Tracking down Maggie

By DCSO CyTec Blog

Published: 2022-10-13 · Archived: 2026-04-05 15:26:41 UTC

In our recent blog post “MSSQL, meet Maggie” we shared our research on a novel backdoor malware targeting Microsoft SQL servers that DCSO CyTec refers to as “Maggie“.

Press enter or click to view image in full size

Tracking Down “Maggie”

Maggie can be summarised as malware that comes in form of an “Extended Stored Procedure” DLL, a special type of extension used by Microsoft SQL servers. Once loaded into a server by an attacker, it is controlled solely using SQL queries and offers a variety of functionality to run commands, interact with files and function as a network bridge head into the environment of the infected server.

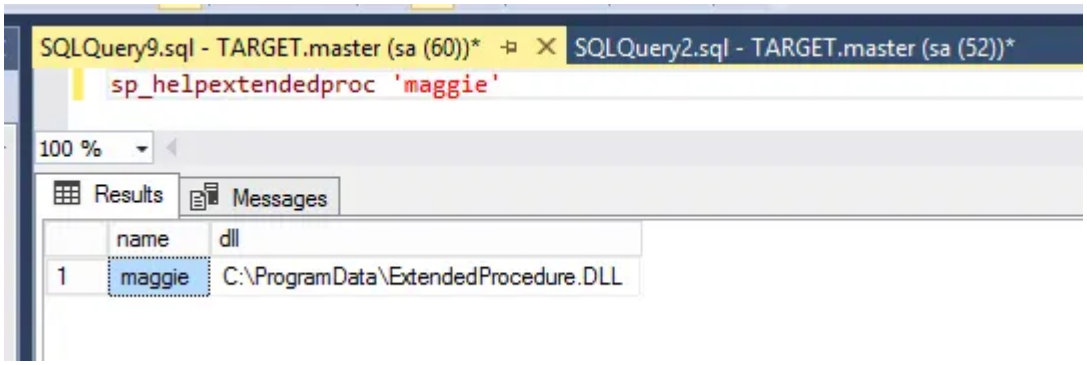
In this blog post DCSO’s Incident Response Team (DIRT) aims to provide security teams with some insights on how to detect this novel threat in their environment.

Blog authored by Denis Szadkowski, Johann Aydinbas and Axel Wauer

Am I affected?

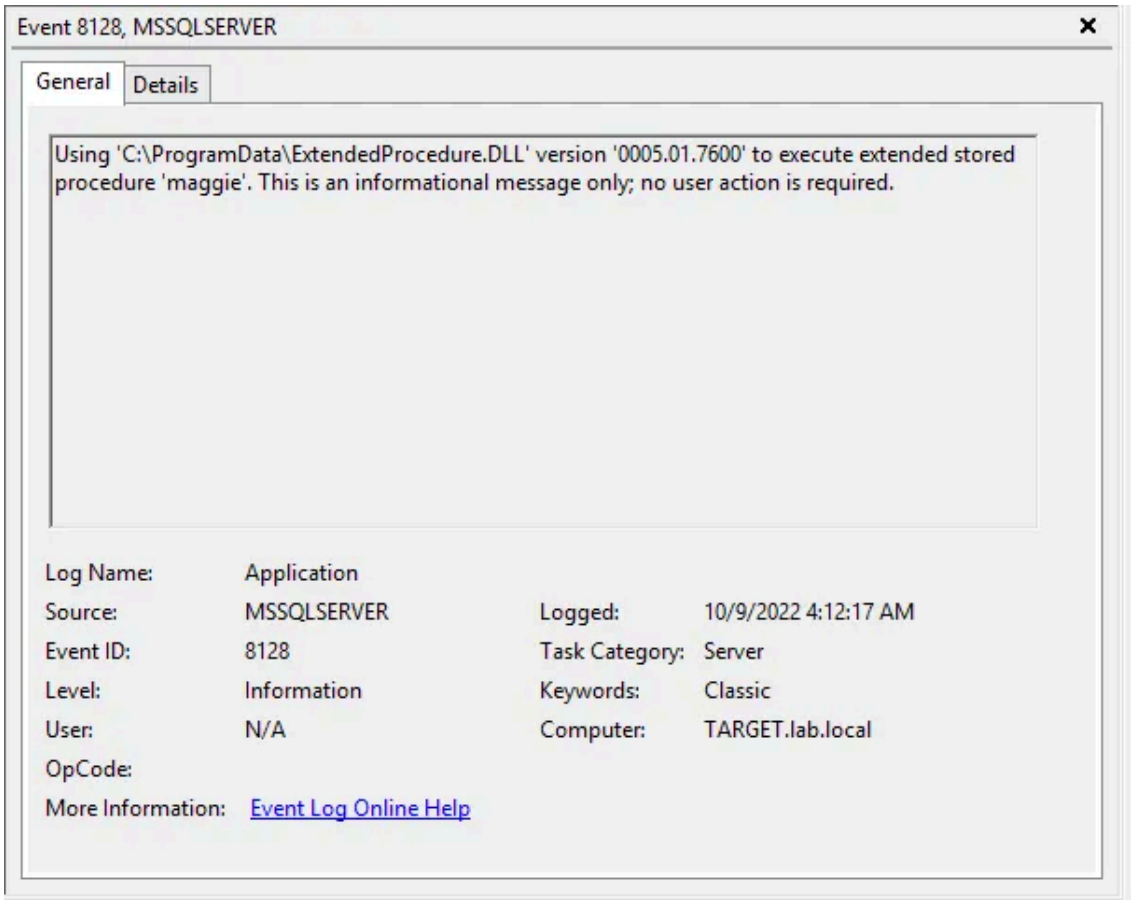
The first question that probably jumped into the minds of the majority of our readers after reading our blog post “MSSQL, meet Maggie” was “Are my Microsoft SQL servers affected by this threat?”.

There are multiple ways of identifying the presence of this threat in your environment. A good starting point for Microsoft SQL administrators is to review the “Extended Stored Procedures” (ESP) installed on their Microsoft SQL servers. One way of doing that is to use a “System Stored Procedures” called `sp_helpextendedproc`. By specifying the name of the ESP it is possible to quickly check if the Microsoft SQL server is affected as shown in the following figure. If the SQL query does not return any results this means there is no ESP with the name “maggie”.



Using sp_helpextendedproc to check the presence of Maggie

Another way of verifying if a Microsoft SQL server has been backdoored by this threat actor is to check the application log. When a ESP is executed, the event id `8128` is logged in the application log of the system as can be seen in the figure below.



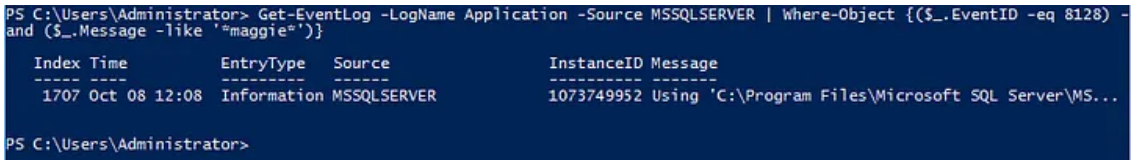
Execution of the extended stored procedure “maggie”

Furthermore, the application log can easily be searched with the help of the following PowerShell one-liner. The presented one-liner can also be used in combination with the PowerShell Cmdlet `Invoke-Command` to search application logs across multiple systems.

```
Get-EventLog -LogName Application -Source MSSQLSERVER | Where-Object {($_.EventID -eq 8128) -and ($_
```

If the Microsoft SQL server is affected by this threat the PowerShell one-liner will return output similar to the one presented in the figure below.

Press enter or click to view image in full size



PowerShell one-liner output for affected MSSQL server

In addition to event id `8128` the usage of the ESP will generate the event `33090` that provides information about the ESP DLLs that were recently loaded into the memory of the Microsoft SQL server process `sqlservr.exe`.

Once the presence of the ESP named “maggie” has been confirmed by using one of the methods described above, it is recommended to check if the threat actor was able to create a backdoor database user in the Microsoft SQL server. This can be accomplished with the SQL query presented below:

```
select sp.name,
       sp.type_desc,
       sp.create_date,
       sp.modify_date,
       case when sp.is_disabled = 1 then ‘Disabled’
            else ‘Enabled’ end as status
from sys.server_principals sp order by sp.create_date DESC
```

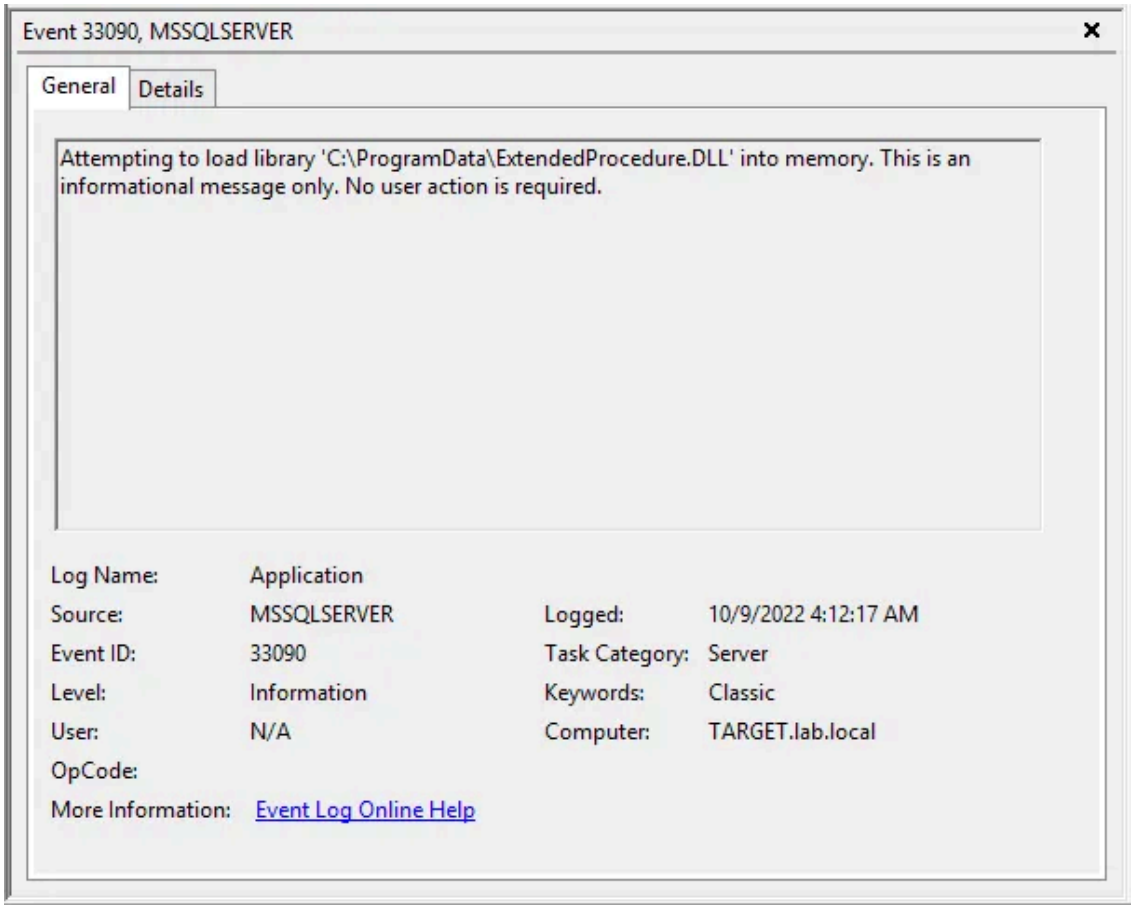
As described in our previous blog post one of the first things that *Maggie* does after successfully bruteforcing its way into a Microsoft SQL server is to check if the current database user has the administrative role assigned, if this is the case a backdoor database user is created. The SQL query above will show the most recently added users accounts. It is recommended to verify if the presented accounts are legitimate.

Identifying malicious ESPs

After the publication of our *Maggie* research it is safe to assume that the use of ESPs as a backdoor mechanism will increase among threat actors targeting Microsoft SQL servers. Moreover, it can be assumed that the threat actors behind *Maggie* will take the necessary steps to change their backdoor to make it harder for defenders to detect it. Because of that a more generic detection approach is needed to identify malicious ESPs.

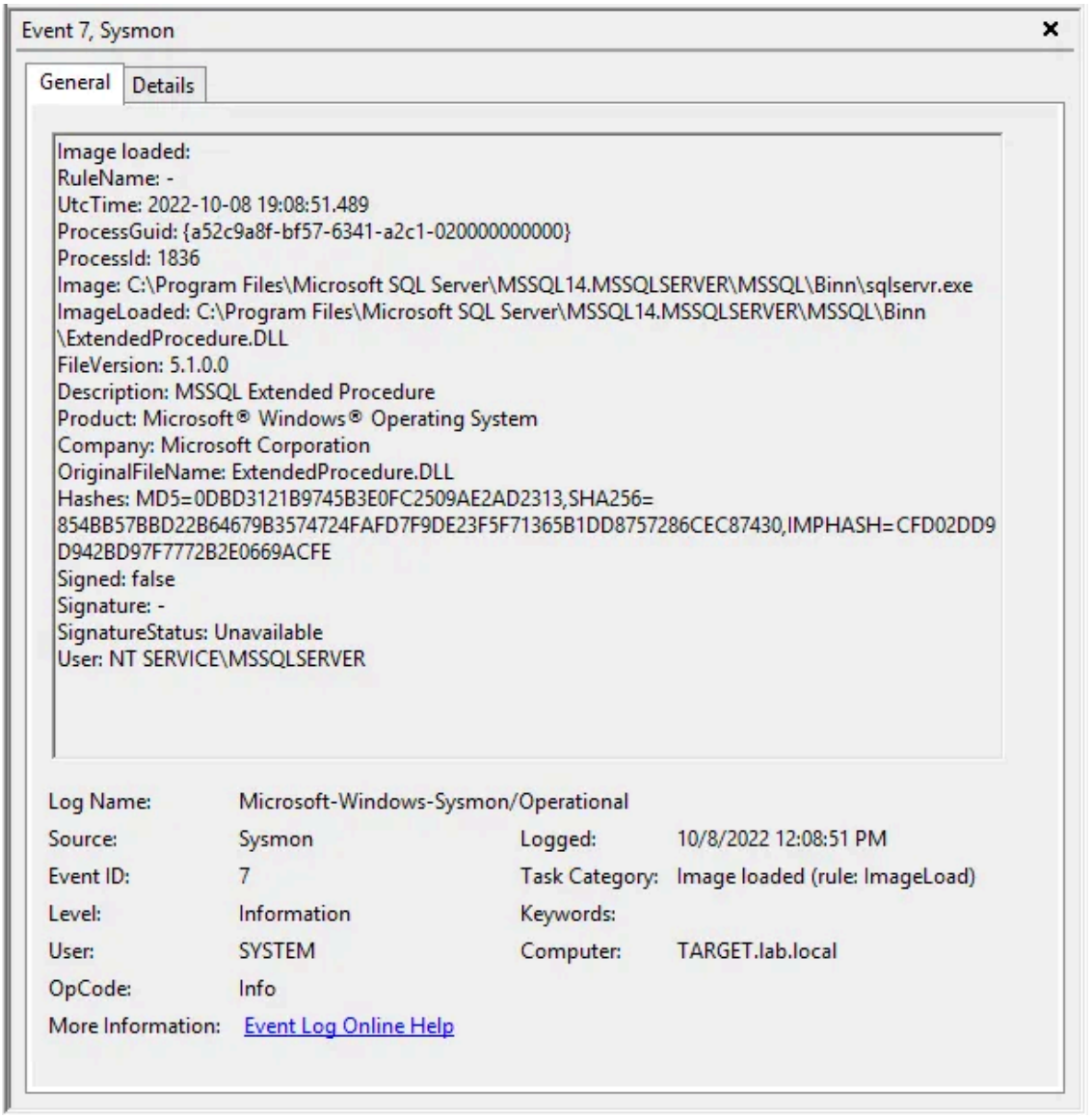
Get DCSO CyTec Blog’s stories in your inbox

One way of accomplishing this goal is to build a baseline of ESPs present in a Microsoft SQL server and then to monitor the application log for outliers with the help of event ids `8128` and `33090`. In order to fine-tune detection rules additional criteria can be added e.g., the path of the ESP. Under normal circumstances ESP are located in `C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Binn`, which is not always the case for malicious ESPs.



Malicious ESP DLL loaded into sqlservr.exe

Furthermore, tools like Sysmon from Windows Sysinternals can provide added visibility that can help to detect malicious ESPs. For example event id `7` (Image loaded) can be leveraged to identify newly loaded ESP DLLs:

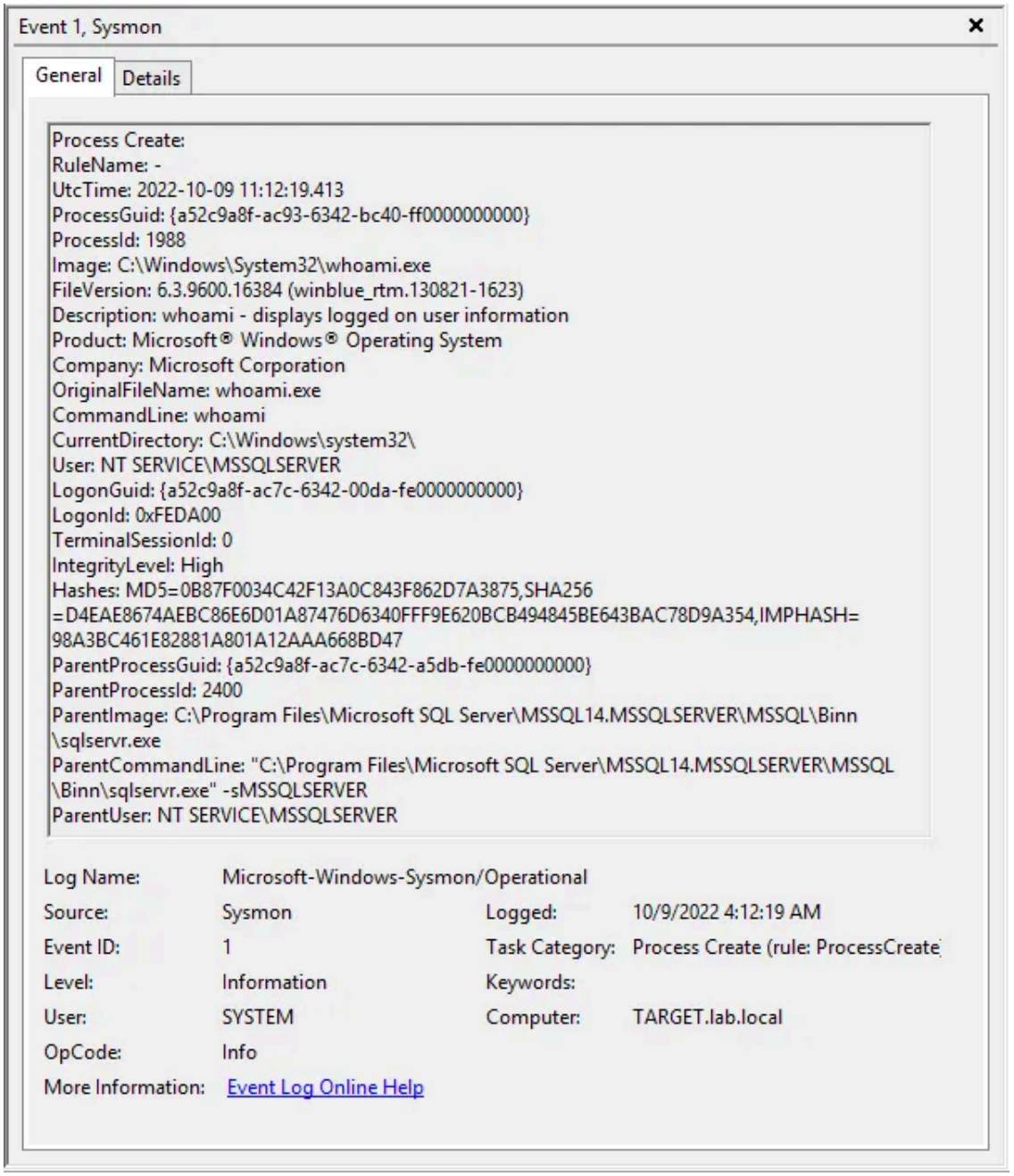


ESP DLLs loaded into `sqlservr.exe`

In addition to Sysmon event id `7`, Sysmon process creation events can be used to detect processes spawned by the malicious ESP. In the following example the malicious ESP `ExtendedProcedure.DLL` is used to execute the `whoami.exe` application to find out the current user context:

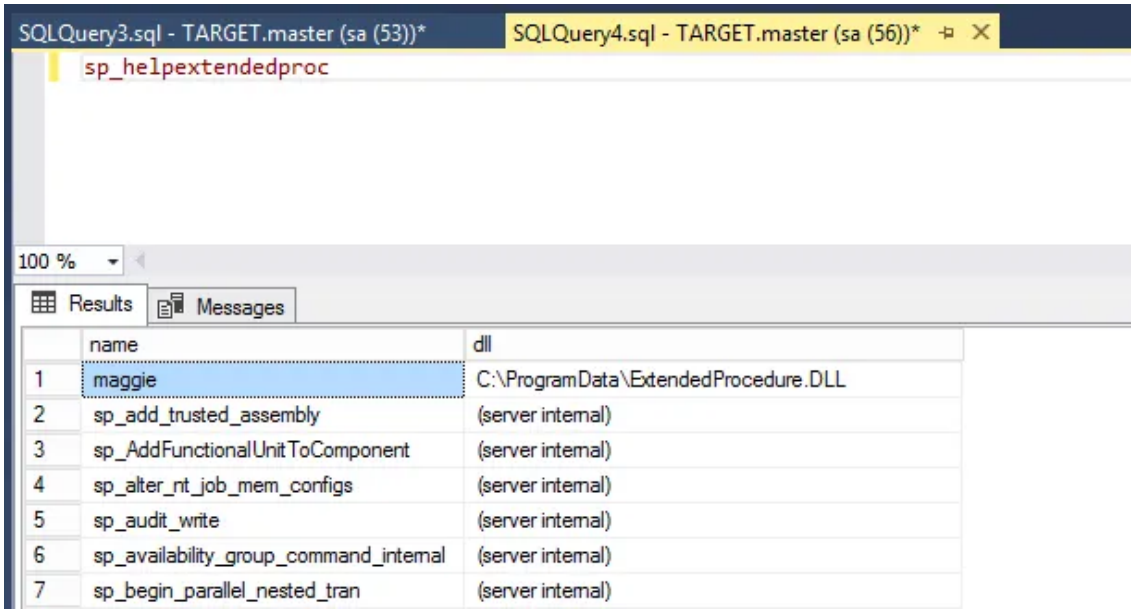
```
$ exec maggie 'Exec whoami';MSSQL Procedure 12/08/2021
Execute Command: Exec whoami
nt service\mssqlserver
```

After running the command above the following process creation event is logged by Sysmon. As presented in the figure below the `ParentImage` of the `whoami.exe` application is the Microsoft SQL server process `sqlservr.exe` which should ring a bell for the seasoned defender.



Sysmon process creation event

For organisations that use Microsoft SQL servers but are lacking a SIEM solution a good practice might be to occasionally check which ESPs are installed on their Microsoft SQL servers. This can be done by running the "System Stored Procedure" called `sp_helpextendedproc` without specifying any parameters as shown in the following figure:



	name	dll
1	maggie	C:\ProgramData\ExtendedProcedure.DLL
2	sp_add_trusted_assembly	(server internal)
3	sp_AddFunctionalUnitToComponent	(server internal)
4	sp_alter_nt_job_mem_configs	(server internal)
5	sp_audit_write	(server internal)
6	sp_availability_group_command_internal	(server internal)
7	sp_begin_parallel_nested_tran	(server internal)

Listing installed ESPs

When reviewing the installed ESPs Microsoft SQL administrators should focus on ESPs that are not part of their baseline. Another indicator of a malicious ESP might be the path in which the ESP DLL is located. Under normal circumstances ESPs are placed under `C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Binn`. In the example above the malicious ESP was placed under `C:\ProgramData` to avoid detection.

Mitigation

One of the ways that *Maggie* can compromise Microsoft SQL servers is by using its SQL bruteforce functionality. In order to thwart bruteforce attempts, it is highly recommend to use strong passwords for Microsoft SQL accounts and to enforce a strong password policy. Additionally, the Microsoft SQL database port (default: `1433`) should never be exposed to the Internet. It is sufficient when the front-end tier systems can reach the Microsoft SQL database.

You can find additional information about securing Microsoft SQL servers at the vendors web site under Securing SQL Server.

YARA

```
import "pe"

rule maggie_backdoor
{
    meta:
        description = "Detect MSSQL extended stored procedure backdoor Maggie files"
        author = "Johann Aydinbas, TEC / DCSO CyTec"
        reference = "https://medium.com/@DCSO_CyTec/mssql-meet-maggie-898773df3b01"
    strings:
        $ = "Account Owner Not Found For The SID"
```

```
        $ = "Call SO_UPDATE_ACCEPT_CONTEXT To Get IP"
        $ = "Socks5 Stopped Failure"
        $ = "It Has Been Hooked"
        $ = "AllowedIP IP Port"
        $ = "Wait 5 To 10 Seconds For TS Taking Effect"
        $ = "\\Binn\\sqlservr.exe"
        $ = "HostList UserList PassList"
        $ = "opends60.dll"
        $ = "WinSockScan"
        $ = "ResetClientData"
        $ = "ViewClientData"
        $ = "ElevateTS"
    condition:
        pe.is_pe and
        8 of them
}
```

SIGMA

```
title: MSSQL Extended Stored Procedure Backdoor Maggie
id: 711ab2fe-c9ba-4746-8840-5228a58c3cb8
description: This rule detects the execution of the extended storage procedure backdoor named Maggie
tags:
    - attack.persistence
    - attack.t1546
status: experimental
date: 2022/10/09
modified: 2022/10/09
references:
    - https://medium.com/@DCSO_CyTec/mssql-meet-maggie-898773df3b01
author: Denis Szadkowski, DIRT / DCSO CyTec
logsource:
    product: windows
    service: application
detection:
    selection:
        Provider_Name: 'MSSQLSERVER'
        EventID: 8128
        Message|contains: 'maggie'
    condition: selection
falsepositives:
    - Legitimate extended stored procedures named maggie
level: high
```

win_mssql_sp_maggie.yml

Are you affected?

Are you one of the organisations affected by this novel Microsoft SQL server backdoor? We would like to hear from you! If desired, we can help your organisation to investigate and to respond to this threat.
