the Availability Digest

www.availabilitydigest.com

Oracle’s Ticking Time Bomb

February 2012

A potentially catastrophic bug that has been around for years has been discovered in the Oracle database. Left unfixed, the bug could crash all of the interconnected databases in a large enterprise, though no data would be lost. Recovery would take days or even weeks. Such a disaster might occur due to normal operation, or it could be exploited by a malicious attacker.

It was journalists at *InfoWorld* who discovered the flaw and who notified Oracle in November, 2011. Oracle asked *InfoWorld* not to report the flaw until a patch could be issued in order to avoid exploitation of the flaw against unsuspecting users. The patch was issued on January 17, 2012, and *InfoWorld* published its exposé later the same day.[1] In this article, we summarize the *InfoWorld* report.

The System Change Number

The problem stems from a mechanism deep within the Oracle database – one with which Oracle DBAs seldom deal. It is the System Change Number (SCN).

Oracle’s SCN Clock

The SCN is essentially the Oracle timekeeper. It is a number that is incremented and appended to every committed transaction. Like any clock, the SCN can never move backwards – it must always tick forwards. To ensure that the SCN will never be exhausted, it is configured as a very large number – a 48-bit field, which supports a maximum value of $2.8x10^{14}$ ticks.

The SCN is the key to maintaining data consistency. Every block in the Oracle Redo log contains the beginning and ending SCN of transactions in that block.

The SCN serves many purposes. For one, it is used by Oracle to ensure that query results are consistent. If the data to be returned by a query is being actively updated, Oracle ensures that all data items returned in response to the query have an SCN that is equal to or less than the SCN value at the time of the query.

Another use of the SCN is to guarantee that replicated transactions are applied to the target database in the correct order to guarantee integrity of the target database. The SCN also can be used to restore a database to a specified point in time. With Oracle 10g, for instance, the SCN can be mapped to a specific time within three seconds.

[1] Fundamental Oracle Flaw Revealed, *InfoWorld*, January 17, 2012

The SCN Soft Limit

To ensure that a database is managing its SCN properly, a soft SCN limit is maintained. It is calculated under the assumption that the database has been committing 16,384 (2^{14}) transactions per second since January 1, 1988. This is intended to be an unrealistically high transaction rate in order that the SCN of a database should never even approach the soft limit. At 16K transactions per second, it will take almost five and one-half centuries (544 years, to be more exact) to exhaust the SCN – long past the probable life of the Oracle product.

Should the SCN of a database exceed the soft limit, it is an indication of a database malfunction. The database is considered unstable and unusable.

Linked Databases

Oracle databases can be linked across the enterprise. If two or more databases connect to each other, they synchronize their SCNs to maintain data consistency. Since the SCN can never tick backwards, all databases set their SCN to that of the database with the highest SCN. Therefore, a database's SCN may jump forward when it links to another database.

If a database's SCN exceeds the SCN soft limit, other databases cannot connect to it since it is considered unreliable.

The Hot-Backup Bug

On the surface, Oracle's SCN method to maintain data consistency seems solid. In fact it has worked well for decades. But *InfoWorld* discovered a potentially damaging flaw.

The problem is not with SCN per se but is rather a coding flaw in Oracle's hot-backup utility. Oracle allows a database administrator to make a hot backup of an operational database by entering a simple command. Though there are several ways for a DBA to initiate a hot backup, one is via the command 'ALTER DATABASE BEGIN BACKUP.' Unfortunately, a bug in this method of invoking a hot backup causes the database's SCN to increase dramatically – perhaps by millions or even billions of ticks in a short time.

A billion (10^9) is small compared to the SCN range of about 2.8×10^{14} – it would take 280,000 billion ticks to exhaust the SCN. However, it is not the SCN range that is of concern – it is the SCN soft limit. If it takes 544 years to exhaust the SCN, and it is now 2012, the soft limit has been escalating for 24 years and has reached about 4.4% of the SCN limit. It will not take 280,000 billion ticks to reach the soft limit – it will take about 12,000 billion ticks. If the average escalation of the SCN due to hot backups is one billion ticks (this number is not really known), the system is within 12,000 hot backups of reaching the soft SCN limit.

Herein lies the problem. Each time a database administrator performs a hot backup on his database, the database's SCN increases dramatically. When he then connects his database to other databases, their SCNs are reset to the new, expanded SCN and increase, say, by a billion. The elevated SCN flows from one database instance to another like a virus. There is no going back.

Some companies have hundreds of database servers running hundreds of instances of Oracle. In such a large Oracle installation, if many of the databases are backed up online with the problematic command, it will not take too many years for all of the databases to reach the soft limit. Based on standard database administration procedures, a DBA is unlikely to ever see this happening.

What results when the soft SCN limit is reached? None of the databases can interconnect, the applications cannot run, and the entire IT infrastructure of the enterprise crashes!

A Security Flaw

This catastrophic situation can be reached through normal operation of the company's data centers. However, the SCN flaw raises another concern. A bad actor could easily cause a system-wide crash of interconnected Oracle databases. All he would need is access to a low-priority database that can link to a high-priority database. He could write a script that performed multiple hot backups on the low-priority database and that periodically connected the low-priority database to the high-priority database. The SCN of the high-priority database would be escalated by perhaps trillions of ticks and would propagate this expanded SCN to other databases to which it is linked. A few cycles of this malicious activity could crash large segments of a company's IT infrastructure.

Recovery from a Soft SCN Limit Violation

Fortunately, there have been no reported instances yet of a soft SCN limit violation taking down a company's IT operations. But if it should happen, how can the IT systems be recovered?

One method would be to shut down all IT operations for several weeks to let the soft limit move forward. However, it is unlikely that any company could afford to do that. In any event, that is just putting off the problem.

A database could not simply be reloaded, since the current SCN for the database would be used as the base for the replayed transactions. Rather, the database would have to be exported, dropped, and then imported. For today's petabyte databases, this could take days if not weeks, during which time much of the company's infrastructure would be unusable.

Oracle came out quickly with a patch that lets the DBA specify the soft SCN limit increment. The increment defaults to 32K ticks per second, which doubles the soft limit. A DBA can increase this parameter even further. This certainly gains some time, but the problem remains. Furthermore, if a patched database links to an unpatched database, the connection will fail along with the application. To make matters worse, the patch is available only for the most recent Oracle versions.

This Oracle flaw probably will be experienced only by the largest Oracle installations. However, it is these installations that will take the longest time to recover and that will suffer the most consequences financially as well as with respect to customer retention, regulatory issues, and adverse press.

The Oracle Fix

On November 17th, Oracle issued its first Critical Patch Update (CPU) of 2012. This CPU addressed 78 security issues across its product line.

Only two patches within this CPU related to Oracle. The patch of interest to this article (CVE-2012-0082) removes the methods that arbitrarily increase the SCN. These methods include not only the hot-backup bug described above but also other similar bugs that were found. The patch also includes an "inoculation" protection to prevent a database from connecting with another database with an unreasonably high SCN, though that SCN may be within the soft SCN limit.

This patch is available only for certain Oracle 10g and 11g database versions.[2] Oracle administrators must ensure that no unpatched databases are allowed to connect to a patched database.

[2] Oracle 11g 11.1.0.7, 11.2.0.2, and 11.2.0.3 as well as Oracle 10g 10.1.0.5, 10.2.0.3, 10.2.0.4, and 10.2.0.5.

Summary

Oracle's SCN flaw is a ticking time bomb that is set to go off in 544 years. Occasionally, however, it ticks out of control and may explode quite prematurely.

The SCN flaw has a low likelihood of impacting most companies except those that are running hundreds of Oracle instances. However, if it does happen, the results are catastrophic. A company's data centers might be down for weeks.

It is therefore imperative that companies immediately install the patch that Oracle has provided to fix this problem. Unfortunately, older versions of Oracle cannot be patched and will continue to exhibit the SCN flaw. DBAs must make sure that patched databases do not link to unpatched databases. If older, unpatched database instances are to be included in linked configurations, they should be upgraded to a patchable version.

Acknowledgments

Our thanks to our subscriber Bruce Holenstein of Gravic, Inc., for bringing this issue to our attention.

In addition to the *InfoWorld* article previously referenced, information for this article was taken from the following sources:

Ask Tom (asktom.oracle.com)
Oracle Patches 78 Security Flaws, *eSecurity Planet*; January 18, 2012.
Oracle Accused Of Downplaying Severity Of Database Security Flaws, *Tech Week Europe*; January 19, 2012.
en.allexperts.com