Triage2.exe Analysis

trevor achtermann

indicators and technical details

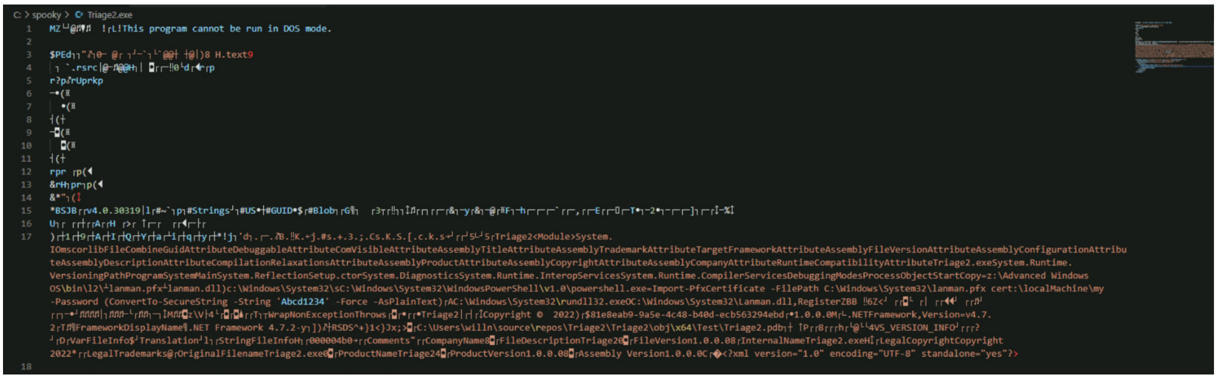
Date \ Time	Identifier	ATT&CK ID	Comment
2023-01-25 14:55:17	./Triage2.exe	TA0004	Triage2.exe was executed on the host machine with administrator level access
2023-01-25 14:55:18	Lanman.dll rundll32.exe PowerShell.exe Abcd1234 Lanman.pfx	TA0006	powershell.exe initiated a personal Information exchange file import, automatically selected the local machine and passed the string Abcd1234 as a password, successfully importing the pfx certificate in an automated fashion
2023-01-25 14:56:00	procsrv.exe 162.125.6.18:http	T1560 ?	TCP traffic observed between procsrv.exe and 162.125.6.18:http, which is associated with dropbox.com

executive summary

on January 25 2023 at 14:55:17 an executable file [Triage2.exe] was executed with administrator level access. We believe the executable to be a powershell script that automates a personal information exchange certificate import to the local machine. However, odd file and network behavior following the execution has caused concern among the security team.

technical summary

Opening Triage2.exe in a text editor program yielded some interesting results. While the code that makes up the executable isn't human readable at first glance, filtering the output as C, the language most windows applications run on, as well as removing the clutter by selecting a problematic character, and using ctrl -shift -l to select all of the associated characters and delete them makes it much easier to see how the program is written.



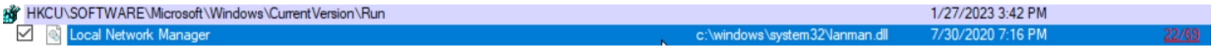
```
MZ└@♬▼♬  !┌L!This program cannot be run in DOS mode.

$PEd┐┐"♁┐0─ @┌ ┐┘─`┐└`@@┼ ┼@|)8 H.text9
 ┐ `.rsrc|@─♬@@╫| ◘┌┌─‼0└d┌◄┌p
r?p♁rUprkp
─•(Ж
  •(Ж
┤(┼
─◘(Ж
  ◘(Ж
┤(┼
rpr ┌p(◄
&rH┐pr┐p(◄
&*"┐(↕
*BSJB┌┌v4.0.30319|l┌#~`┐p┐#Strings┘┐#US•┼#GUID•$┌#Blob┐┌G¶┐  ┌3┬┌‼┐┐↕♬┌┐┌─┌&┐─y┌&┐─@┌ЖF┐─h┌─┌─┌─`┌┌─,┌┌─E┌┌─0┌─T•┐─2•┐─┌─┌─]┐┌─┌↕─%↕
U┐┌ ┌┌┼┌┌A┌┌H ┌>┌ ↑┌─┌  ┌┌◄┌─┼┌
)┌┼1┌┼9┌┼A┌┼I┌┼Q┌┼Y┌┼a┌┴i┌┼q┌┼y┌┼*!j┐'d┐.┌─.♁B.‼K.←j.#s.+.3.;.Cs.K.S.[.c.k.s↲┌┌┘5└┘5┌Triage2<Module>System.
IOmscorlibFileCombineGuidAttributeDebuggableAttributeComVisibleAttributeAssemblyTitleAttributeAssemblyTrademarkAttributeTargetFrameworkAttributeAssemblyFileVersionAttributeAssemblyConfigurationAttribu
teAssemblyDescriptionAttributeCompilationRelaxationsAttributeAssemblyProductAttributeAssemblyCopyrightAttributeAssemblyCompanyAttributeRuntimeCompatibilityAttributeTriage2.exeSystem.Runtime.
VersioningPathProgramSystemMainSystem.ReflectionSetup.ctorSystem.DiagnosticsSystem.Runtime.InteropServicesSystem.Runtime.CompilerServicesDebuggingModesProcessObjectStartCopy=z:\Advanced Windows
OS\bin\l2\┴lanman.pfx┴lanman.dll)c:\Windows\System32\sC:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe=Import-PfxCertificate -FilePath C:\Windows\System32\lanman.pfx cert:\localMachine\my
-Password (ConvertTo-SecureString -String 'Abcd1234' -Force -AsPlainText)┌AC:\Windows\System32\rundll32.exeOC:\Windows\System32\Lanman.dll,RegisterZBB ‼6Z<┘ ┌┌◘└ ┌| ┌┌◄◄┘ ┌┌♬┘
┌┌─•┘♬♬♬|┐♬♬─└┌♬♬┐─┐↕M♬♬◘z\V|4└┌◘┌◘♣┌┌T┐┬WrapNonExceptionThrows┌◘┌•┌┌•Triage2|┌┤┌↕Copyright ©  2022)┌$81e8eab9-9a5e-4c48-b40d-ecb563294ebd┌•1.0.0.0M┌└.NETFramework,Version=v4.7.
2┌T♬‖FrameworkDisplayName¶.NET Framework 4.7.2-y┐])♁┼RSDS^+}1<}Jx;>◘┌C:\Users\willn\source\repos\Triage2\Triage2\obj\x64\Test\Triage2.pdb┐┼ ↑P┌┌8┌┌┌h┌└@└└4VS_VERSION_INFO┘┌┌┌?
┘┌D┌VarFileInfo$┘Translation┘l┐┌StringFileInfoH┐┌000004b0→┌┌Comments"┌┌CompanyName8◘┌FileDescriptionTriage20◘┌FileVersion1.0.0.08┌InternalNameTriage2.exeH↕┌LegalCopyrightCopyright
2022*┌┌LegalTrademarks@┌OriginalFilenameTriage2.exe0◘┌ProductNameTriage24◘┌ProductVersion1.0.0.08◘┌Assembly Version1.0.0.0C┌�<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```

technical summary (cont.)

The first thing that stood out as pertinent within the system during our investigation was a persistence mechanism registry written to the HKCU/SOFTWARE/Windows/CurrentVersion/Run. This was observed in autoruns. However, the time stamp shown lists a date in 2020 which has made it difficult to associate this with the initial file execution.



```
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run                                   1/27/2023 3:42 PM
☑  Local Network Manager                       c:\windows\system32\lanman.dll      7/30/2020 7:16 PM    22/69
```

The following file path can be observed in the screenshot of the executable codebase mentioned above as well: Z:\Advanced Windows OS\bin\l2\lanman.pfx lanman.dll

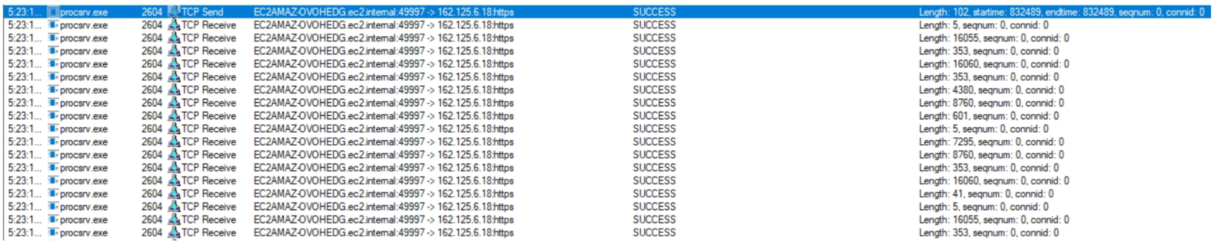
After navigating to the directory, you can find both lanman.pfx and Lanman.dll among a variety of other files with currently unknown purpose.

- the odd directory for this Lanman.dll file is concerning. However, the persistence mechanism written to the registry points to the file located on the system32 directory. For this reason we believe the Lanman.dll file above the lanman.pfx file is some form of software that is written to the host machines system32 directory and then called upon each time the host machine runs. At this time we are unsure if the software is acting maliciously, but lanman.dll is not a native system32 file, so this is important to note.

We observed another process acting on the network level, titled procsrv.exe. Connections are made between the executable and IP : 162.125.6.18

IP: 162.125.6.18 is associated to a dropbox.com account

Most of the connections are tcp receptions from dropbox.com to the host machine, and the few send packets from the host to dropbox appear almost just like pings to keep letting dropbox know the host machine is there.



```
5:23:1...  procsrv.exe  2604  TCP Send     EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 102, startime: 832489, endtime: 832489, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 5, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 16055, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 353, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 16060, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 353, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 4380, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 8760, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 601, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 5, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 7295, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 8760, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 353, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 16060, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 41, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 5, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 16055, seqnum: 0, connid: 0
5:23:1...  procsrv.exe  2604  TCP Receive  EC2AMAZ-OVOHEDG.ec2.internal:49997 -> 162.125.6.18:https  SUCCESS  Length: 353, seqnum: 0, connid: 0
```

This is where things started to get interesting. Using powershell, we determined the file hash for each of the 3 pertinent files:

Triage2.exe: C3EA45EFE97E7D50D73E22E0EA40D7FCCA0CC0FDA04A9F76A8606F4C4D8BB535

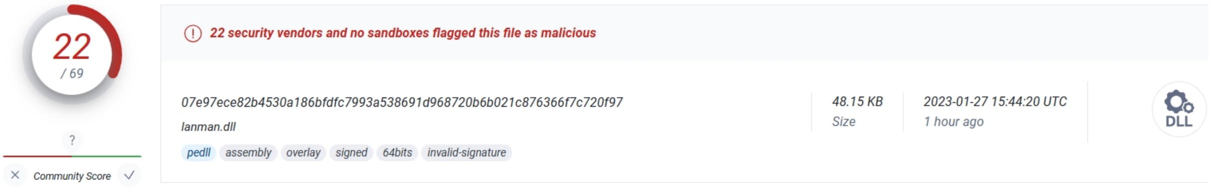
lanman.dll : 07E97ECE82B4530A186BFDFC7993A538691D968720B6B021C876366F7C720F97

Lanman.pfx: C9E42837C0C5382D279097B968916B0378C8CC0F745BE890794356D3E7F5878E

The hash for Triage2.exe was flagged by zero sandboxes or security vendors, and the same was true for lanman.pfx. However, when scanned by VirusTotal, the hash for Lanman.dll was flagged as malicious by 31 security vendors but by 0 sandboxes. After checking a different windows machine's system32 directory and finding it to be missing, we believe the program is adding lanman.dll to system32 from the bin folder located on Z:drive.

technical summary (cont.)

This has made the security team more suspicious of the Lanman.dll in question, however virus checkers like virus total can be known to light up when loaded with code that uses native windows functions, as many malware producers abuse these functions to gain the access that they're after.



22 security vendors and no sandboxes flagged this file as malicious

07e97ece82b4530a186bfdfc7993a538691d968720b6b021c876366f7c720f97

lanman.dll

48.15 KB Size

2023-01-27 15:44:20 UTC
1 hour ago

findings and analysis

Input	Output
strings lanman.dll > C:\Users\Administrator.EC2AMAZ-OVOHEDG\Desktop\lanmanstrings.txt	dumps all strings for lanman.dll to a text file on the desktop for investigation using powershell
Ctrl - Shift - L	Selects all of a selected character within a visual studio code

remediation and recommendation

With the potential for the virus scanner false positive and without knowing what the network traffic consists of without deeper investigation, we recommend these remediation steps if you encounter Triage2.exe on your machine and are unsure of where it came from \ what its doing on the system:

> open either task manager or process explorer (procexp)

> find the process named procsrv.exe and end the process

> open regedit

> navigate to Computer\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

> find the key named Local Network Manager. Right click and delete the key

> navigate to Z:\Advanced Windows OS\

> delete the folder named bin

> delete Triage2.exe from the directory in which it was found

> delete lanman.dll from C:\windows\system32

> empty recycle bin

> restart machine and ensure the key from step 4 is still gone and procsrv.exe is not running