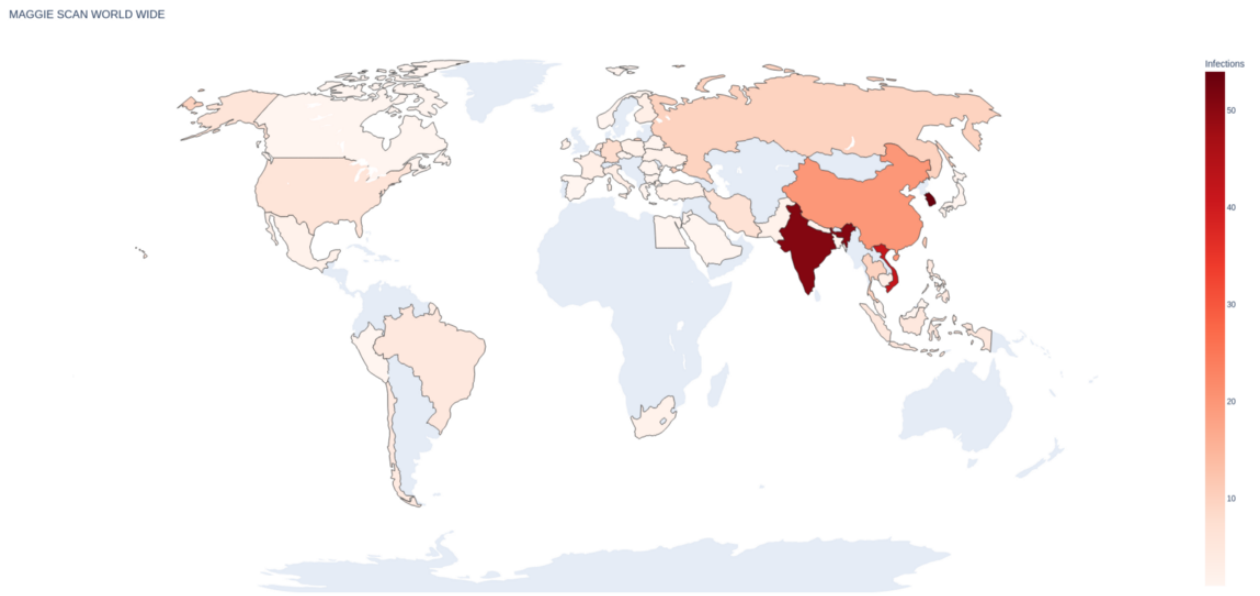
MSSQL, meet Maggie. A novel backdoor for Microsoft SQL… | by DCSO CyTec Blog | Oct, 2022

medium.com/@DCSO_CyTec/mssql-meet-maggie-898773df3b01

DCSO CyTec Blog October 11, 2022



DCSO CyTec Blog

Oct 4

.

6 min read

MSSQL, meet Maggie

Heatmap of Maggie backdoor user by country
Continuing our monitoring of signed binaries, *DCSO CyTec* recently found a novel backdoor malware targeting Microsoft SQL servers.

The malware 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 addition, the backdoor has capabilities to bruteforce logins to other MSSQL servers while adding a special hardcoded backdoor user in the case of successfully bruteforcing admin logins. Based on this finding, we identified over 250 servers affected worldwide, with a clear focus on the Asia-Pacific region.

Based on artifacts found in the malware, *DCSO CyTec* calls this novel threat “Maggie”.

In our follow-up post “” we also provide practical tips on how to detect Maggie in your environment.

Blog authored by and

Discovery

While looking for new threats, a file caught our attention. Detected as `APT_ShadowForce_Malware_ON_Nov17_1` by THOR and with a matching AV detection by AhnLab-V3 as `Trojan/Win.ShadowForce.R472810` we decided to take a closer look.

THOR detection on VirusTotal
The DLL file is signed by `DEEPSoft Co., Ltd.` on 2022–04–12. According to its export directory, the file calls itself `sqlmaggieAntiVirus_64.dll` and only offers a single export called `maggie` .

DLL export in IDA

Extended Stored Procedures

Closer inspection revealed this DLL to be an `Extended Stored Procedure` .

Extended Stored Procedures are a way to offer extended functionality to SQL queries for use in an MSSQL server, similar to the infamous xp_cmdshell stored procedure, which allows SQL queries to run shell commands.

ESPs are common DLL files using a simplistic API. When executed, ESPs are passed a handle to the client connection which allows them to fetch user-supplied arguments (via srv_paramdata) and return unstructured data to the caller (via srv_sendmsg).

Maggie utilizes this message-passing interface to implement a fully functional backdoor controlled only using SQL queries.

In order to install *Maggie*, an attacker has to be able to place an ESP file in a directory accessible by the MSSQL server, and has to have valid credentials to load the *Maggie* ESP into the server. It is unclear how an actual attack with *Maggie* is performed in the real-world.

After manually loading *Maggie* with

```
sp_addextendedproc maggie, '<path to DLL>';
```

an authenticated user could start to issue commands to the backdoor via SQL queries, e.g. to call the `whoami` shell command:

```
$ exec maggie 'Exec whoami';MSSQL Procedure 04/08/2022Execute Command: Exec
whoamiExecuting whoami Successfullynt service\mssqlserver
```

Commands

Once installed, *Maggie* offers a variety of commands to query for system information, interact with files and folders, execute programs as well as various network-related functionality like enabling TermService, running a Socks5 proxy server or setting up port forwarding to make *Maggie* act as a bridge head into the server's network environment.

The full list of commands we have identified:

List of commands available in Maggie
Commands can take multiple arguments, separated by spaces. For some commands, *Maggie* even includes usage instructions:

Usage instructions for SqlScan command

Maggie as a network bridge head

Maggie contains functionality for simple TCP redirection, allowing it to function as a network bridge head from the Internet to any IP address reachable by the infected MSSQL server.

When enabled, *Maggie* redirects any incoming connection (on any port the MSSQL server is listening on) to a previously set IP and port, if the source IP address matches a user-specified IP mask. The implementation enables port reuse, making the redirection transparent to authorized users, while any other connecting IP is able to use the server without any interference or knowledge of *Maggie*.

For this to work, `StartHook` instructs *Maggie* to install network API hooks for the following functions:

- accept
- AcceptEx
- WSAAccept
- setsockopt
- CreateIoCompletionPort

allowing *Maggie* to intercept connections before reaching the underlying services.

The redirection setup can be controlled using the `SetClientData` command with

```
SetClientData <allowed IP mask> <destination IP> <destination port>
```

in order to enable redirection for the given IP mask (can end with '*' wildcard) to the specified IP and port.

Once finished, an attacker can simply disable the IP redirection feature using `StopHook` again.

In addition, *Maggie* contains SOCKS5 proxy functionality for more complex network operations.

Debug messages for SOCKS5 functionality

The unknown Exploit commands

Maggie's command list includes four commands that suggest exploit usage:

```
Exploit AddUser Exploit Run Exploit Clone Exploit TS
```

It appears that the actual implementation of all four exploit commands depends on a DLL not included with Maggie directly. Instead, the caller provides a DLL name as well as an additional parameter when calling each function. We therefore assume the caller manually uploads the exploit DLL prior to issuing any exploit commands.

Maggie would then load the user-specified DLL, look for an export named either `StartPrinter` or `ProcessCommand` (depending on the exact command used) and pass the user-supplied argument.

We were not able to dig up any potential candidate DLLs *Maggie* might be referencing during our research so it's unclear what specific exploit may be utilized here.

SQL bruteforcing and the curious Maggie backdoor user

Maggie's command set also includes two commands that allow it to bruteforce logins to other MSSQL servers:

```
SqlScanWinSockScan
```

To start a bruteforce scan, the controller would have to specify a host, user and password list file previously uploaded to the infected server, as well as an optional thread count. *Maggie* then creates every combination of (host,user,pass) and attempts to log in via SQL using ODBC, or a reimplementation only using basic socket functions in the case of `WinSockScan`.

Successful logins are written to a hardcoded log file, which can be in one of two locations:

```
C:\ProgramData\success.dat<MAGGIE_LOCATION>\success.dat
```

Maggie then tries to determine if the bruteforced login has admin rights. In case it successfully bruteforced an admin user, *Maggie* proceeds with adding a hardcoded backdoor user.

Based on this finding, *DCSO CyTec* conducted a scan on publicly reachable MSSQL servers in order to determine how prevalent the identified backdoor user is.

Out of approximately 600,000 scanned servers worldwide, we identified 285 servers infected with Maggie's backdoor user, spread over 42 countries.

Heatmap of backdoor user by country
The distribution shows a clear focus on the Asia-Pacific region, with South Korea, India and Vietnam as top 3 followed by China and Taiwan in the fourth and fifth place. Other countries appear to be incidental.

Prevalence of backdoor user by country
A logical next step would be to see if and how the affected servers are being utilized, which however goes beyond the scope of our analysis.

IoCs

As usual, you can find below IoCs in the form of a MISP event on our GitHub.

```
f29a311d62c54bbb01f675db9864f4ab0b3483e6cfdd15a745d4943029dcdf14a375ae44c8ecb158895356
  Mozilla/4.0
(compatible)C:\ProgramData\Success.datSuccess.datFailure.datAccessControl.Dat
```

MITRE ATT&CK

```
      Brute Force       Connection Proxy
```