the Availability Digest

www.availabilitydigest.com

Why Back Up?

April 2009

The blogging platform JournalSpace suddenly went out of business when it lost its database that was not backed up and could not be recovered. Thousands of bloggers lost years of their work. How could this have happened?

JournalSpace

Started in 2003, JournalSpace was a popular and growing blog-hosting service. It was primarily a free service supported by advertising, but it offered an upgraded “professional service” for a fee.

JournalSpace is perhaps best known as the site of the “Queen of Sky” blog by Ellen Simonetti, who was fired in 2004 from her job as an airline flight attendant by Delta Airlines.[1] Delta deemed some of her blog content, including photographs, to be inappropriate. The national coverage of this event raised an international outcry against “employer blog backlash.”

Ellen’s continuing blog was wiped out along with thousands of others on December 18, 2008, when JournalSpace lost its entire database and was unable to recover. This was the second time in its history that it had lost its database, the first time being a warning that evidently went unheeded.

Three weeks later, the JournalSpace domain was purchased by a third party who resurrected the site but not the data. Six years of memories were erased for many of the site’s bloggers.

What Happened to JournalSpace’s Database?

Apparently, the database’s demise was the malicious act of a disgruntled employee – even worse, the IT manager. JournalSpace claims that it had caught the IT manager stealing from the company. They summarily fired him, but he did a slash-and-burn on his way out, overwriting the entire database with garbage.

This should have been only a minor irritant because all that was needed to cure the problem was to restore the database from the backup copy. The problem? No backup copy!

It was, of course, the IT manager’s responsibility to ensure that a backup copy was periodically taken and preserved. However, though he dutifully backed up the HTML code for the site on a remote server, his backup strategy for the blog database was to use a RAID 2 mirrored disk. If one disk failed, the database was still available on the mirror.

[1] Wikipedia: Ellen Simonetti

Unfortunately, upper management should have known that this was not a backup strategy at all. True, it protected against a hard-disk failure. But it did not protect against a site disaster – or a malicious act.

JournalSpace's management described the incident as follows:

> "It was the guy handling the IT (and, yes, the same guy who I caught stealing from the company, and who did a slash-and-burn on some servers on his way out) who made the choice to rely on RAID as the only backup mechanism for the SQL server. He had set up automated backups for the HTTP server which contains the PHP code, but, inscrutably, had no backup system in place for the SQL data. The ironic thing here is that one of his hobbies was telling everybody how smart he was."

Sounds like he may have been smarter than the management.

The Attempt At Recovery

In a panic, the JournalSpace management sent the hard disks to DriveSavers, a service known for recovering data from burnt, drowned, and crushed hard drives. On Saturday, the day after the disaster, JournalSpace posted the following message on its blog:

> "What happened is that both disk drives which hold the databases have failed. On Monday, we'll be sending then to DriveSavers for recovery. Because of postal transit times and the holidays, JournalSpace will likely be down for most or all of Christmas week. We're very sorry for this inconvenience."

On Sunday, it said:

> "The drives will be on their way to DriveSavers in the morning. Its estimated charge for a full data recovery is roughly equal to the amount of money JournalSpace made in the past year."

Later, it added:

> This is day three of DriveSavers' 5 to 7 day turnaround time. Some progress has been made, but they cannot give a "yes" or "no" answer on full recovery.

Unfortunately, the answer was ultimately "no." The disks were unrecoverable. They had been overwritten with random data, obliterating the original data.

JournalSpace closed its doors.

The Aftermath

Other bloggers entered the fray with criticism, support, and ideas. One blogger (Andrew Heart) found that he could recover many of the blog pages from Google's cache. He suggested doing a Google search on <username>.journalspace.com, which recovered many pages. However, any pages that the owner had marked as "private" were not cached by Google and were lost forever.

On January 11, 2009, the JournalSpace domain name was sold; and the new owners resurrected the site but not the data. They promised to back up their data every night to an offsite backup facility and to take local backups every hour. In this way, bloggers should never lose more than an hour's worth of data.

The original JournalSpace operators are now trying to make their blogging platform open source under a new name.

Lessons Learned

The obvious lesson from this catastrophe is simple – back up!

But it’s not quite as simple as that. Backup should occur at every level. We should know to do this with personal data on our PC (whether we back up or not). We often believe that our data will be protected when we use online hosting services for our blogs, our online retail stores, or SaaS (software as a service) offerings. But many bad experiences tell us otherwise.

At the highest level, the hosting service must periodically back up its database. The backup should be to a remote site to protect against a site disaster such as a tornado, earthquake, fire, or flood. The hosting service should disclose its backup policy and should include it in a service-level agreement. It should also disclose the anticipated recovery time following a database loss.

Furthermore, the backup storage should be durable and preclude deletion or modification. This protects against malfeasance such as occurred with JournalSpace. Magnetic tape satisfies this latest criterion but has its own problems. Database recovery from tape can take days if the database is large, and what happens if a tape proves to be unreadable? Online storage services such as Carbonite (www.carbonite.com) offer another solution.

But the most exhaustive backup strategies can still fail. Therefore, it is incumbent on the user, whether he be a blogger, a retailer, or a business using the new technology of cloud computing, to ensure that critical data is backed up independently of the online service. For a blogger, this may simply mean that he periodically backs up his own PC files on a removable medium or on an online backup site. For applications in which the data is created by the hosting service, such as a retail store, the user should determine the facilities provided by the hosting service that will allow that data to be independently backed up by the user.

Our procedures here at the Availability Digest attest to that practice. Our hosting service gathers all of our subscriber information. It is placed in a file on its servers (which we hope are backed up), but this file is available for us to download. We periodically download a copy of the current subscriber list and save it on our local server. In the background, we use Carbonite to back up the HTML for our web site and to back up our subscriber list. Carbonite, our hosting service, and our server will all have to be destroyed in order for us to lose our data. Though this is always possible, we feel confident in the longevity of our database. Do you?

References

Information for this article was taken from the January, 2009, issues of Techcrunch, Word Press, geek.com, Quick Online Tips, and the JournalSpace blog.