

[NT] SQL Server Installation Process May Leave Passwords on System

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-07/0038.html>

From: support@securiteam.com

Date: 07/11/02

From: support@securiteam.com

To: list@securiteam.com

Date: Thu, 11 Jul 2002 11:31:26 +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.

SQL Server Installation Process May Leave Passwords on System

SUMMARY

When installing SQL Server 7.0 (including MSDE 1.0), SQL Server 2000, or a service pack for SQL Server 7.0 or SQL Server 2000, the information provided for the install process is collected and stored in a setup file called setup.iss. The setup.iss file can then be used to automate the installation of additional SQL Server systems. SQL Server 2000 also includes the ability to record an unattended install to the setup.iss file without having to actually perform an installation. The administrator setting up the SQL Server can supply a password to the installation routine under the following circumstances:

- If the SQL Server is being set up in "Mixed Mode", a password for the SQL Server administrator (the "sa" account) must be supplied.
- Whether in Mixed Mode or Windows Authentication Mode, a User ID and password can optionally be supplied for the purpose of starting up SQL Server service accounts.

In either case, the password would be stored in the setup.iss file. Prior to SQL Server 7.0 Service Pack 4, the passwords were stored in clear text. For SQL Server 7.0 Service Pack 4 and SQL Server 2000 Service Packs 1 and 2, the passwords are encrypted and then stored. Additionally, a log file

Securiteam: [NT] SQL Server Installation Process May Leave Passwords on System

is created during the installation process that shows the results of the installation. The log file would also include any passwords that had been stored in the setup.iss file.

A security vulnerability results because of two factors:

- The files remain on the server after the installation is complete. Except for the setup.iss file created by SQL Server 2000, the files are in directories that can be accessed by anyone who can interactively log on to the system.
- The password information stored in the files is either in clear text (for SQL Server 7.0 prior to Service Pack 4) or encrypted using fairly weak protection. An attacker who recovered the files could subject them to a password cracking attack to learn the passwords, potentially compromising the "sa" password and/or a domain account password.

DETAILS

Vulnerable systems:

Microsoft SQL Server 7.0, Microsoft Data Engine 1.0 (MSDE 1.0), or SQL Server 2000

Mitigating factors:

- * The vulnerability could only be exploited by an attacker who had the ability to interactively log onto an affected system. However, best practices suggest that unprivileged users not be allowed to interactively log onto business-critical servers, including database servers.
- * The vulnerability with regard to the sa password only affects servers configured to use Mixed Mode. Customers using Windows Authentication Mode (which is the recommended mode) would only have credentials at risk if they had chosen to provide a domain credential to be used in starting the SQL Server services.
- * The passwords stored in the setup.iss and log files are those provided at installation time and are not kept up-to-date when password changes are made. As a result, if the administrator changed a password, the information in the setup.iss and log files would not allow any access.
- * In the case of SQL 2000, setup.iss is stored in a directory that only allows access by administrators and the user installing SQL Server.
- * If the setup.iss and log files containing domain user and/or sa passwords are deleted, the passwords could not be retrieved.

Patch availability:

Download locations for this patch

The KillPwd utility can be obtained at the following location:

- * Microsoft SQL 7, MSDE 1.0, and Microsoft SQL Server 2000:
<<http://www.microsoft.com/Downloads/Release.asp?ReleaseID=40205>>
<http://www.microsoft.com/Downloads/Release.asp?ReleaseID=40205>

What is the scope of the vulnerability?

This is a privilege elevation vulnerability. The SQL Server installation routines can, under certain conditions, store passwords that were provided by the administrator doing the setup. However, they are not stored

Securiteam: [NT] SQL Server Installation Process May Leave Passwords on System

securely, with the result that it could be possible for an attacker to access and compromise the passwords.

The passwords are only stored under two conditions: if SQL Server was configured in a mode that Microsoft recommends against using, or if the administrator chose a particular install-time option discussed below. Even in cases where one or more passwords were stored, the vulnerability could only be exploited by an attacker who could log onto an affected SQL Server interactively — that is, at the system keyboard. If an administrator had changed a password after installation, the stored password would no longer allow any access.

What causes the vulnerability?

The installation routines for SQL Server 7.0, SQL Server 2000, and MSDE 1.0 create several files as part of their operation. These files contain information recorded during the installation process, potentially including the SQL Server administrator password (if the Server is configured to use Mixed Mode) and/or a domain userid and password (if the administrator chooses to provide this information in order to allow SQL Server services to automatically start).

A security vulnerability results because of two factors: the files can be accessed by interactively logged-on users, and the information in them is insufficiently well protected. In some cases, the data is in plaintext; in others, it's encrypted, but only weakly. A user who accessed one or more of the files could potentially recover the passwords within them, thereby compromising the accounts.

What is MSDE, and how is it related to SQL Server?

Microsoft Data Engine (MSDE) is a database engine that's built and based on SQL Server technology, and which ships as part of several Microsoft products, including Microsoft Visual Studio and Microsoft Office Developer Edition. There is a direct connection between versions of MSDE and versions of SQL Server. MSDE 1.0 is based on SQL Server 7.0 technology; MSDE 2000 is based on SQL Server 2000.

The vulnerability here involves files that are created by the installation routines for various versions of SQL Server and MSDE – specifically, it involves SQL Server 7.0 and MSDE 1.0, and SQL Server 2000 (but, in a noteworthy exception, not MSDE 2000).

What are the installation files, and why are they created?

There are two types of files involved in this vulnerability, both of which are created when installing SQL Server 7.0, SQL Server 2000, or MSDE 1.0. (Both fresh installations and service pack installations create the files). The files are:

* An unattended installation file. This file, setup.iss, is created as part of the installation process for SQL Server 7.0, MSDE 1.0 or SQL Server 2000, and contains all of the information entered by the administrator during the installation process. Setup.iss is created in

Securiteam: [NT] SQL Server Installation Process May Leave Passwords on System

order to allow unattended installs; having created setup.iss once, an administrator can use it to automate additional identical installations on other servers.

* Log files. These files, named sqlstp.log when SQL Server 7.0, MSDE 1.0 or SQL Server 2000 is initially installed, and sqlspX.log when a service pack is installed (where X is the service pack number), contain data logged by the installation process as it progresses. The purpose of the log files is to allow administrators to confirm successful installations and troubleshoot unsuccessful ones.

What is wrong with these files?

The files have two problems. First, they are created with inappropriate permissions that would allow anyone who could log onto the server interactively to read them. (The sole exception is the SQL 2000 setup.iss file, which is created with the correct permissions). Second, the information within them isn't adequately protected. In some cases, the information in them is unencrypted; in others, it's encrypted, but the encryption used offers only weak protection.

Under what conditions is the password data in the files encrypted, and when is it left in clear text?

The passwords in the unattended installation file are created in clear text by versions of SQL Server 7.0 prior to Service Pack 4. All versions of SQL Server 2000 and all versions of SQL Server 7.0 beginning with Service Pack 4 encrypt the passwords before storing them. The log file for an installation will have the same clear text or encrypted passwords as are found in the unattended installation file.

Why does it matter whether someone could read the files? What data is in them?

In general, the data in these files is not sensitive. However, there are two noteworthy exceptions:

* SQL Server administrator password. During installation, the administrator must choose between two operating modes, known as Mixed Mode and Windows Authentication Mode. If Mixed Mode is selected, the password for the administrator account (the so-called "sa" account) is recorded in the unattended installation file.

* Domain user credentials. Another install-time option enables the administrator to configure various SQL-related services to run automatically in the security context of a domain user account, by providing the account's userid and password. If this option has been selected, the userid and password are recorded in both the unattended installation file and the log file.

What could an attacker do via the vulnerability?

The risk posed by this scenario is straightforward. An attacker who gained access to the files could compromise any passwords stored within them, and potentially use them to gain control of either the SQL Server or the domain account.

Securiteam: [NT] SQL Server Installation Process May Leave Passwords on System

Who could exploit the vulnerability?

The vulnerability could only be exploited by an attacker who had the ability to log onto an affected server interactively – that is, at the system keyboard. (As discussed above, the SQL 2000 unattended installation file is stored in a folder that can be accessed only by administrators, so in this case exploiting the vulnerability would require the attacker to already have administrative privileges).

How can I tell if my server is at risk?

A server would only be at risk if it was configured to operate in Mixed Mode or if the administrator had chosen the installation–time option to automatically start SQL services using a domain account. If the server was configured to operate in Windows Authentication Mode (which is the recommended mode) and the administrator had not chosen to automatically start the services, the server would not be at risk.

Suppose the passwords had been changed after installation. Would the server be at risk?

The files contain snapshots of the passwords at installation time, and are never updated. If the passwords were changed after installation, it would not benefit the attacker to compromise the data in the files.

If an attacker did compromise the passwords, would he gain complete control over the server?

Compromising the password for the "sa" account would give the attacker complete control over SQL Server, but would not convey administrative privileges on the system itself. Nor would it provide access to any other servers in the domain.

Compromising the domain account would grant the attacker all the privileges that the account possessed; the specific ones would depend on how the account had been configured. Best practices always recommend that users be provided the fewest privileges necessary.

Why is not MSDE 2000 affected by the vulnerabilities?

SQL Server 7.0, MSDE 1.0, and SQL Server 2000 all use the same installer technology while MSDE 2000 uses a different installer technology. MSDE 2000 does not create the setup.iss and log files during installation and so is not affected by this vulnerability.

Could this vulnerability be exploited remotely?

No. An attacker would need to log on to the SQL Server machine and be able to access the directories where the setup and log files are kept.

What can I do to eliminate this vulnerability?

Microsoft recommends that customers running affected systems take either of the three following steps:

- * If the unattended installation file and log files are not needed, delete them.
- * If the files must be retained, move them to a folder that is only accessible by administrators or, better yet, save them to well–protected

Securiteam: [NT] SQL Server Installation Process May Leave Passwords on System

offline storage.

* Use the KillPwd utility provided below to remove passwords from the setup.iss and log files.

If I want to delete or move the files, where can I find them?

The unattended installation file is named setup.iss, and is stored in the following locations by default:

* SQL Server 7.0 and MSDE 1.0: The file is stored in the %windir% directory (e.g. "C:\Winnt" by default on Windows 2000).

* SQL Server 2000: The file is stored in the "install" subdirectory associated with the SQL Server installation (e.g. "C:\Program Files\Microsoft SQL Server\mssql\install" by default).

The log file created by Gold installations is named sqlstp.log, and the one created by service packs is named sqlspX.log (where X is the service pack number). The files are stored in the following locations by default:

* SQL Server 7.0 and MSDE 1.0: The files are stored in the %windir%\temp directory (e.g. "C:\Winnt\temp" by default on Windows 2000).

* SQL Server 2000: The files are stored in the %windir% directory (e.g. "C:\Winnt" by default on Windows 2000).

What is the KillPwd utility?

The KillPwd utility provided below is an updated version of the tool first described in Microsoft Security Bulletin MS00-035. This utility searches the Microsoft SQL Server log and setup files for passwords and deletes any passwords that are found, whether encrypted or not. It does not by default, delete passwords in the setup.iss file created by SQL Server 2000 installations. This is because the setup.iss file created by SQL 2000 installations is saved in a directory that only allows access by administrators and the user setting up SQL Server 2000.

If I am not sure if a system is affected, can I run the KillPwd utility anyway?

Yes, the KillPwd utility removes any passwords in user accessible directories that may remain in the setup.iss and log files after a SQL Server installation. There is no problem in running the utility even if no passwords exist.

ADDITIONAL INFORMATION

The information has been provided by

<mailto:0_33449_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

Securiteam: [NT] SQL Server Installation Process May Leave Passwords on System

=====
=====

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: "[\[UNIX\] Multiple Vulnerabilities in ToolTalk Database Server](#)"
- **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)