Spoofed Invoice Used to Drop IcedID | FortiGuard Labs

By James Slaughter

Published: 2022-03-28 · Archived: 2026-04-05 16:08:26 UTC

Spearphishing crafted with industry-specific terms derived from intelligence gathering techniques to trick a recipient into opening a file is especially difficult to identify. This is especially true when an adversary has knowledge of how a business works and the processes that underpin it. Using this knowledge, a lure can be crafted that takes advantage of these day-to-day processes – for example, settling the cost of a fuel transaction.

FortiGuard Labs recently encountered such a scenario, where a fuel company in Kyiv, Ukraine received a spearphishing e-mail that contained an attached invoice—seemingly from another fuel provider—that was spoofed. The attachment is a zip file that contains the IcedID Trojan.

IcedID has been observed as far back as 2017. Its primary function is to steal banking credentials and personal information. It is also capable of deploying additional malware from the same group or partner organizations.

This instance also uses an interesting deployment method. It uses the ISO format, which is mounted automatically as a disk in Windows. ISO files can also be used to create bootable CD-ROMs or install an operating system or virtual machine. It also contains a LNK (shortcut file) used to launch a DLL (Dynamic-link Library).

This blog details the infection process and subsequent malware deployment by the threat actors behind IcedID.

Affected Platforms: Windows
Impacted Users: Windows users
Impact: Compromised machines are under the control of the threat actor
Severity Level: Medium

The Phishing E-mail

The e-mail originated from an IP address in Belize, at 179[.]60[.]150[.]96. It spoofs the originating e-mail address to appear to have been sent from another fuel provider in Ukraine. The e-mail contains both English and Ukrainian elements and looks realistic given the mention of extra security measures regarding the attachment.

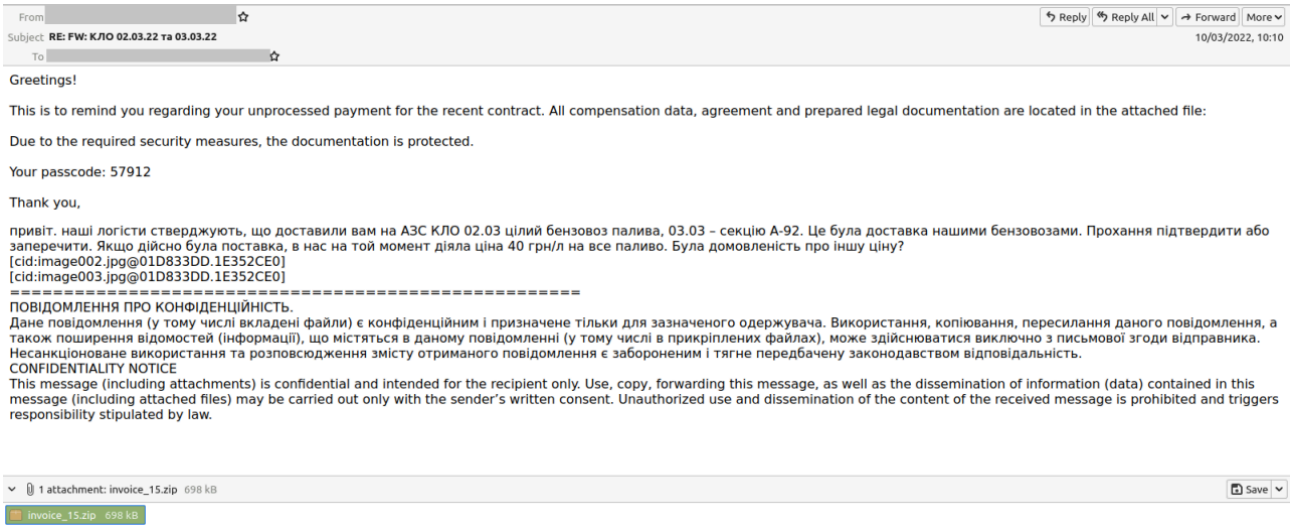

Figure 1. Phishing e-mail.

Attached to the e-mail is a file named “invoice_15.zip”. Extracting the Zip file will drop “invoice_15.iso” and begin the first phase of infection.

ISO

Windows is capable of mounting iso files as external disks. Doing so will present the user with a shortcut called “document.” In most cases, the file extension will be hidden from the user, making it appear as an actual document.

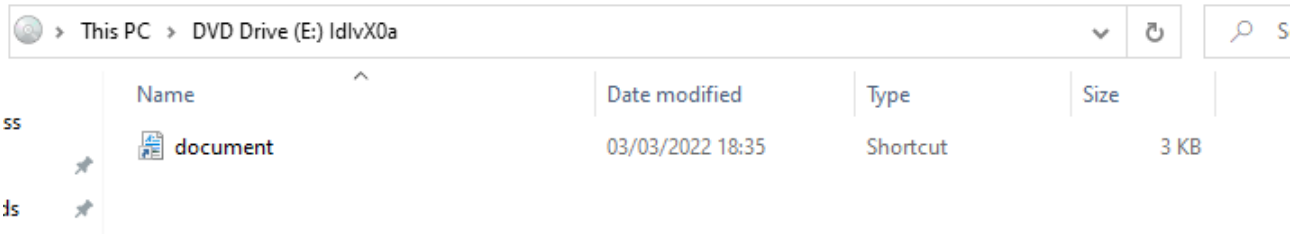

Figure 2. ISO file with contents hidden.

When the full contents of the iso container are revealed, a DLL file can also be seen.

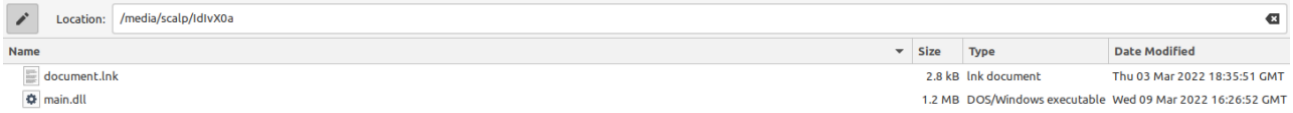

Figure 3. Full contents of the ISO file.

LNK

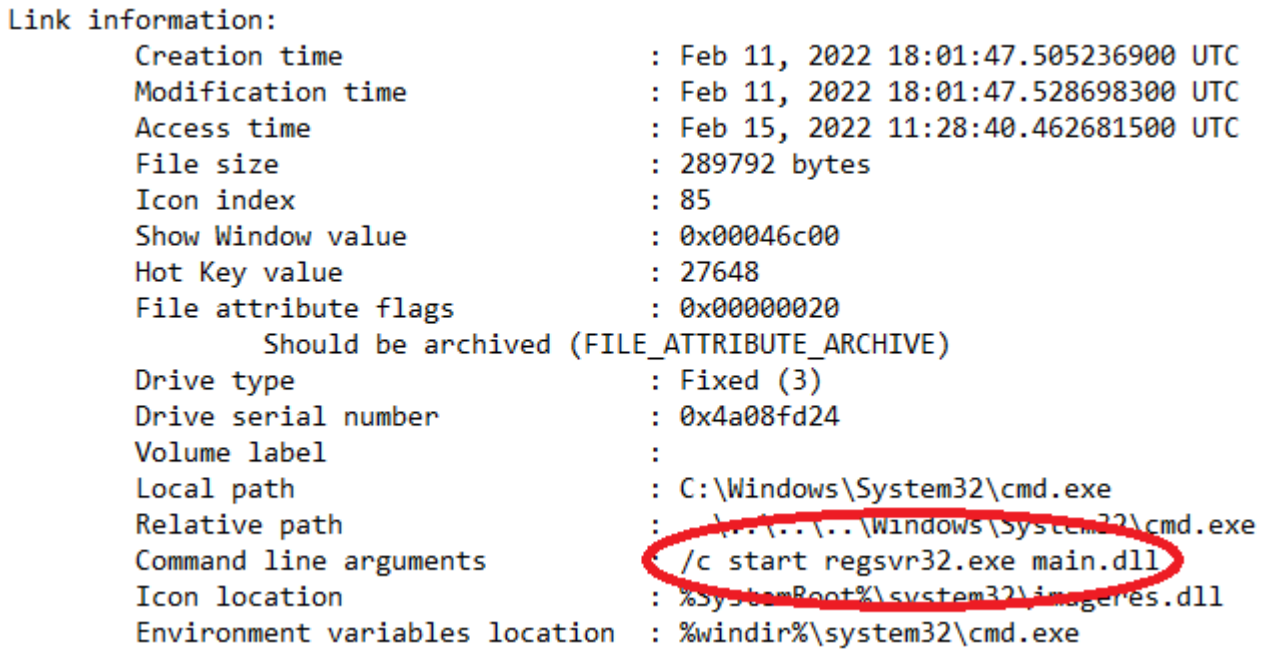

Figure 4. Shortcut details.

As seen in Figure 4, the shortcut file was created some time prior to the sending of the phishing e-mail. Additionally, the highlighted area shows what will occur should the shortcut be clicked on by a user.

In this case, Regsvr32 is used to register “main.dll” with the Windows registry and launch the code contained within. This action begins the next phase of infection.

Dropper

“main.dll” acts as a dropper for IcedID. Static analysis of the file reveals an interesting point.

```
210.125.167.240
GET /skin/base/board/view_brit2.gif HTTP/1.0
Host: cwc.ghc.ac.kr
connect() failed
send() sent a different number of bytes than expected
C:\a\example.gif
```

Figure 5. Example of strings embedded in “main.dll”

What at appears at first glance to be an easy win for IOCs (Indicators of Compromise) because it contains a domain and IP address, turns out to be slightly more complicated.

```
.rdata:0000000180112B18 aTypelib_1:                             ; DATA XREF: .rdata:00000001801131D8↓o
.rdata:0000000180112B18                 text "UTF-16LE", 'TypeLib',0
.rdata:0000000180112B28 a210125167240   db '210.125.167.240',0
.rdata:0000000180112B38 aGetSkinBaseBoa db 'GET /skin/base/board/view_brit2.gif HTTP/1.0',0Dh,0Ah,0
.rdata:0000000180112B67                 align 8
.rdata:0000000180112B68 aHostCwcGhcAcKr db 'Host: cwc.ghc.ac.kr',0Dh,0Ah,0
.rdata:0000000180112B7E                 align 20h
.rdata:0000000180112B80 aConnectFailed  db 'connect() failed',0
.rdata:0000000180112B91                 align 8
.rdata:0000000180112B98 aSendSentADiffe db 'send() sent a different number of bytes than expected',0
.rdata:0000000180112BCE                 align 10h
.rdata:0000000180112BD0 aCAExampleGif   db 'C:\a\example.gif',0
.rdata:0000000180112BE1                 align 8
```

Figure 6. Code represented in IDA Pro showing the information from Figure 5.

In comparing the area of code where the strings in Figure 5 are stored, we find that this area is not called by any functions within “main.dll”. To illustrate this, the right-hand side of the very first line in Figure 6 contains “Data XREF:”. This indicates that it is referenced elsewhere in the code. The strings from Figure 5, however, do not include this information, indicating they are not.

By investigating further, the story becomes even more interesting. This code appears in a StackOverflow question from approximately 10 years ago concerning an issue about downloading an image over HTTP (https://stackoverflow.com/questions/9389183/downloading-a-picture-with-http-get-only-downloads-a-small-part-of-it). It should be noted that there is no malicious intent with the content of that posting.

That it is now part of “main.dll” indicates it is a decoy for analysts in the hope the actual indicators won’t be blocked.

Process Name	PID	Operation	Path	Result	Detail
regsvr32.exe	5668	Process Create	C:\Windows\system32\ipconfig.exe	SUCCESS	PID: 756, Command line: ipconfig /all
regsvr32.exe	5668	Process Create	C:\Windows\system32\systeminfo.exe	SUCCESS	PID: 3312, Command line: systeminfo
regsvr32.exe	5668	Process Create	C:\Users\jim\AppData\Local\Temp\Arur.exe	SUCCESS	PID: 4728, Command line: "C:\Users\jim\AppData\Local\Temp\Arur.exe"
regsvr32.exe	5668	Process Create	C:\Windows\system32\net.exe	SUCCESS	PID: 1520, Command line: net config workstation
net.exe	1520	Process Create	C:\Windows\system32\net1.exe	SUCCESS	PID: 8368, Command line: C:\Windows\system32\net1 config workstation
regsvr32.exe	5668	Process Create	C:\Windows\system32\nltest.exe	SUCCESS	PID: 6224, Command line: nltest /domain_trusts
regsvr32.exe	5668	Process Create	C:\Windows\system32\nltest.exe	SUCCESS	PID: 8940, Command line: nltest /domain_trusts /all_trusts
regsvr32.exe	5668	Process Create	C:\Windows\system32\net.exe	SUCCESS	PID: 1144, Command line: net view /all /domain
regsvr32.exe	5668	Process Create	C:\Windows\system32\net.exe	SUCCESS	PID: 8852, Command line: net view /all
regsvr32.exe	5668	Process Create	C:\Windows\system32\net.exe	SUCCESS	PID: 5940, Command line: net group "Domain Admins" /domain
net.exe	5940	Process Create	C:\Windows\system32\net1.exe	SUCCESS	PID: 2844, Command line: C:\Windows\system32\net1 group "Domain Admins" /domain

Figure 7. Information gathering by IcedID.

As can be seen in Figure 7, once running, the malware uses several Windows command-line tools to obtain information about the local environment. These include capturing the local IP address (ipconfig), enumerating domain trusts (nltest), and capturing a list of domain administrators (net group), among others.

The sample then tries to communicate outbound to a command and control (C2) server. There are multiple addresses the malware can connect to in the event one of the destinations becomes unavailable.

Process Name	PID	Operation	Path	Result
regsvr32.exe	5668	TCP Connect	DESKTOP-LETOT1K.mshome.net:50777 -> ip-160-153-32-99.ip.secureserver.net:https	SUCCESS
regsvr32.exe	5668	TCP Send	DESKTOP-LETOT1K.mshome.net:50777 -> ip-160-153-32-99.ip.secureserver.net:https	SUCCESS
regsvr32.exe	5668	TCP Receive	DESKTOP-LETOT1K.mshome.net:50777 -> ip-160-153-32-99.ip.secureserver.net:https	SUCCESS

Figure 8. Network communication.

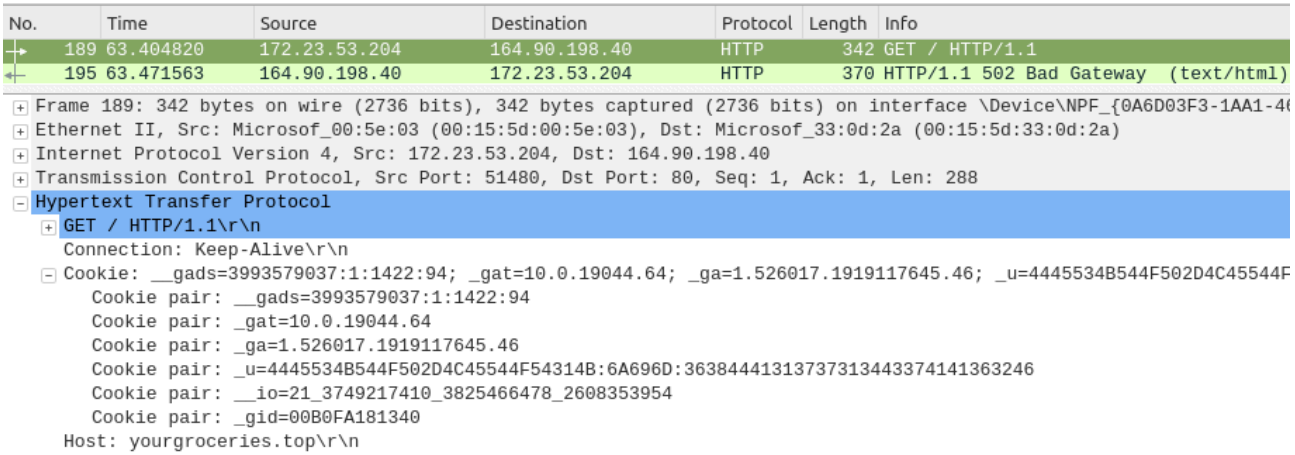

Figure 9. HTTP GET request.

If a connection to a C2 server has been made, the malware then moves to ensure persistence. It installs a copy of itself in the user's temp directory, "%APPDATA%\local\temp".

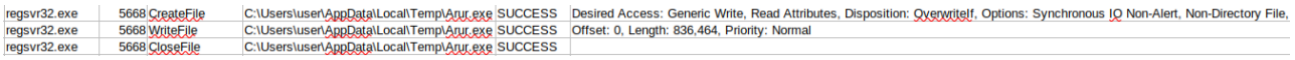

Figure 10. Dropping "Arur.exe" into the Temp directory.

Conclusion

Threat actors that are knowledgeable of their targets are able to increase their chances of installing an implant within an organization. Based on our observations, the efforts used in this IcedID attack highlight the groups methodical effort, as evidenced by their research of Ukraine's retail fuel industry. Additionally, the use of uncommon deployment methods (zipped ISO file) to establish a foothold—and ultimately gain persistence within an organization—reveals how crafty the threat actors are able to be to obtain unauthorized access.

Fortinet Protections

All IcedID samples mentioned in this blog are detected by the following (AV) signatures:

W32/Kryptik.HOTN!tr
W64/Kryptik.CXY!tr
W64/Kryptik.CXY!tr
W64/Kryptik.CXY!tr
LNK/IceID.AW!tr
W64/Kryptik.CXY!tr

All network based URI's are blocked by the WebFiltering client.

Fortinet has multiple solutions designed to help train users to understand and detect phishing threats:

The FortiPhish Phishing Simulation Service uses real-world simulations to help organizations test user awareness and vigilance to phishing threats and to train and reinforce proper practices when users encounter targeted

phishing attacks.

In addition to these protections, we suggest that organizations also have their end users go through our FREE NSE training: NSE 1 – Information Security Awareness. It includes a module on Internet threats that is designed to help end users learn how to identify and protect themselves from various types of phishing attacks.

IOCs

Filename	SHA256
invoice_15.zip	83bd20009107e1f60479016046b80d473436d3883ad6989e5d42bc08e142b5bb
invoice_15.iso	3542d5179100a7644e0a747139d775dbc8d914245292209bc9038ad2413b3213
document.lnk	a17e32b43f96c8db69c979865a8732f3784c7c42714197091866473bcfac8250
main.dll	698a0348c4bb8fffc806a1f915592b20193229568647807e88a39d2ab81cb4c2
Arur.exe	283d5eea1f9fc34e351deacc25006fc1997566932fae44db4597c84f1f1f3a30

Network IOCs:

160[.]153[.]32[.]99
160[.]90[.]198[.]40
yourgroceries[.]top
ssddds1ssd2[.]com
ip-160-153-32-99[.]ip[.]secureserver[.]net

Thanks to Val Saengphaibul and Fred Gutierrez who helped contribute to this blog.

Learn more about Fortinet's FortiGuard Labs threat research and intelligence organization and the FortiGuard Security Subscriptions and Services portfolio.
