# Real-Time Prevention of the Kaseya VSA Supply Chain REvil Ransomware Attack

blog.morphisec.com/real-time-prevention-of-the-kaseya-vsa-supply-chain-revil-ransomware-attack

Posted by Morphisec Labs on July 5, 2021

- Tweet
-

## Introduction

On July 2, 2021, Morphisec Keep, our Cloud Workload Protection Platform, successfully identified and prevented a REvil Ransomware infection within some of our customer domains. This attack was automatically blocked in real time due to Morphisec's proactive protection mechanism, which resulted in no harm to the customer.

Attacks like this demonstrate the critical nature of a strong prevention strategy for servers. Any such strategy needs to acknowledge that supply chain attacks such as this one can often bypass the most up-to-date network, identity, and antivirus controls. The following is a summary of our findings after preventing this attack in several customer environments.

## Technical details

### Initial Findings

Most of the attacked endpoints were Windows servers. The Morphisec Keep product immediately and automatically identified the process chain that led to the prevented ransomware execution as shown in the screenshot below. This attack is particularly evasive because all the attack chain components are signed with digital certificates, starting from the Kaseya process, continuing with a vulnerable Microsoft Defender process, and ending with the side-loaded signed ransomware.

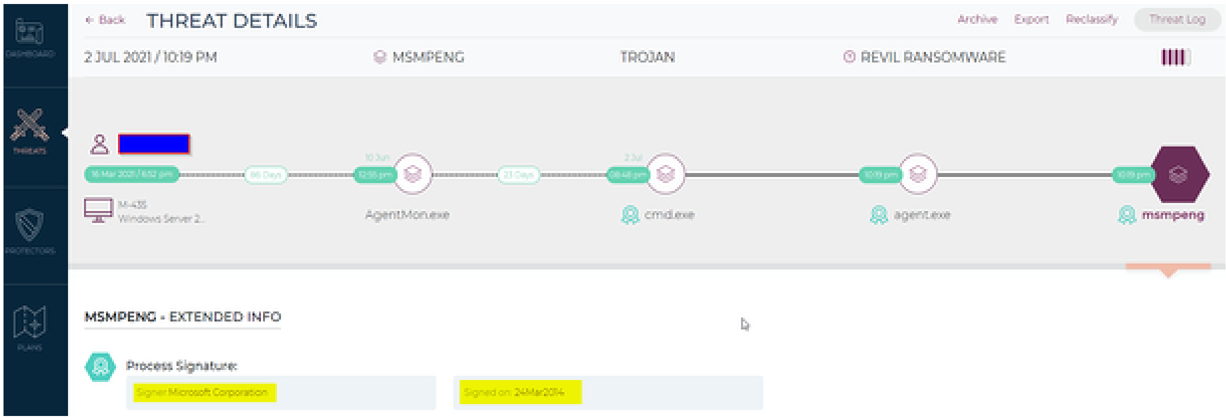


Figure 1: The threat summary in the Morphisec GUI

The initial stage begins with *AgentMon.exe* – a legitimate process for the *Kaseya Virtual Server Administrator Agent*. Then this process executes suspicious Windows batch commands via *cmd.exe*.

Those batch commands are followed by the execution of a malicious file named *agent.exe*. The following process then executes a legitimate signed Windows Defender process (*MsMpEng.exe*) that was dropped in the “*C:\Windows\”* directory.

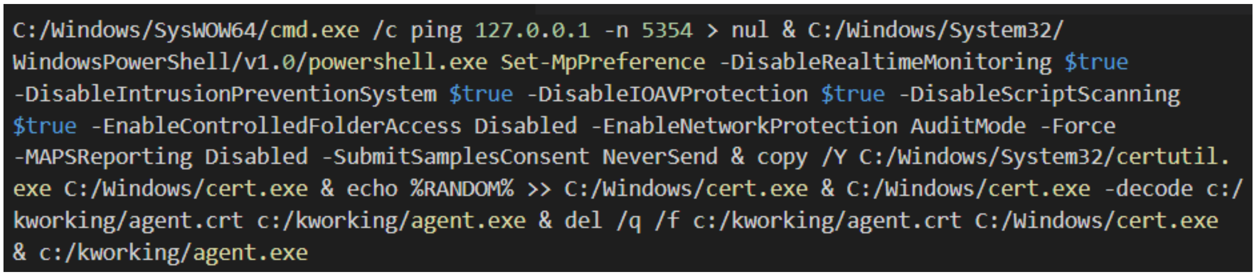
This Windows Defender application file is known for its side-loading vulnerability that has since been patched. Once executed it loads a malicious DLL file named *mpsvc.dll* that was also dropped within the Windows directory.

This *mpsvc.dll* DLL is the actual functionality of the ransomware. Morphisec automatically blocked its execution and prevented the servers from being infected.

## The Batch Commands

These commands are executed after the *agent.crt* file is dropped by a hijacked update routine called 'Kaseya VSA Agent Hot-fix’.

The first part sleeps for about 1.5 hours by utilizing the ping command to localhost. Then, it executes a PowerShell command that first disables Microsoft Defender security features, such as Real-Time Protection, Intrusion Prevention System, Script Scanning, Automatic Sample Submission, Controlled Folder Access, and network protection.



```
C:/Windows/SysWOW64/cmd.exe /c ping 127.0.0.1 -n 5354 > nul & C:/Windows/System32/
WindowsPowerShell/v1.0/powershell.exe Set-MpPreference -DisableRealtimeMonitoring $true
-DisableIntrusionPreventionSystem $true -DisableIOAVProtection $true -DisableScriptScanning
$true -EnableControlledFolderAccess Disabled -EnableNetworkProtection AuditMode -Force
-MAPSReporting Disabled -SubmitSamplesConsent NeverSend & copy /Y C:/Windows/System32/certutil.
exe C:/Windows/cert.exe & echo %RANDOM% >> C:/Windows/cert.exe & C:/Windows/cert.exe -decode c:/
kworking/agent.crt c:/kworking/agent.exe & del /q /f c:/kworking/agent.crt C:/Windows/cert.exe
& c:/kworking/agent.exe
```

Figure 2: Batch command execution by AgentMon.exe

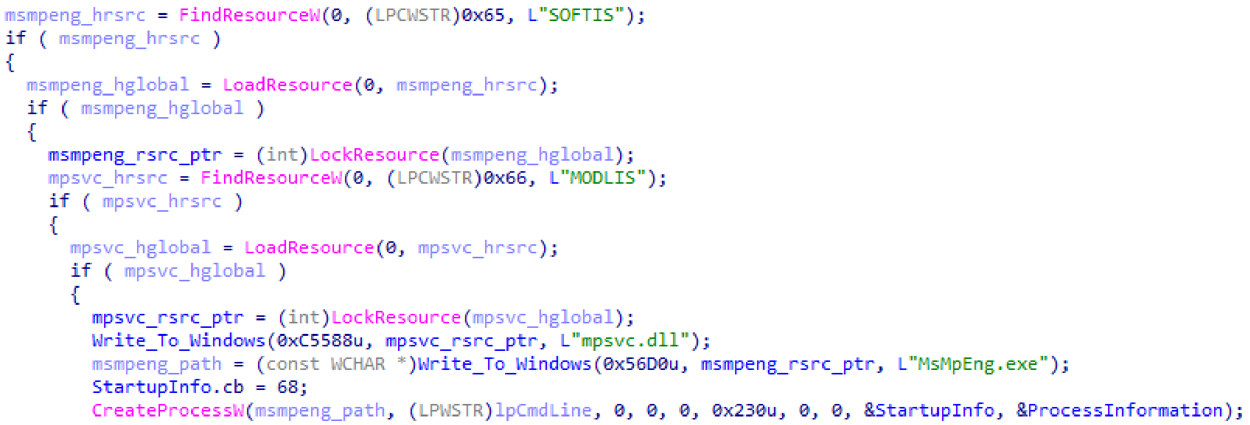
Next, it copies *certutil.exe,* a legitimate executable under the name *cert.exe,* and adds a random value to the end of the executable – this breaks the detection of *certutil* utilization by third-party vendors. It will later use this executable to decode *agent.crt*, delete it, and execute the decoded payload: *agent.exe*.

## The Loader (agent.exe)

The loader is signed with a certificate named "PB03 TRANSPORT LTD," a basic loader without any obfuscations. The loader contains two resources:

- 'SOFTIS': Contains *MsMpEng.exe*, the legitimate executable
- 'MODLIS': Contains *mpsvc.dll*, the malicious payload

Once the resources are loaded into memory, the loader writes these files to C:\Windows and executes the MsMpEng.exe file (as System).

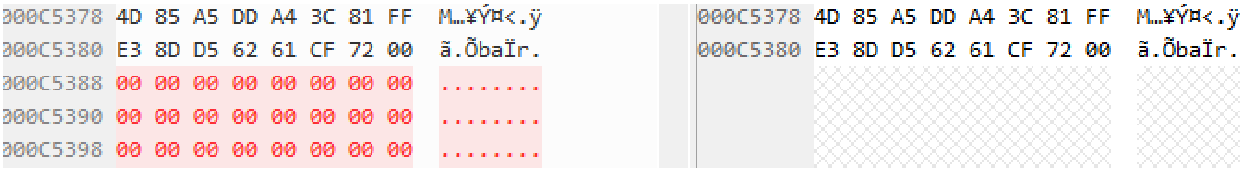


```
msmpeng_hrsrc = FindResourceW(0, (LPCWSTR)0x65, L"SOFTIS");
if ( msmpeng_hrsrc )
{
  msmpeng_hglobal = LoadResource(0, msmpeng_hrsrc);
  if ( msmpeng_hglobal )
  {
    msmpeng_rsrc_ptr = (int)LockResource(msmpeng_hglobal);
    mpsvc_hrsrc = FindResourceW(0, (LPCWSTR)0x66, L"MODLIS");
    if ( mpsvc_hrsrc )
    {
      mpsvc_hglobal = LoadResource(0, mpsvc_hrsrc);
      if ( mpsvc_hglobal )
      {
        mpsvc_rsrc_ptr = (int)LockResource(mpsvc_hglobal);
        Write_To_Windows(0xC5588u, mpsvc_rsrc_ptr, L"mpsvc.dll");
        msmpeng_path = (const WCHAR *)Write_To_Windows(0x56D0u, msmpeng_rsrc_ptr, L"MsMpEng.exe");
        StartupInfo.cb = 68;
        CreateProcessW(msmpeng_path, (LPWSTR)lpCmdLine, 0, 0, 0, 0x230u, 0, 0, &StartupInfo, &ProcessInformation);
```

Figure 3: The agent.exe loader execution flow

## Bug in Production?

As others have written, there are two different hashes for *mpsvc.dll* that represent the same payload. The only difference is that one of them is padded with nulls at the end.



```
000C5378 4D 85 A5 DD A4 3C 81 FF  M…¥Ý¤<.ÿ        000C5378 4D 85 A5 DD A4 3C 81 FF  M…¥Ý¤<.ÿ
000C5380 E3 8D D5 62 61 CF 72 00  ã.ÕbaÏr.        000C5380 E3 8D D5 62 61 CF 72 00  ã.ÕbaÏr.
000C5388 00 00 00 00 00 00 00 00  ........
000C5390 00 00 00 00 00 00 00 00  ........
000C5398 00 00 00 00 00 00 00 00  ........
```

Figure 4: The differences between mpsvc.dll files

- The instance without the nulls is the DLL that was extracted from the 'MODLIS' resource
- The instance with the null padding is the one saved within the Windows directory

While the unpadded instance is digitally signed, the padding breaks the file digital signature in the other instance, so what has gone wrong?

The actual size of the DLL within the resource is 807,816 bytes while the size of bytes for writing this file within the loader is 808,328 bytes (0xC5588 in the agent.exe execution flow figure). This difference results in the null padding that breaks the digital signature and leads to a potential successful detection by some vigilant vendors.

## Conclusion

Attacks like this demonstrate the importance of real-time prevention that does not rely on signatures by leveraging zero-trust at the endpoint. As shown in the first screenshot, all components of this attack are signed. This attacker is adept at abusing the implicit trust given to signed processes so their attacks can progress in target environments. Furthermore, detection-centric technology, like EDR, cannot be relied on when malicious activity is present on servers. The attacker is simply too close to their final goal for it to make sense to rely on reactive remediation through human intervention.

Morphisec Keep is built to deal with evasive threats like this automatically, in real time, and without prior knowledge of the attack. Through Morphisec Keep, you can extend your zero-trust strategy beyond identity and the network so that attacks like this one, where the supply chain is compromised, can still be prevented when they can land on the endpoint and make their way into the process memory.

## IOCs

agent.exe (REvil Loader):
d55f983c994caa160ec63a59f6b4250fe67fb3e8c43a388aec60a4a6978e9f1e

mpsvc.dll saved on disk (REvil payload):

8dd620d9aeb35960bb766458c8890ede987c33d239cf730f93fe49d90ae759dd

mpsvc.dll within agent.exe resource (REvil payload):

e2a24ab94f865caeacdf2c3ad015f31f23008ac6db8312c2cbfb32e4a5466ea2

Contact SalesInquire via Azure