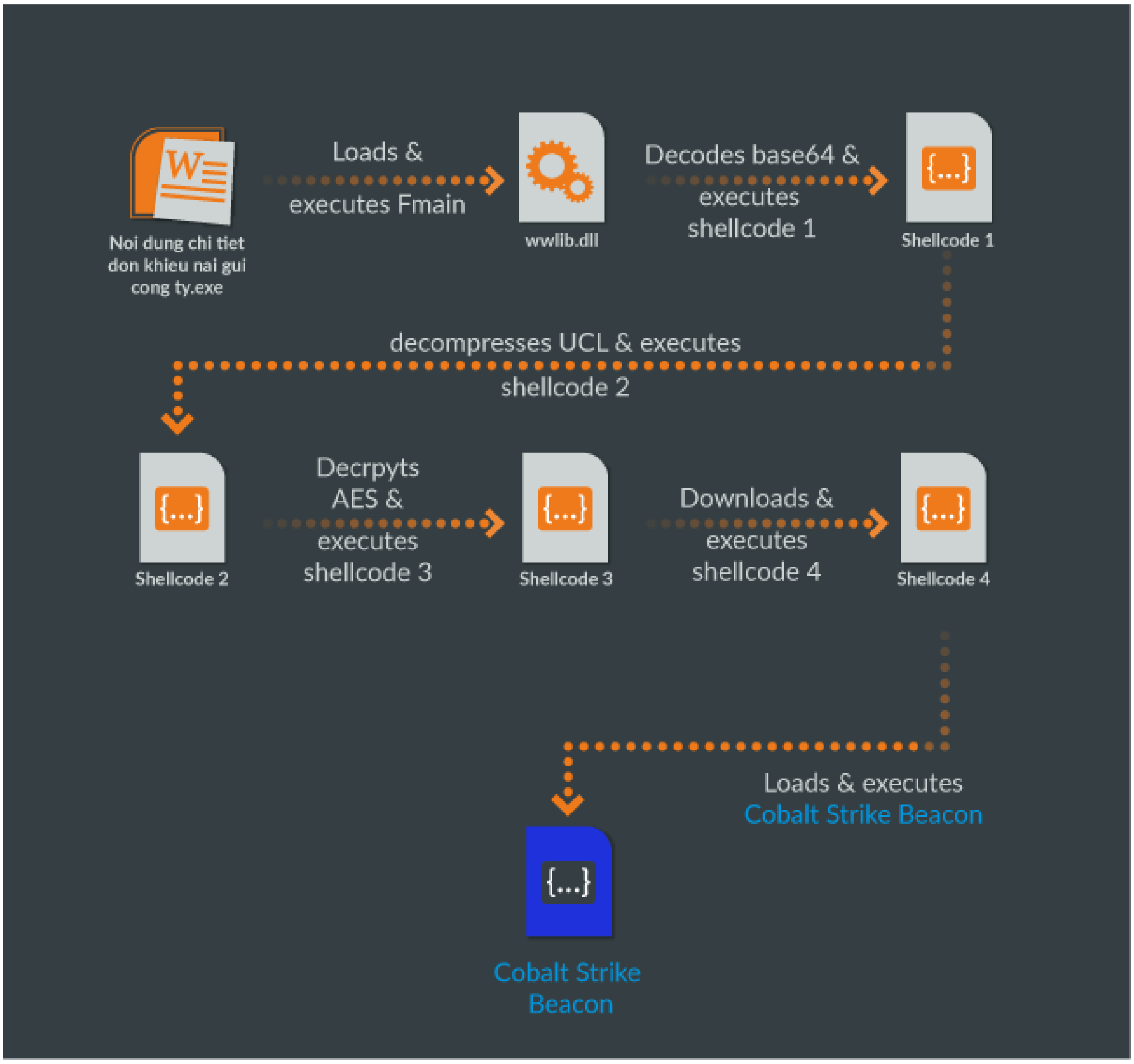

Noi dung chi tiet don khieu nai gui cong ty.exe
Loads & executes Fmain
wwlib.dll
Decodes base64 & executes shellcode 1
{...}
Shellcode 1
decompresses UCL & executes shellcode 2
Shellcode 2
Decrpyts AES & executes shellcode 3
Shellcode 3
Downloads & executes shellcode 4
Shellcode 4
Loads & executes Cobalt Strike Beacon
Cobalt Strike Beacon

Figure 6: Execution flow of sideloaded malicious downloader

Looking at the compile timestamps of all the KerrDown samples in our datasets we were able to discern a couple of observations:

> OceanLotus has been using the new downloader in their campaigns since at least March 2018 and continues to actively use it in their campaigns. Figure 7 shows the timeline of the KerrDown samples:

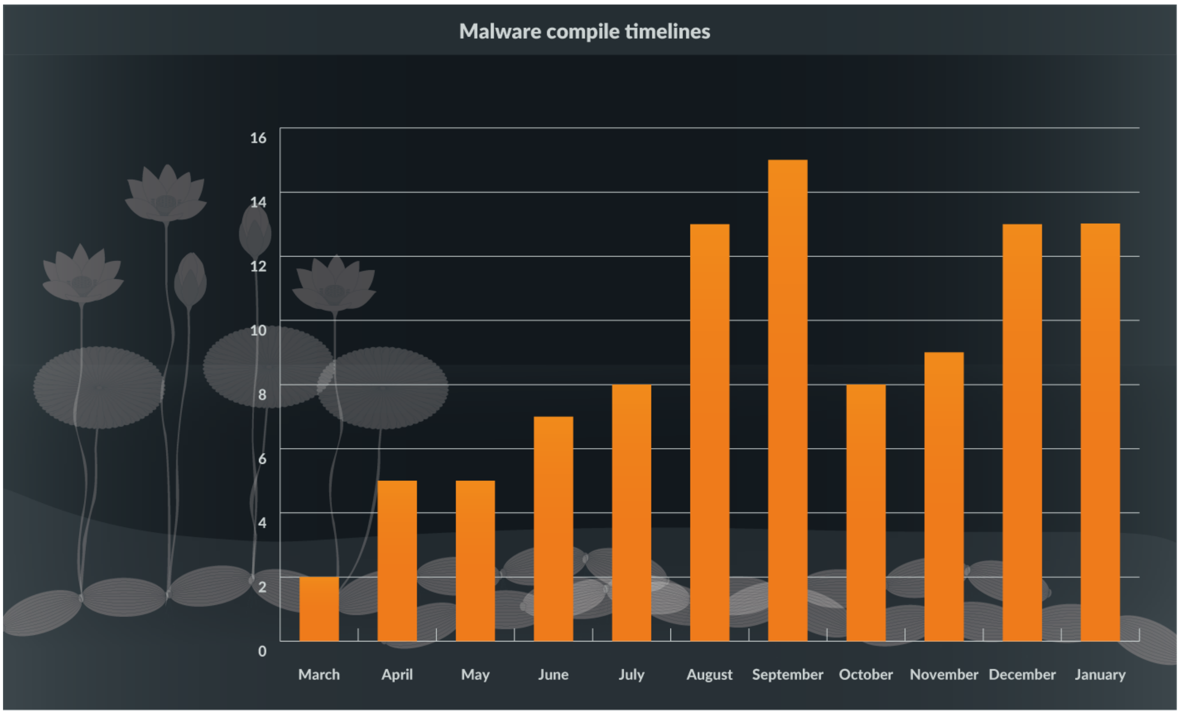


Figure 7: Downloader DLL compile time lines

While it is already widely believed that the OceanLotus group may originate from Vietnam, we wanted to find possible working hour patterns from the samples in our datasets. We plotted the compile times based on GMT +7 and found a clear pattern of the possible working hours of the group. The OceanLotus group has a typical 9 AM to 6 PM working pattern with most samples compiled during this period of the day. Figure 8 shows the malware compilation timestamps in GMT +7 for each unique sample found in our dataset.

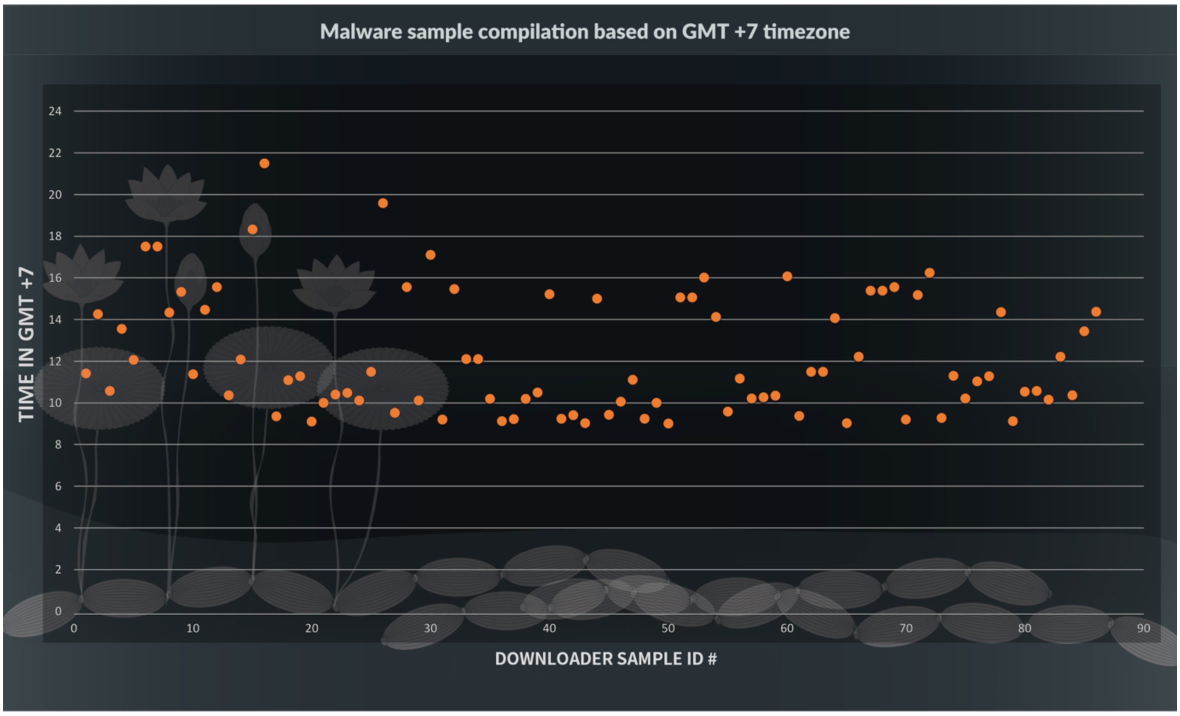


Figure 8: Malware compilation times in GMT +7

We also observed all the samples were compiled during the weekdays – between Monday to Friday. Therefore, it is clear that the OceanLotus group works during weekdays and takes a break during the weekends. Figure 9 shows the samples compiled during the week.

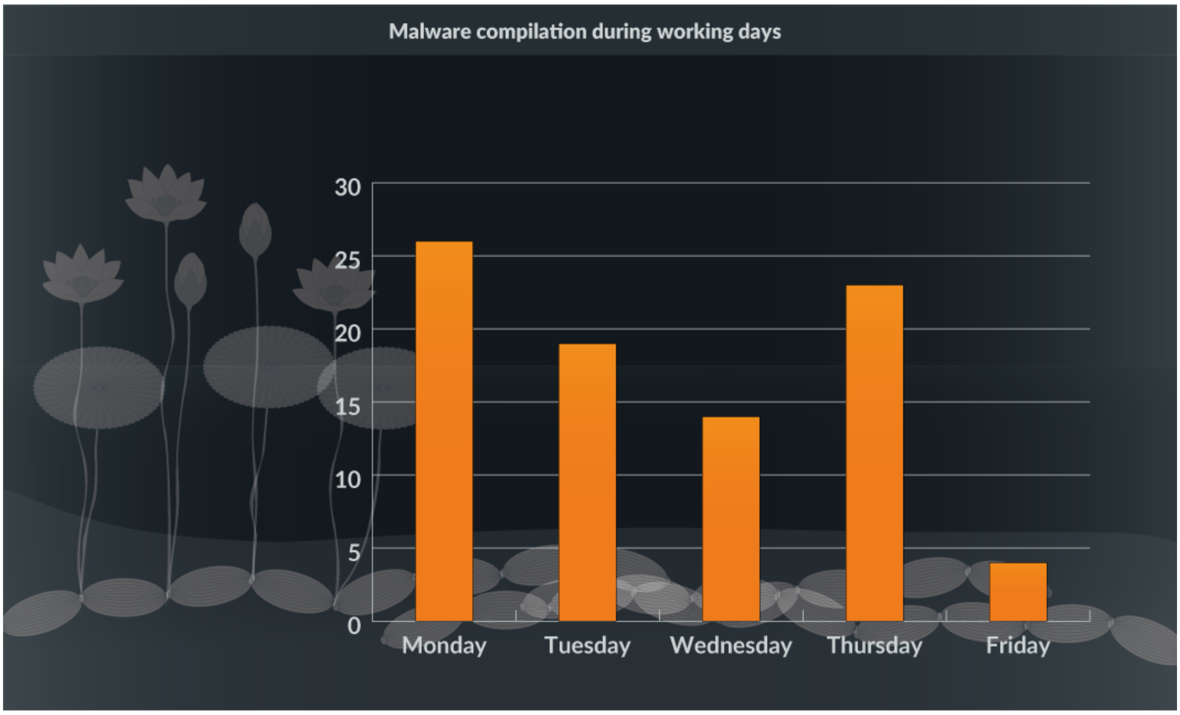


Figure 9: Malware compilation during weekdays

Conclusion

OceanLotus has been an active threat actor group for a number of years and remains one of the most sophisticated threat actors in the APAC region. As we have seen with the new KerrDown downloader being used in their recent campaigns, the group continues to build and employ new tools and techniques in their overall operations and playbooks. It is therefore imperative to understand and keep a track of the group’s ongoing operations and capability to better defend against such threats. Given the high number of samples observed, we were also able to discern possible working hour patterns which shows us that the group likely has formal working hours and operating out of a region which is like Vietnam or nearby countries. While most of the targeting observed is towards Vietnamese speaking victims, given the known broader geographic and industry wide target base of OceanLotus, the group may use similar tools and playbooks against other targets.

Palo Alto Networks customers are already protected via:

Indicators of Compromise:

Lure Docs:

73dcbcc47d6bd95dcf031ebbd34ac42301a20ee1143ac130b405e79b4ba40fc8

89e19df797481ae2d2c895bcf030fe19e581976d2aef90c89bd6b3408579bfc3

a4a066341b4172d2cb752de4b938bf678ceb627ecb72594730b78bd05a2fad9d

8bf22202e4fd4c005afde2266413cba9d1b749b1a2d75deac0c35728b5eb3af8

df8210d20c5eb80d44ba8fa4c41c26c8421dcb20168e4f796e4955e01ebc9e13

94fab926b73a6a5bc71d655c8d611b40e80464da9f1134bfce7b930e23e273ab

4321a9f95901a77b4acfbaef3596cf681712345e1cbd764873c6643fe9da7331

KerrDown DLLs:

4a0309d8043e8acd7cb5c7cfca95223afe9c15a1c34578643b49ded4b786506b

4b431af677041dae3c988fcc901ac8ec6e74c6e1467787bf099c4abd658be5be

4bc00f7d638e042da764e8648c03c0db46700599dd4f08d117e3e9e8b538519b

4e2f8f104e6cd07508c5b7d49737a1db5eeba910adfdb4c19442a7699dc78cfc

4e791f2511c9bd3c63c8e37aa6625d8b590054de9e1cca13a7be2630bc2af9ce

539e8a53db3f858914cfe0d2132f11de34a691391ba71673a8b1e61367a963c7

53cd92f37ffd0822cc644717363ba239d75c6d9af0fa305339eaf34077edd22d

53efaac9244c24fab58216a907783748d48cb32dbdc2f1f6fb672bd49f12be4c

5c18c3e6f7ac0d0ac2b5fa9a6435ee90d6bd77995f85bed9e948097891d42ca2

5cda7d8294a8804d09108359dd2d96cdf4fdcf22ec9c00f0182d005afff76743

5f0db8216314da1f128b883b918e5ac722202a2ae0c4d0bf1c5da5914a66778e

6010d44cdca58cdec4559040e08798e7b28b9434bda940da0a670c93c84e33cd

60b65ebb921dca4762aef427181775d10bbffc30617d777102762ab7913a5aa1

6146aedfe47597606fb4b05458ec4b99d4e1042da7dc974fa33a57e282cd7349

6245b74b1cc830ed95cb630192c704da66600b90a331d9e6db70210acb6c7dfa

67cd191eb2322bf8b0f04a63a9e7cb7bc52fb4a4444fcb8fed2963884aede3aa

68f77119eae5e9d2404376f2d87e71e4ab554c026e362c57313e5881005ae79e

69e679daaaff3832c39671bf2b813b5530a70fb763d381f9a6e22e3bc493c8a9

6faa7deb1e1e0c3a7c62c2bb0ecdfa56b6e3ba4fe16971ec4572267ac70b9177

6fb397e90f72783adec279434fe805c732ddb7d1d6aa72f19e91a1bf585e1ea5

70db041fb5aadb63c1b8ae57ba2699baa0086e9b011219dcebcccbf632017992

7673f5468ba3cf01500f6bb6a19ce7208c8b6fc24f1a3a388eca491bc25cd9cd

77805a46f73e118ae2428f8c22ba28f79f7c60aeb6305d41c0bf3ebb9ce70f94

788265447391189ffc1956ebfec990dc051b56f506402d43cd1d4de96709c082

7be613237b57fbc3cb83d001efadeed9936a2f519c514ab80de8285bdc5a666c

7dbb7fab4782f5e3b0c416c05114f2a51f12643805d5f3d0cd80d32272f2731a

7ec77e643d8d7cc18cc67c123feceed91d10db1cc9fa0c49164cba35bb1da987

860f165c2240f2a83eb30c412755e5a025e25961ce4633683f5bc22f6a24ddb6

868ed69533fac80354a101410d3dd0a66f444385c6611cc85c5b0be49db2d6fd

89759e56d5c23085e47d2be2ce4ad4484dfdd4204044a78671ed434cec19b693

8b7fb1cd5c09f7ec57ccc0c4261c0b4df0604962556a1d401b9cbfd750df60ba

8d6e31c95d649c08cdc2f82085298173d03c03afe02f0dacb66dd3560149184f

942d763604d0aefdff10ce095f806195f351124a8433c96f5590d89d809a562f

98a5f30699564e6d9f74e737a611246262907b9e91b90348f7de53eb4cf32665

9e6011d6380207e2bf5105cde3d48e412db565b92cdc1b3c6aa15bd7bd4b099f

a106e0a6b7cc30b161e5ea0b1ec0f28ab89c2e1eb7ba2d5d409ddbabc3b037e6

a2b905c26e2b92e63de85d83e280249258cb21f300d8c4a3a6bdb488676e9bcf

a4a86e96f95f395fcf0ceb6a74a2564f4ba7adbe1b40cc702b054427327a0399

a8192656dd1db0be4cec9d03b4d10e0529d9c52c899eda8d8e72698acfb61419

a8f776bd3a9593e963b567ce790033fec2804ea0afb40a92d40e21d8f33d066f

b4966f8febdba6b2d674afffc65b1df11e7565acbd4517f1e5b9b36a8c6a16ed

bb25f1a73d095d57b2c8c9ac6780e4d412ddf3d9eef84a54903cc8e4eaefc335

bc82bce004afb6424e9d9f9fc04a84f58edf859c4029eda08f7309dbeec67696

c30198e0b0e470d4ac8821bd14bb754466e7974f1c20be8b300961e9e89ed1ea

caabc45e59820a4349db13f337063eddede8a0847ae313d89a800f241d8556c8

d3ef6643ad529d43a7ec313b52c8396dc52c4daad688360eb207ee91a1caf7b2

e3c818052237bb4bb061290ab5e2a55c3852c8a3fef16436b1197e8b17de2e18

e56ffcf5df2afd6b151c24ddfe7cd450f9208f59b5731991b926af0dce24285a

e8704bf6525c90e0f5664f400c3bf8ff5da565080a52126e0e6a62869157dfe3

e8a454cd8b57a243f0abeec6945c9b10616cfdcc4abfb4c618bfc469d026d537

eac776c3c83c9db1a770ffaf6df9e94611c8293cbd41cb9257148603b8f2be0b

ead0f3e6f0ca16b283f09526d09e8e8cba687dab642f0e102e5487cb565bf475

f011a136996fa53fdbde944da0908da446b9532307a35c44ed08241b5e602cc9

f2a2f4fa2ed5b2a94720a4661937da97ab21aa198a5f8c83bb6895aa2c398d22

f62f21ee7e642f272b881827b45ceb643c999a742e1d3eac13d1ba014d1e7f67 f9f0973dc74716b75291f5a9b2d59b08500882563011d1def2b8d0b1b9bbb8ae

C2:

theme[[.]]blogsite[.]org

cortana[.]homelinux[.]com

word[.]webhop[.]info

work[.]windownoffice[.]com

cortanasyn[.]com

e[.]browsersyn[.]com

syn[.]servebbs[.]com

service[.]windown-update[.]com

check[.]homeip[.]net

outlook[.]updateoffices[.]net

mail[.]fptservice[.]net

office[.]windown-update[.]com

cortanazone[.]com

beta[.]officopedia[.]com

videos[.]dyndns[.]org

service[.]serveftp[.]org

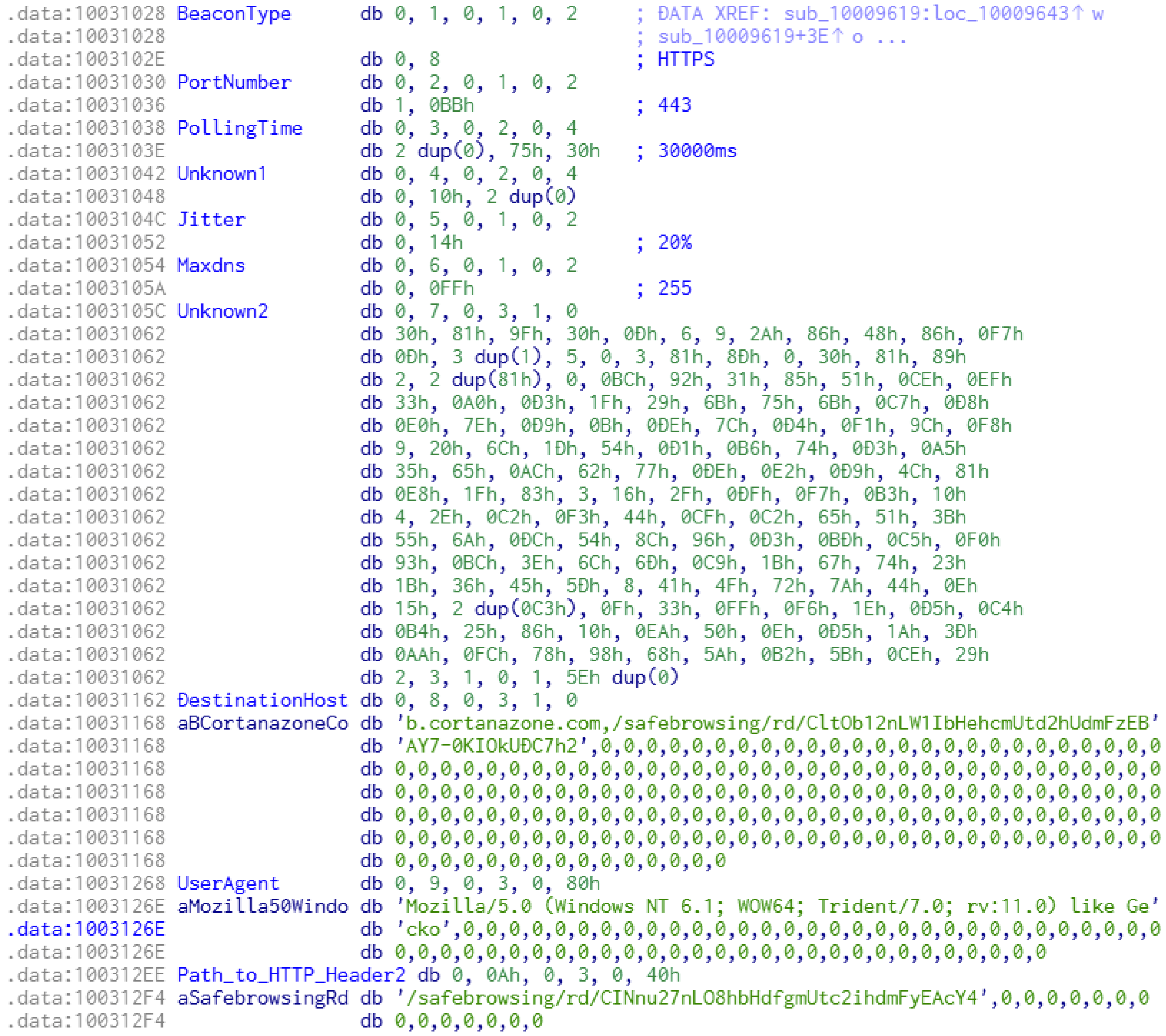
syn[.]browserstime[.]com

check[.]webhop[.]org

ristineho[.]com

Appendix A:

Cobalt Strike Beacon contains the hard-coded configuration data in its body. JPCERT published an article about the structure of the configuration. The sample we obtained has the following configuration (Figure 10) and connects to the C2 server, https:// b.cortanazone[.]com.

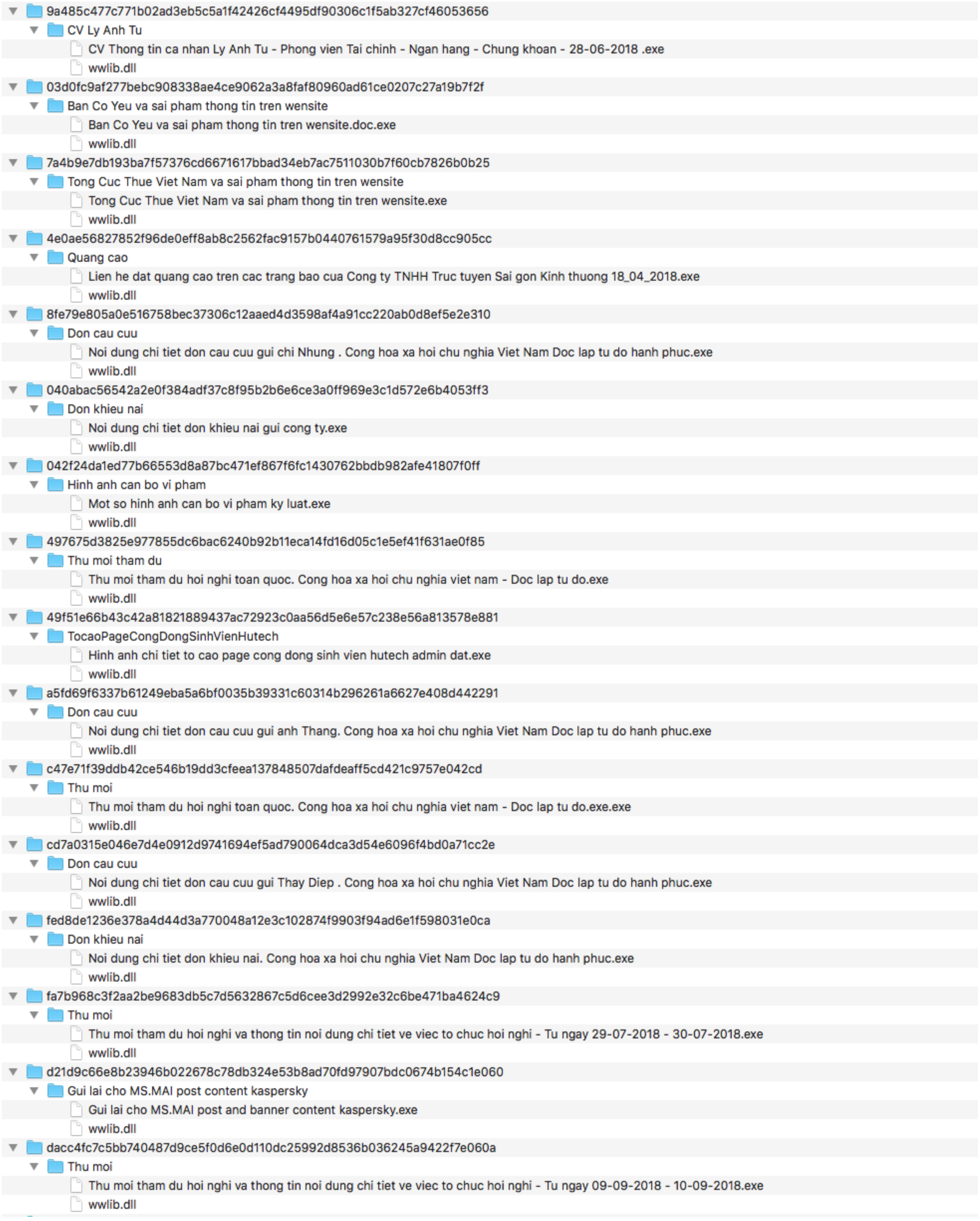


```
.data:10031028 BeaconType      db 0, 1, 0, 1, 0, 2      ; DATA XREF: sub_10009619:loc_10009643↑ w
.data:10031028                                          ; sub_10009619+3E↑ o ...
.data:1003102E                 db 0, 8                  ; HTTPS
.data:10031030 PortNumber      db 0, 2, 0, 1, 0, 2
.data:10031036                 db 1, 0BBh               ; 443
.data:10031038 PollingTime     db 0, 3, 0, 2, 0, 4
.data:1003103E                 db 2 dup(0), 75h, 30h    ; 30000ms
.data:10031042 Unknown1        db 0, 4, 0, 2, 0, 4
.data:10031048                 db 0, 10h, 2 dup(0)
.data:1003104C Jitter          db 0, 5, 0, 1, 0, 2
.data:10031052                 db 0, 14h                ; 20%
.data:10031054 Maxdns          db 0, 6, 0, 1, 0, 2
.data:1003105A                 db 0, 0FFh               ; 255
.data:1003105C Unknown2        db 0, 7, 0, 3, 1, 0
.data:10031062                 db 30h, 81h, 9Fh, 30h, 0Dh, 6, 9, 2Ah, 86h, 48h, 86h, 0F7h
.data:10031062                 db 0Dh, 3 dup(1), 5, 0, 3, 81h, 8Dh, 0, 30h, 81h, 89h
.data:10031062                 db 2, 2 dup(81h), 0, 0BCh, 92h, 31h, 85h, 51h, 0CEh, 0EFh
.data:10031062                 db 33h, 0A0h, 0D3h, 1Fh, 29h, 6Bh, 75h, 6Bh, 0C7h, 0D8h
.data:10031062                 db 0E0h, 7Eh, 0D9h, 0Bh, 0DEh, 7Ch, 0D4h, 0F1h, 9Ch, 0F8h
.data:10031062                 db 9, 20h, 6Ch, 1Dh, 54h, 0D1h, 0B6h, 74h, 0D3h, 0A5h
.data:10031062                 db 35h, 65h, 0ACh, 62h, 77h, 0DEh, 0E2h, 0D9h, 4Ch, 81h
.data:10031062                 db 0E8h, 1Fh, 83h, 3, 16h, 2Fh, 0DFh, 0F7h, 0B3h, 10h
.data:10031062                 db 4, 2Eh, 0C2h, 0F3h, 44h, 0CFh, 0C2h, 65h, 51h, 3Bh
.data:10031062                 db 55h, 6Ah, 0DCh, 54h, 8Ch, 96h, 0D3h, 0BDh, 0C5h, 0F0h
.data:10031062                 db 93h, 0BCh, 3Eh, 6Ch, 6Dh, 0C9h, 1Bh, 67h, 74h, 23h
.data:10031062                 db 1Bh, 36h, 45h, 5Dh, 8, 41h, 4Fh, 72h, 7Ah, 44h, 0Eh
.data:10031062                 db 15h, 2 dup(0C3h), 0Fh, 33h, 0FFh, 0F6h, 1Eh, 0D5h, 0C4h
.data:10031062                 db 0B4h, 25h, 86h, 10h, 0EAh, 50h, 0Eh, 0D5h, 1Ah, 3Dh
.data:10031062                 db 0AAh, 0FCh, 78h, 98h, 68h, 5Ah, 0B2h, 5Bh, 0CEh, 29h
.data:10031062                 db 2, 3, 1, 0, 1, 5Eh dup(0)
.data:10031162 DestinationHost db 0, 8, 0, 3, 1, 0
.data:10031168 aBCortanazoneCo db 'b.cortanazone.com,/safebrowsing/rd/CltOb12nLW1IbHehcmUtd2hUdmFzEB'
.data:10031168                 db 'AY7-0KIOkUDC7h2',0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
.data:10031168                 db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
.data:10031168                 db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
.data:10031168                 db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
.data:10031168                 db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
.data:10031168                 db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
.data:10031268 UserAgent       db 0, 9, 0, 3, 0, 80h
.data:1003126E aMozilla50Windo db 'Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Ge'
.data:1003126E                 db 'cko',0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
.data:1003126E                 db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
.data:100312EE Path_to_HTTP_Header2 db 0, 0Ah, 0, 3, 0, 40h
.data:100312F4 aSafebrowsingRd db '/safebrowsing/rd/CINnu27nLO8hbHdfgmUtc2ihdmFyEAcY4',0,0,0,0,0,0,0
.data:100312F4                 db 0,0,0,0,0,0,0
```

Figure 10: Cobalt Strike Beacon configuration

Appendix B:

Figure 11 shows some of the contents of the individual RAR files. All the .exe files are copies of Windows Word and the associated 'wwlib.dll' file is the malicious downloader DLL KerrDown, which is sideloaded when the .exe file gets executed.



```
▼ 9a485c477c771b02ad3eb5c5a1f42426cf4495df90306c1f5ab327cf46053656
    ▼ CV Ly Anh Tu
        CV Thong tin ca nhan Ly Anh Tu - Phong vien Tai chinh - Ngan hang - Chung khoan - 28-06-2018 .exe
        wwlib.dll
▼ 03d0fc9af277bebc908338ae4ce9062a3a8faf80960ad61ce0207c27a19b7f2f
    ▼ Ban Co Yeu va sai pham thong tin tren wensite
        Ban Co Yeu va sai pham thong tin tren wensite.doc.exe
        wwlib.dll
▼ 7a4b9e7db193ba7f57376cd6671617bbad34eb7ac7511030b7f60cb7826b0b25
    ▼ Tong Cuc Thue Viet Nam va sai pham thong tin tren wensite
        Tong Cuc Thue Viet Nam va sai pham thong tin tren wensite.exe
        wwlib.dll
▼ 4e0ae56827852f96de0eff8ab8c2562fac9157b0440761579a95f30d8cc905cc
    ▼ Quang cao
        Lien he dat quang cao tren cac trang bao cua Cong ty TNHH Truc tuyen Sai gon Kinh thuong 18_04_2018.exe
        wwlib.dll
▼ 8fe79e805a0e516758bec37306c12aaed4d3598af4a91cc220ab0d8ef5e2e310
    ▼ Don cau cuu
        Noi dung chi tiet don cau cuu gui chi Nhung . Cong hoa xa hoi chu nghia Viet Nam Doc lap tu do hanh phuc.exe
        wwlib.dll
▼ 040abac56542a2e0f384adf37c8f95b2b6e6ce3a0ff969e3c1d572e6b4053ff3
    ▼ Don khieu nai
        Noi dung chi tiet don khieu nai gui cong ty.exe
        wwlib.dll
▼ 042f24da1ed77b66553d8a87bc471ef867f6fc1430762bbdb982afe41807f0ff
    ▼ Hinh anh can bo vi pham
        Mot so hinh anh can bo vi pham ky luat.exe
        wwlib.dll
▼ 497675d3825e977855dc6bac6240b92b11eca14fd16d05c1e5ef41f631ae0f85
    ▼ Thu moi tham du
        Thu moi tham du hoi nghi toan quoc. Cong hoa xa hoi chu nghia viet nam - Doc lap tu do.exe
        wwlib.dll
▼ 49f51e66b43c42a81821889437ac72923c0aa56d5e6e57c238e56a813578e881
    ▼ TocaoPageCongDongSinhVienHutech
        Hinh anh chi tiet to cao page cong dong sinh vien hutech admin dat.exe
        wwlib.dll
▼ a5fd69f6337b61249eba5a6bf0035b39331c60314b296261a6627e408d442291
    ▼ Don cau cuu
        Noi dung chi tiet don cau cuu gui anh Thang. Cong hoa xa hoi chu nghia Viet Nam Doc lap tu do hanh phuc.exe
        wwlib.dll
▼ c47e71f39ddb42ce546b19dd3cfeea137848507dafdeaff5cd421c9757e042cd
    ▼ Thu moi
        Thu moi tham du hoi nghi toan quoc. Cong hoa xa hoi chu nghia viet nam - Doc lap tu do.exe.exe
        wwlib.dll
▼ cd7a0315e046e7d4e0912d9741694ef5ad790064dca3d54e6096f4bd0a71cc2e
    ▼ Don cau cuu
        Noi dung chi tiet don cau cuu gui Thay Diep . Cong hoa xa hoi chu nghia Viet Nam Doc lap tu do hanh phuc.exe
        wwlib.dll
▼ fed8de1236e378a4d44d3a770048a12e3c102874f9903f94ad6e1f598031e0ca
    ▼ Don khieu nai
        Noi dung chi tiet don khieu nai. Cong hoa xa hoi chu nghia Viet Nam Doc lap tu do hanh phuc.exe
        wwlib.dll
▼ fa7b968c3f2aa2be9683db5c7d5632867c5d6cee3d2992e32c6be471ba4624c9
    ▼ Thu moi
        Thu moi tham du hoi nghi va thong tin noi dung chi tiet ve viec to chuc hoi nghi - Tu ngay 29-07-2018 - 30-07-2018.exe
        wwlib.dll
▼ d21d9c66e8b23946b022678c78db324e53b8ad70fd97907bdc0674b154c1e060
    ▼ Gui lai cho MS.MAI post content kaspersky
        Gui lai cho MS.MAI post and banner content kaspersky.exe
        wwlib.dll
▼ dacc4fc7c5bb740487d9ce5f0d6e0d110dc25992d8536b036245a9422f7e060a
    ▼ Thu moi
        Thu moi tham du hoi nghi va thong tin noi dung chi tiet ve viec to chuc hoi nghi - Tu ngay 09-09-2018 - 10-09-2018.exe
        wwlib.dll
```

Figure 11: RAR archives with malicious DLL for sideloading