SANS ISC: InfoSec Handlers Diary Blog - SANS Internet Storm Center SANS Site Network Current Site SANS Internet Storm Center Other SANS Sites Help Graduate Degree Programs Security Training Security Certification Security Awareness Training Penetration Testing Industrial Control Systems Cyber Defense Foundations DFIR Software Security Government OnSite Training InfoSec Handlers Diary Blog

Agent Tesla hidden in a historical anti-malware tool

Published: 2021-02-11
Last Updated: 2021-02-11 07:17:18 UTC
by Jan Kopriva (Version: 1)
1 comment(s)

While going through attachments of e-mails, which were caught in my e-mail quarantine since the beginning of February, I found an ISO file with what turned out to be a sample of the Agent Tesla infostealer. That, by itself, would not be that unusual, but the Agent Tesla sample turned out to be unconventional in more ways than one...

The e-mail carrying the ISO attachment was a run-of-the-mill-looking malspam, informing the recipient about a new delivery from DHL. It had a spoofed sender address “dhlSender@dhl.com”, which – although looking at least somewhat believable – certainly didn’t have the impact of making the message appear trustworthy, which is what the authors of the e-mail were most likely hoping for. On the contrary, it must have resulted in very few of the messages actually making it past any security analysis on e-mail gateways. The reason is that DHL has a valid SPF record set up for dhl.com, so any SPF check (i.e. something that most of the worlds e-mail servers perform automatically these days) would lead to a “soft fail” result, which would consequently most likely lead to the message being quarantined (if not deleted outright).

dhlSender@dhl.com | 0

DHL EXPRESS SERVICE [YOU HAVE A PACKAGE READY FOR PICKUP]

Download_Tracking_Re...
0 bytes

FYI

DHL EXPRESS

Your Dedicated
International Specialist

Dear valued customer

We are pleased to inform you that your consignment was booked via DHL Express

Copies of the shipment has been attached

To be able to check the status of the above shipment, simply check the enclosed file to track your shipment and more detailed information of the consignment is available

Download your attachment for your reference

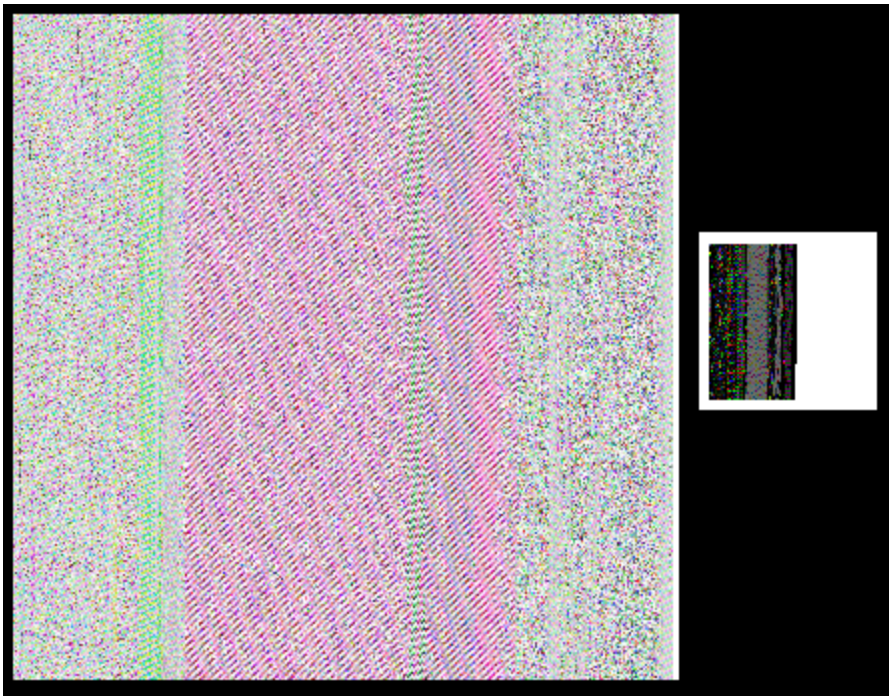
Wishing you and your company a fruitful business!

Best Regards,
DHL International Express Ltd.

Keep the downloaded documents safe because, we will need you to provide them for confirmation before delivering your parcel.

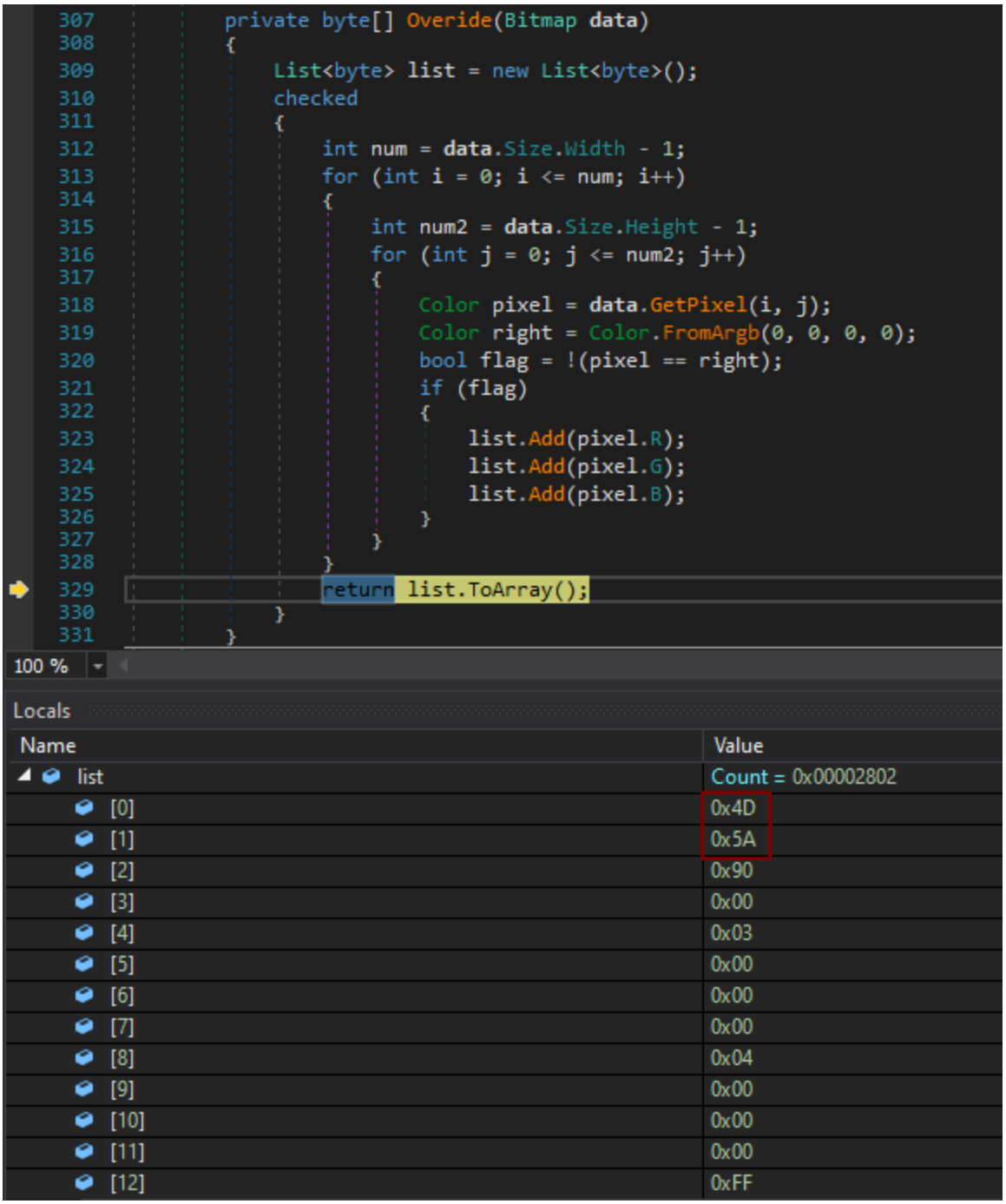
The attached file *Download_Tracking_Reference.01.02.2021.xlsx.iso* contained only one EXE with identical name (except for the second extension, of course).

The executable was written in VB.NET and its malicious payload was hidden in it in an interesting way – the file had two bitmaps embedded in its Resources section, both of which were in fact encoded/encrypted DLLs.



While the use of bitmaps for embedding DLLs is not new for Agent Tesla[1], it is certainly an interesting way to hide malicious code and prevent its detection. In this case, it didn't seem to help the file too much, given its 41/71 VT score at the time of writing[2], but it is quite imaginative technique nonetheless.

After the file was executed, it would first decode and load a small (10kB) DLL named *BestFit.dll*.



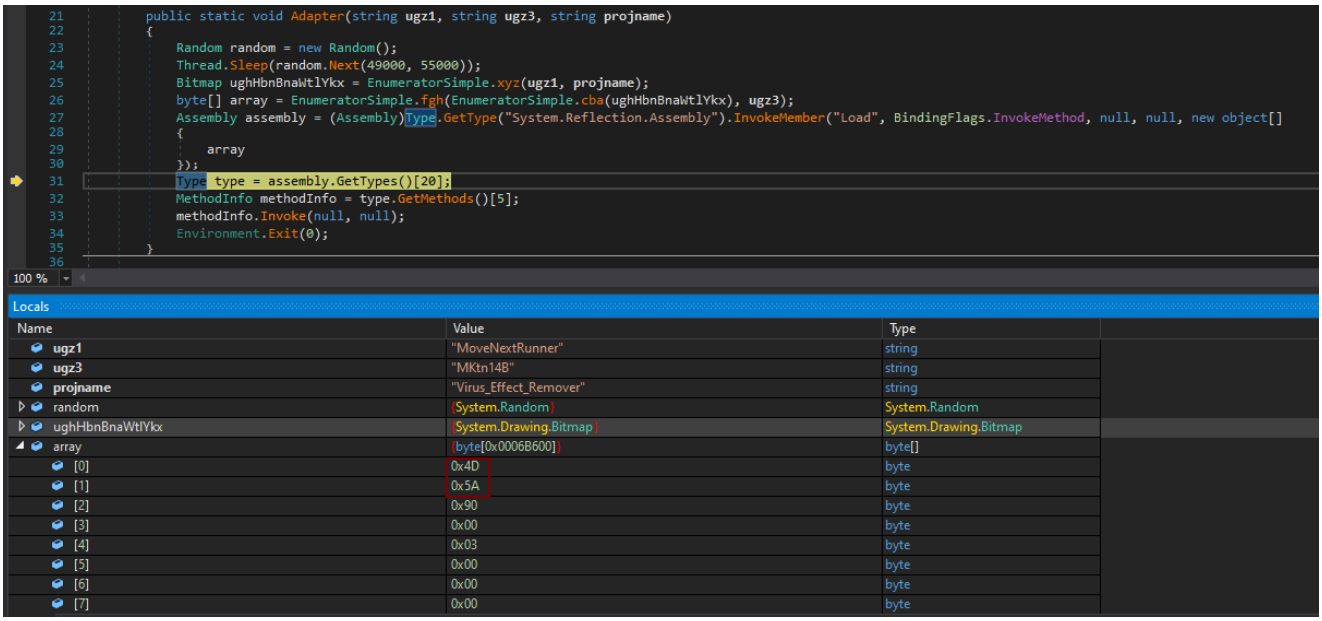
```
307         private byte[] Overide(Bitmap data)
308         {
309             List<byte> list = new List<byte>();
310             checked
311             {
312                 int num = data.Size.Width - 1;
313                 for (int i = 0; i <= num; i++)
314                 {
315                     int num2 = data.Size.Height - 1;
316                     for (int j = 0; j <= num2; j++)
317                     {
318                         Color pixel = data.GetPixel(i, j);
319                         Color right = Color.FromArgb(0, 0, 0, 0);
320                         bool flag = !(pixel == right);
321                         if (flag)
322                         {
323                             list.Add(pixel.R);
324                             list.Add(pixel.G);
325                             list.Add(pixel.B);
326                         }
327                     }
328                 }
329                 return list.ToArray();
330             }
331         }
```

100 %

Locals

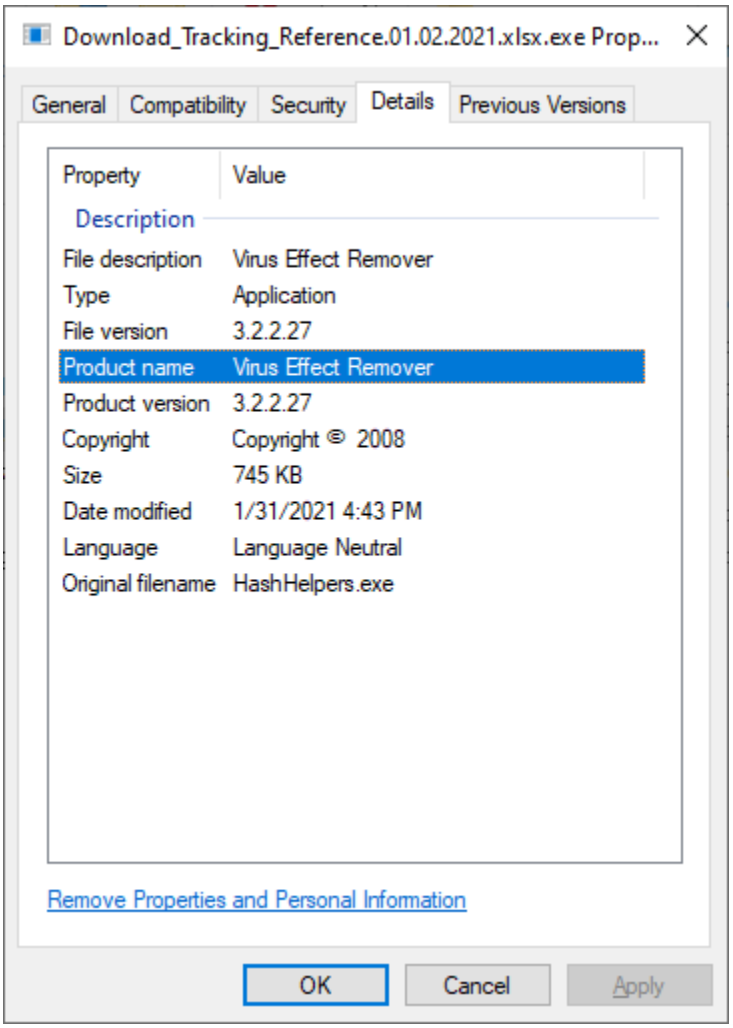
Name	Value
list	Count = 0x00002802
[0]	0x4D
[1]	0x5A
[2]	0x90
[3]	0x00
[4]	0x03
[5]	0x00
[6]	0x00
[7]	0x00
[8]	0x04
[9]	0x00
[10]	0x00
[11]	0x00
[12]	0xFF

Using this first DLL, the malware would then decode, decrypt and load a much larger (430kB) DLL called *PositiveSign.dll*.



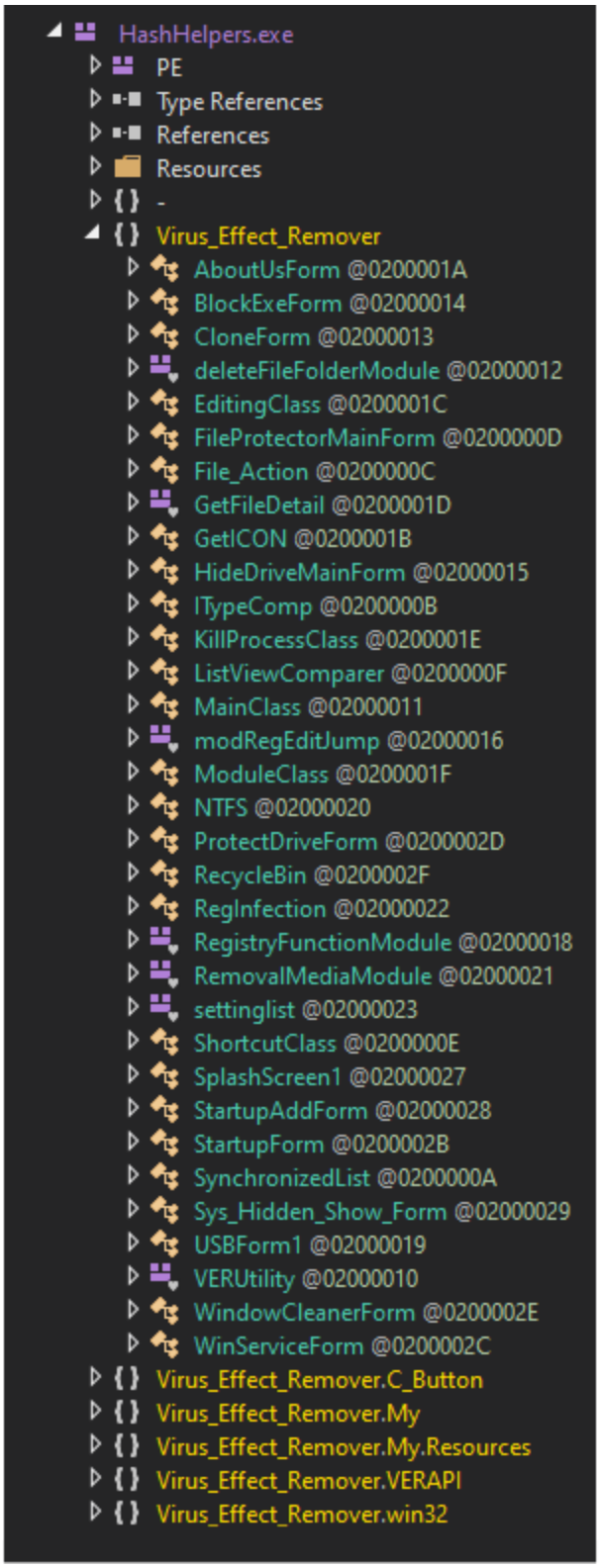
Since the second DLL was heavily obfuscated and its authors used couple of anti-analysis techniques in it, I didn't have time to go through it in detail, but from the portions of the code I saw, it did appear to contain the final stage of the payload.

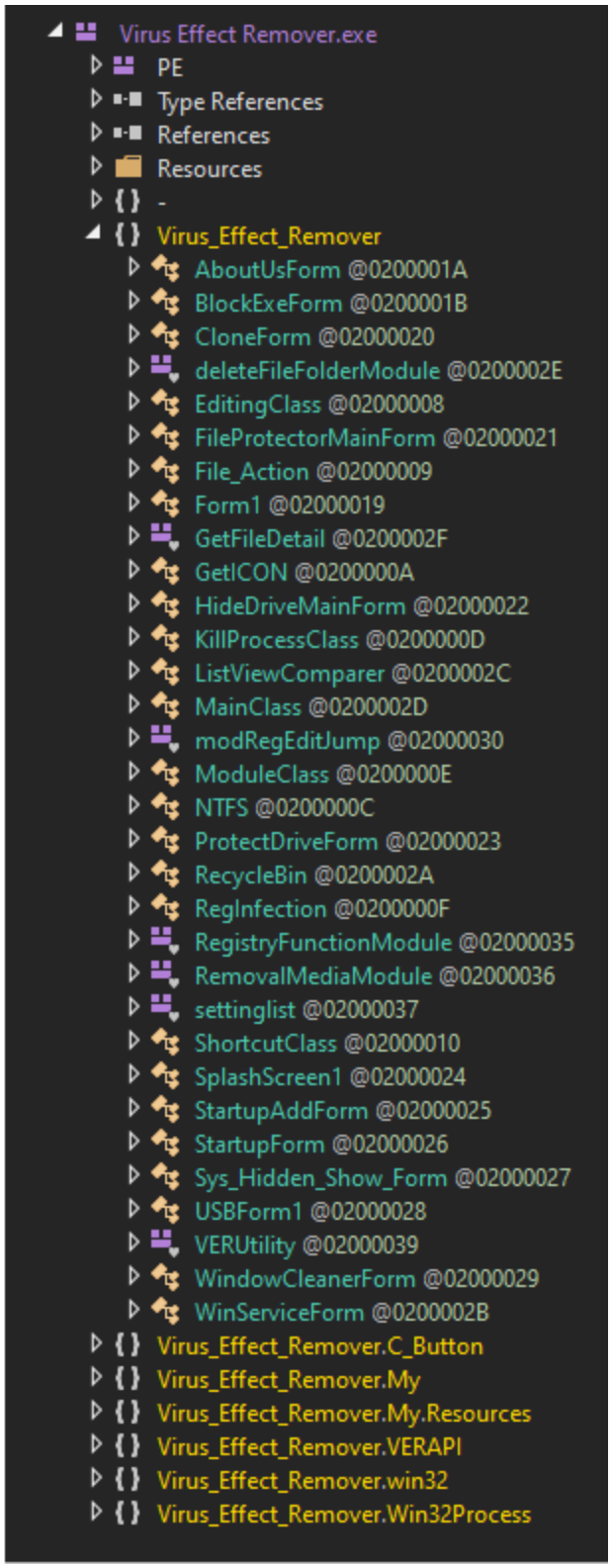
What turned out to be even more interesting than the use bitmaps to store encoded/encrypted DLLs, however, was the code of the original executable, in which the "malicious bitmaps" were hidden. The EXE, which was originally named *HashHelpers.exe*, had its description and product name set to *Virus Effect Remover*. This was a name of a legitimate anti-malware tool developed during the 2000s and first half of 2010s.



This, by itself, would not be that unusual, since malware authors sometimes like to name their creations in creative or provocative ways. Nevertheless, in this case, the name wasn't the only thing which authors of Agent Tesla borrowed from the anti-malware tool… They reused significant portions of its code as well.

When comparing the malicious file with the latest available release of the real tool[3], it can be clearly seen that large parts of both binaries are (nearly) identical.





Although the original code in the malicious EXE is never executed, authors of the malware reused large parts of it when making their creation. Since Virus Effect Remover was also written in VB.NET, getting to the code and repurposing it, even if they were working from a compiled executable, would of course be trivial for them.

Even though use of "trojanized" security tools is not a novel concept by any means, I think this was the first time I've seen it done in this way – i.e. by using code of an old anti-malware solution without trying to pass the resulting executable to target users as the original tool.

While we can only speculate on why creators of the malicious code chose to hide it in a code of a historical security tool, by far the most probable explanation seems to be that this was done in an attempt to make the malware seem benign to anti-malware scanners. And since some security tools use signature-based allow-listing mechanisms to avoid scanning of known security tools, this might have actually worked in some instances...

Indicators of Compromise (IoCs)
Download_Tracking_Reference.01.02.2021.xlsx.iso (806 kB)
MD5 - 2ceb9c4347aed5dd387d261b40473f46
SHA-1 - d4b93dd1bfb531b228353451977185f039407741

Download_Tracking_Reference.01.02.2021.xlsx.exe / HashHelpers.exe (745 kB)
MD5 - 9417df6dc7d716b0b69e587c9d89981b
SHA-1 - e905472faad91b87dbfc7afc838564fde3c87aa3

BestFit.dll (10 kB)
MD5 - a32a0b1cc226475671801360f6c53419
SHA-1 - aa6d74a2db3c430175e79f581afc29240b17ae6c

PositiveSign.dll (430 kB)
MD5 - 8b1e495e40571a5912f672f38f47058d
SHA-1 - c900af54932bdd4c8fd749cecb5689e7e2082037

[1] https://www.zscaler.com/blogs/security-research/linkedin-job-seeker-phishing-campaign-spreads-agent-tesla
[2] https://www.virustotal.com/gui/file/101399675ec99fcca0b69a0d6c146431c3a28c10d322499c817b2197e86971b5/detection
[3] https://sourceforge.net/projects/viruseffectremo/

Jan Kopriva
@jk0pr
Alef Nula

Keywords: Agent Tesla DLL Malware
1 comment(s)

Top of page
×

Diary Archives