

[NT] Unchecked Buffer in MDAC Function Could Enable SQL Server Compromise

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-08/0004.html>

From: support@securiteam.com

Date: 08/01/02

From: support@securiteam.com

To: list@securiteam.com

Date: Thu, 1 Aug 2002 12:10:57 +0200 (CEST)

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> -- Know that you're safe.

Unchecked Buffer in MDAC Function Could Enable SQL Server Compromise

SUMMARY

The Microsoft Data Access Components (MDAC) provides a number of supporting technologies for accessing and using databases. Included among these functions is the underlying support for the T-SQL OpenRowSet command. A security vulnerability results because the MDAC functions underlying OpenRowSet contain an unchecked buffer.

An attacker who submitted a database query containing an especially malformed parameter within a call to OpenRowSet could overrun the buffer, for the purpose of either causing the SQL Server to fail or causing the SQL Server service to take actions dictated by the attacker.

DETAILS

Affected Software:

- * Microsoft Data Access Components 2.5
- * Microsoft Data Access Components 2.6
- * Microsoft Data Access Components 2.7

Mitigating factors:

- * In order to exploit the vulnerability, the attacker would need the

Securiteam: [NT] Unchecked Buffer in MDAC Function Could Enable SQL Server Compromise

ability to load and execute a database query on the server. This is strongly discouraged by best practices, and servers that have been configured to prevent this (e.g., with the DisallowAdhocAccess registry setting, as discussed in the FAQ) would not be at risk from the vulnerability.

* Under default conditions, the system-level privileges gained through a successful attack would be those of a Domain User.

* Even though MDAC ships as part of all versions of Windows, the vulnerability can only be exploited on SQL Servers. Customers who are not using SQL Server do not need to take action, despite the fact that MDAC may be installed on their systems.

Patch availability:

Download locations for this patch

* MDAC 2.5:

<http://www.microsoft.com/Downloads/Release.asp?ReleaseID=41076>
<http://www.microsoft.com/Downloads/Release.asp?ReleaseID=41076>

* MDAC 2.6:

<http://www.microsoft.com/Downloads/Release.asp?ReleaseID=41077>
<http://www.microsoft.com/Downloads/Release.asp?ReleaseID=41077>

* MDAC 2.7:

<http://www.microsoft.com/Downloads/Release.asp?ReleaseID=41072>
<http://www.microsoft.com/Downloads/Release.asp?ReleaseID=41072>

What's the scope of this vulnerability?

This is a buffer-overflow vulnerability. An attacker who successfully exploited it would be able to take action with all the privileges of an affected SQL Server. At a minimum, this would grant the attacker complete control over the database, and potentially could grant administrative privileges at the operating system level as well.

Although the technology involved in the vulnerability does ship as part of Windows and other products, the vulnerability only poses a risk to SQL Servers – no other systems require the patch. Even in the case of SQL Server, the vulnerability could only be exploited by an attacker who had the ability to submit and execute database queries against an affected server. Best practices strongly recommend against ever allowing untrusted users to do this.

What causes the vulnerability?

The vulnerability results because a function in the Microsoft Data Access Components that provides some of the underlying functionality for the Transact-SQL OpenRowSet command contains an unchecked buffer. If a query called OpenRowSet using an especially malformed parameter, it could be possible to overrun the buffer in the underlying function.

What is Microsoft Data Access Components?

Microsoft Data Access Components (MDAC) is a collection of components used to provide database connectivity on Windows platforms. The components provide the underlying functionality for a number of database operations. (A good discussion of MDAC and the components it provides is available on

MSDN).

One point that is especially important to understand for the purposes of this vulnerability is the fact that MDAC is a collective name for a number of technologies, some of which are used by database clients and others of which are used by database servers. In this case, the component containing the flaw is one that is used only by SQL Server, and even then can only be exploited with a single Transact-SQL command, called OpenRowSet.

What is the Transact-SQL OpenRowSet command?

Transact-SQL (also known as T-SQL) is the language that Microsoft SQL Server uses to query and manipulate the database information. This allows a broad variety of applications, from web-based applications to ones based on C++, Java or other language to interrogate SQL Server databases. Among the commands available in T-SQL is OpenRowSet, which allows a program to connect to a selected data source and potentially execute a query against it.

What's wrong with the OpenRowSet command?

There is nothing wrong with the OpenRowSet command per se. However, the underlying technology that MDAC provides to support the command has an unchecked buffer. If an application called OpenRowSet and provided an extremely long value for a particular one of the parameters, it could overrun the buffer.

What could an attacker do via the vulnerability?

It would depend on the specific way the attacker overran the buffer. If the attacker provided input data that overran the buffer with random data, it would cause the attacker's connection to the SQL Server to be dropped; this would not pose a security risk to the server. On the other hand, if the attacker carefully selected the data, it would be possible to modify MDAC's functionality to perform any task the attacker specified.

What privileges would the attacker gain through the latter scenario?

The attacker would gain the ability to do anything MDAC could do. At a minimum, this would enable the attacker to take any desired action on the database, including adding, deleting, or modifying data. However, it would likely not provide similar privileges over the system at larger. Both SQL Server 7.0 and 2000 run by default with Domain User privileges rather than as part of the operating system.

Who could exploit the vulnerability?

In order to exploit the vulnerability, the attacker would need the ability to create and submit data queries. Best practices strongly recommend against ever allowing untrusted users to do this. For instance, in this case, the administrator would need to have granted untrusted users the ability to open and work directly with databases on the server. Clearly, it is unwise to do this, even in the absence of this vulnerability.

Is there a way to ensure that users can't submit OpenRowset queries?

Yes. It is often done by ensuring that users access the database only via

Securiteam: [NT] Unchecked Buffer in MDAC Function Could Enable SQL Server Compromise

a front-end application that limits what they can do. However, it is also possible to simply block user-submitted queries containing the OpenRowset command by setting the DisallowAdhocAccess option to a non-zero value using the advanced sp_serveroption command.

How do I know which version of the patch I need?

There's a patch for each supported version of MDAC. The following table shows which version of MDAC was supplied with various Microsoft products:

Version of MDAC – Shipped in...

MDAC 2.5 – Windows 2000, Office 2000 SR1 and later, SQL Server 7.0 Service Packs 2 and later

MDAC 2.6 – SQL Server 2000

MDAC 2.7 – Windows XP, Visual Studio .Net

An alternative way to determine the version of MDAC you are using is to consult the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\DataAccess registry key. The FullInstallVer key provides a value of the form x.xx.yyyy.y, where x.xx is the version number (e.g., if the FullInstallVer value were 2.70.7713.0, it would mean that MDAC 2.7 is installed on the system).

A final way to determine the version of MDAC is to right click on C:\Program Files\Common Files\System\ado\msado15.dll, select Properties, and then consult the Version information. The version information has the same format as that of the FullInstallVer value -- x.xx.yyyy.y, where x.xx is the version number.

I see that MDAC shipped in various versions of Windows, as well as Office. Does this mean that anyone using those products needs the patch?

No. MDAC ships with a number of products, but the function containing the vulnerability is only exposed on database servers. If you are not operating a database server, you do not need the patch, even if you are using one of the products listed above.

How does the patch eliminate the vulnerability?

The patch institutes proper buffer handling in the function associated with the vulnerability.

ADDITIONAL INFORMATION

The information has been provided by

<mailto:0_34580_E51E4D7D-DECD-43AE-9A29-36080E8D4C3C_US@Newsletters.Microsoft.com>
Microsoft Product Security.

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

-
- *Previous message:* support@securiteam.com: "[\[UNIX\] OpenSSH Trojaned \(Version 3.4p1\)](#)"
 - *Messages sorted by:* [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)