Global Information Assurance Certification Paper

LONG TERM STORAGE AND SECURITY OF COMPUTER FILES.

Mark Shepherd
GIAC 1.4 (Option # 2)

ABSTRACT

This report will show how we determined the problems of our previous long-term file storage system, the steps we went through to solve each of the problems, and how the system is now easier to use, more secure, and more efficient in the storage of files.

Anyone who needs to maintain long-term archive facilities or version tracking stores might find the information here of use in creating their own systems.

DEFINING THE PROBLEM

I work for the state government in a regulatory agency with approximately one hundred computer users. One of our division's functions is to handle the permitting process for construction within floodplain areas around streams, rivers, and lakes. Part of this process is to run computer models to determine what will happen with the water body if the proposed construction were completed. We have a system in place to handle the daily backup of our files for this process. However, we have experienced a number of problems with the long-term storage of these files.

At the end of last year our department went through reorganization. We decided that was a good time to correct some of the problems with the archival system. Because the division is regulatory in nature, the paperwork generated in the permit process has to be kept for a long period of time as public record. It was decided in the reorganization that we needed to maintain the electronic files as well as the paper files as public record. Therefore, the reorganization became the motivation and opportunity to create a better long-term storage method.

Our computer systems, over time, had changed. We began with a mainframe system, progressed to a few PC's for each group of users, and then to a computer at each person's desk. Backup hardware had also changed over time, so our procedures and the media for storing the data had changed as well.

Storage in nonstandard locations

Our computer systems have more or less evolved over time, but no standard procedure for storing files has evolved along with the computers. Added to that is the fact that we have a large turnover in personnel over time and each person has a different idea of the best way of doing things.

For example, we have found that the computer files worked on by some intermittent personnel were only filed as paper copies of the output. The computer model stayed on the local drive of the computer they were using. Other times some of the intermittent or newer personnel would use the software's default locations when creating or saving files. This would result in files being stored on the local drive in locations that we didn't backup or copy.

As we received new computers and distributed them, these files would be lost because the non-standard locations that the files were stored would not be moved to the person's new computer or saved in any standard location.

Variety of storage media types and lifetime of each media

One of the problems of an evolving computer system is continually running out of drive space to store files. There have been incidents where, during a few of the office-wide mandates to clean up the fileservers, some required files were deleted and the loss went un-noticed for about a year (or more). When we attempted to restore the missing files from the old backup tapes we discovered that the updated software for making the backups had changed enough that we could not read the older tapes.

In one restore attempt, we discovered that the tape could not be read at all even though we had the correct hardware and software. Some of the oldest stored files are still on reel-to-reel tape from the old mainframe systems. I'm not sure that anyone within our department even has equipment to read them any more.

Retrieval of stored files

In our previous method of handling files, there were some files that could be reused and stored in other locations under the same file name. However, most of the files that we were concerned about were engineering files and those files were not within this category.

Some construction permits would be revisited over time. The associated files would have to be modified and run through the programs again to find answers to further questions about the engineering decisions made for the permits.

At times there may have been doubts that the correct file was originally archived, or the original file that created the results could not be located. In

these cases, a duplicate file would have to be created from scratch using the paper copy. This has added unnecessary hours to the engineer's task of evaluating permits.

In one instance there was a question as to whether a paper copy output that was stored for a permit was the correct version. To determine this, the original computer model had to be restored from backup tape. We had to restore the set of files from about eight backup tapes to find the correct set. The process was complicated by the fact that another permit had used some of the same files.

A summary of the problem

Staff discussions of the above issues resulted in the following list of problems to address.

1. No set rules for file archiving
2. Too short of life of archiving media
3. No consistency of media type
4. No method of determining when files were changed or of proving they hadn't been changed.

FINDING POSSIBLE SOLUTIONS

1. No set rules for file archiving

Our main concerns were in being able to limit access to the files from outside personnel against any altering or accidental deletion, and how to maintain a coordinated location for the users to keep all the appropriate files together. We knew that any file structure had to be logical according to the structure of our organization. We also had to keep the file system as simple as possible so it would be adhered to.

Part of the decision on how to handle the files was taken care of by other decision makers in the reorganization process. During the reorganization they divided the state into twenty-eight separate drainage basins based on the main streams and rivers in the state. These were distributed between the three engineering sections, North, Central, and South.

Each team is responsible for any engineering work related to any permits done in their basins. Previously, anyone in the engineering section could be assigned to any permit work. It was decided to break the areas down this way so there would be people experienced with the separate areas instead of everyone having to have a general idea of the whole state.

Because of this new way the sections were laid out, we created a file space on the main fileserver that matched the format of the basin teams. This allows us to restrict the permissions on the volume to a smaller number of people who would need access to any piece of information, thus improving our security.

There is now one volume on our Novell file server that has twenty-eight subdirectories, one for each basin. Within each basin subdirectory there is a

subdirectory that is dedicated to each permit. Each subdirectory on this level is named for the number of the permit that affects a site within the basin area.

Any files that are copied from a previous permit for use in a new permit are to be placed within the new permit's subdirectories. Any temporary files for the permit review can be kept on the local drive of the computer being used by the person working on the project.

This gave us a system people would use now. We also needed to ensure that the procedures would be followed by future personnel as well. To help insure consistency, these archival instructions were added to our Standard Operating Procedure manual, written as part of our reorganization.

Business practice changes gave us a logical way to lay out the file structure. These procedures for storing all the relevant files in one location that can be easily secured, backed up, and restored solved our problems resulting from inconsistent archiving practices.

Creating the new volume on the fileserver allowed us to set up access permissions on the volume. This allows for a tighter control over who has access to the volume and each set of subdirectories thus aiding in the control of whether a file can be changed.

2. Too short of life of archiving media

When we have tried to restore data from some of our oldest files, we discovered that some of the tapes could not be restored. We discovered that two of our biggest problems were that the media didn't last long enough, and that the software and hardware used to restore the data changed over time.

Our storage methods ranged from keeping backup tapes in boxes under desks in the office, to keeping them in a lock box in a warehouse. After some research, we found out that for files to be successfully stored on magnetic tape for a long period of time, we needed to provide a storage location with specific conditions. Unfortunately we didn't know this when we started keeping the tapes.

We should have been keeping the tapes in a temperature and humidity controlled environment. For example, from Vidipax.com: (http://www.vidipax.com/)[1] For magnetic tape-storage of up to at least 10 years, attain a maximum temperature of 73 degrees F and a relative humidity of between 20 percent to 50 percent. The area we keep them in has no such humidity controls, and the temperature can vary more than the recommended range.
The Imation website (<http://searchstorage.techtarget.com/tip/ 1,289483,sid5_gci832122,00.html>)[2] pointed out another problem with our long-term storage

of tapes. The number of passes determines the life of the tape also. We previously did not have an established rule for tracking the number of uses of a tape.

In one case where we needed to restore some files, we discovered that we didn't have any tapes that still had that information. The personnel that had been responsible for saving that specific data had reused the stored backup tapes after five years. We had to request the data from the company that had originally provided it.

In some discussions with our personnel we found that the life of our data wasn't measured in just a few years. Some of our data (even the intermediate steps of some processes) has needed to be stored in our archives well over twenty years.

Because of their short life, we determined that backup tapes are not sufficient for our long-term storage requirements. Magnetic media of any sort that we could locate had only a ten to fifteen year storage life even when properly stored. Clearly, something else was needed.

Our research about other media that might more closely fit our needs pointed out the advantages of an older media that we had not previously considered. That media is the Compact Disc. Originally the Compact Disc had a lifespan of about seventy-five years. Recent advances have improved that to around one hundred years.

A direct quote from the Kodak website: (<http://www.kodak.com/global/en/professional/products/storage/pcd/techInfo/permanence.jhtml>)[3] "We predict the lifetime of KODAK Photo CD, and KODAK Writable CD Media with InfoGuard Protection System, under normal storage conditions in an office or home environment, should be 100 years or more."

Another advantage of the CD-R media is that as long as we use the ISO-9660 standard, we should have no problem ever reading the media because all CD software has been written for backward compatibility. Our experience with tapes is that the software for it is not written for backward compatibility.

When we make a tape backup of the entire system, our software grabs all of the data on all of our file servers. Presently this takes three 'four gigabyte' tapes, or something over eight GB of data but less than twelve GB. Twelve GB of data would take nineteen CDs to store. We could use CDs to perform this backup and still be ahead because of the cost difference between three 4mmDAT tapes and nineteen CDs that would be stored every month. Fifty CDs presently cost the same as what we are spending on one 4mmDAT tape. Any other storage system that we have seen within our price range is based on some kind of tape. Therefore, other storage systems would still leave us with the cost problem and the limited lifetime of the tape.

Therefore using the CDs allows us to store a set of backups in the office, and another set offsite. With reasonable care, the disks will be usable for as long as we anticipate needing them. The total cost of storing data on CD is lower than the cost of using the 4mmDAT tapes to store the same information. We have not found any other storage solution that matches the lifetime and cost advantages of CDs for our situation.

3. No consistency of media type

Some of the oldest backup media that we had used for long-term storage at this point is actually a nine-track reel-to-reel tape. This is a holdover from when the division used remote terminals to a central mainframe computer. We have long ago moved to other systems, and I doubt that we even have the capability of restoring that data from backup. If it is still possible to restore those files it would be very expensive and I doubt that the facilities will be available much longer.

Most of our present archived media is the four millimeter DAT tape format. We recently were using five tapes for a full backup of our entire systems.

Some of our backups over time have even been made on an older format, a 250 MB tape. We still have an older computer with the proper tape drive installed in it on the shelf for any possible restore tasks from that format tape.

All of these media have been used within the last ten years. In other words, we have changed the software and or hardware for making backups to our systems three times in that time span. This means we need to keep at least four different software packages and three different types of hardware available to restore files from most of our existing archive media. Two of these types of hardware are no longer available as new equipment. We need to find a media to transfer the information to from these older media before it is no longer possible to do so. We also need to use a media format that will be readable despite advances in the technology.
Based on the track record of CDs as compared to that of tapes, the CD will continue to be readable by computer systems in general regardless of the kinds of improvements in the technology. With the availability of the CD burner, we are easily able to transfer data from the older formats to CDs before we loose the capability of reading it.

4. No method of determining when files were changed, or of proving they hadn't been changed

A repeated problem with having no set procedures for copying and storing records is that if a file has been changed, we have no way of tracking when it was changed. This is an issue because we need to be able to restore files pertaining to specific permits to check if the changes might affect other work being done on the streams.

Our only method before was to restore the files from tapes on or about the time the original permit was finished and have the engineers run the program to see if the results match. If they don't, then we need to restore files before or after that date. This has proven to require more time than we can afford to give.

To address this issue, we need to keep track of what files were changed, and to store each version of the file. In most of the articles we have read on handling files and checking to see if they have been changed, there are two main methods of integrity checking mentioned, the CRC and the MD5 Hashes.

Each of the methods is available in a variety of command line programs or within programming languages. To determine if one or the other would be beneficial, the following criteria must be met. The method must be able to indicate the slightest change in a file, be simple to use, and easily integrated into the storage system.

The documentation that we have found shows that the CRC-32 function has an output of a hexadecimal string of thirty-two characters. While the MD5 output is also a hexadecimal string of thirty-two characters it makes use of a different type of calculation.

In an article for "Dr. Dobbs Journal" by Mark Nelson http://marknelson.us/articles/crcman/crcman.htm)[4] it is pointed out that, "while difficult, it is possible to add information to the back of a file and recreate the CRC checksum that will be the duplicate to the original."

In the (http://www.infomanage.com/internet/hacking/faq/crypt7of10.html)[5] they point out that: "The MD5 hash function is a one way math function and for some one-way hash functions it's also computationally impossible to determine two messages which produce the same hash value."

As I understand it, this also means that with the MD5 function you can't alter a file in any way and end up with a duplicate hash value. With the CRC function, however, there is a way to alter the file and still have the same resultant value. Therefore, the MD5 function is a more secure method of finding a unique value to identify a file than the CRC-32 function.
Our research indicated both the MD5 and the CRC hash systems are implemented within the PERL programming language as modules. This is important since PERL was chosen for the scripts. This choice was made because we have in-house staff to support PERL programs. Therefore, we incur no extra cost from outsourcing the work. This scripting language is also available on a variety of platforms. Therefore the MD5 and CRC hash systems

would both be simple to use within our archival system. Both systems are capable of indicating if a file has been altered. However, being harder to alter the file and create the same value, the MD5 hash system is a more secure system.

OUR PRESENT SOLUTION

Putting all the above considerations and findings together, we now have a new long-term storage tracking system. This solution was chosen because with it anyone can restore previous versions of files as well as identify which files have been changed and when they were changed. The solution was also chosen because it is a lower cost than the present system.

The storage system presently consists of a set of tables on our Microsoft SQL server and a stand-alone Windows 2000 computer that has all the PERL scripts to perform the long-term archiving for our system. The Windows platform was used because this is our standard for all of our computers. We wanted to keep the system consistent. The SQL tables contain a list of the files archived, the date archived, their path, and their MD5 hash value. This information about each file is duplicated, along with the file itself, on the set of CDs that is created each month.

How the system functions.

Our computer systems run a daily incremental backup to handle the day-to-day needs and security of the system. We also run a full backup of the systems every Friday night. The full backup gives us the ability to do a full restore from a catastrophic failure with a maximum of five tape sets. (The last full Friday backup and Monday through Thursday incremental tapes.)

As part of the layered security setup on our systems, we have virus check software running on the server and the stand-alone computer. These both check for any virus, worms, and Trojans that could have been applied to our systems. We also regularly scan the logs on our servers for any unusual activity.

Our SQL system runs on an NT-4 platform and runs the backup built into the SQL program. The transaction table and the complete database is backed up every night.

All of the computer volumes are hardware mirrored, so if anything happens (drive failure, controller card failure, power supply…etc) we should be able to restore the files from a salvaged drive.
A backup for long term storage is made every month. The monthly time interval

was chosen since, based on the timeframe for creating and revising files, more frequent backups would create unnecessary volumes of data. To do it less often might result in lost files. So, at the beginning of every month, usually over the first weekend of the month, a script will take a “snapshot” of the file structure on each volume of our fileserver. The script will then create a list consisting of the volume name, the path, and the file name.

Next, the script will check against the list from the month before and determine which files are new on the system. The system will add the new information, (filename, date, path, and MD5 value) from these new files to the tracking database. These files are then copied to the computer hard drive that will be used to create the CDs. This part of the process assures that all files that are new since the last backup will be burned to CD.

After that, the system will again take the full list of files and run the process of calculating the MD5 value of each file and comparing it to the value listed in the tracking database. If the values match, the program goes on to the next file. If the values do not match, this file name is added to the tracking database with the new date, and MD5 information. This file is added to the files already copied to the computer hard drive that will create the CDs. This part of the process assures that all files that were revised since the last backup will be burned to CD.

There is no need to check if a file has been deleted because we are only placing files in a long-term storage. We are not interested in when the user decides they are finished with the specific file.

As each of the files are moved to the hard drive storage area, the software also creates a file containing all the same information as the tracking database; file name, archive date, volume and path, and MD5 hash value. This information is written to the disc (or first disc if there is enough data to require multiple discs) as the first item at the root of the CD. This is a security precaution that assures neither the tracking database nor the files have been altered since the CD was created.

As an example, let’s suppose that two weeks before the monthly backup a new file, “foo/bar/baz/pita.txt”, was created and another file, “//foo/bar/baz/fubar.txt”, was altered. A third file, “//foo/bar/baz/zip.txt”, was already existing and not changed.

When the system checks for new files, the first one (pita.txt) will be copied to the subdirectory of the computer making the CD because it was found on the present list and not the previous month’s list.

When the system checks the MD5 values against what is in the database, the second file (fubar.txt) will be copied to the computer making the CD because the

MD5 results do not match.

As the system checks the MD5 values against what is in the database, the third file (zip.txt) will NOT be copied to the computer making the CD because the MD5 value matches what is in the database.

The CD is burned with the included items extracted from the tracking database and saved as a comma delimited file at the beginning of the disk.
Five years later someone needs to find the latest copy of “fubar.txt”. To retrieve the file they need only look the file up on the database and select the CD indicated.

Benefits of the new system.

The benefits of the new system are many. A big timesaver is that it has become easier to locate files that have been archived because we can now search a database for the name, location, or the date(s) saved. Sometimes the users can remember that someone worked on a file around a certain date, but not know what the person named the file. Because of the SQL database, we are able to select any files created before or after a certain date.

We can also find out if someone has moved the file to another location instead of copying it as they thought. This would be accomplished by finding the filename on the database and checking its location and MD5 value. If the MD5 value is the same and the location is different, then the file has been moved without being altered.

The task of finding if a file has been changed has become easier. When someone needs to know if a file has been changed all we need to do is get the MD5 hash value and check with the last entry of that file in the archive system.

Since we are using an ISO standard format to burn the discs, we have no problem with reading the files over time. Most systems are built with a backward compatibility to older CD file standards. This means that once old data is moved to CDs we don’t need to keep specialized equipment to read older archived media.

Another advantage of this system is the cost reduction. We previously were storing five tapes per month at an average cost per tape of fifteen dollars. Even if we need to use twenty discs to back up all the new or altered files on the system, we will have a drastic savings because discs are less than one dollar each.
The fact that we can make multiple copies of the backups easier is another advantage of this system. That way we can have an off-site copy of the latest backups rather than risk keeping the latest single copy on site.

The system also serves as a “poor man’s” TRIPWIRE system. If there is any change on the operating system files it will show up when the MD5 value of any of the operating system files changes.

SUMMARY

Where we had no set rules for file archiving, we now have a logical structure for the storage of files. We also have the added benefit of controlling the security permissions for access to directories and files.

Where we had too short of life of archiving media, we now have over one hundred years of expected life of the media. CDs are also cheaper per MB of storage area.

Where we had no consistency of media type, we now have the ISO-9660 CD format that permits any computer we purchase (or have purchased recently) to restore any of the files that have been archived.

Where we had no method of determining when files were changed, or of proving they hadn’t been changed, we can now query a database for what files are new, which files have been changed, what is the latest version, and what hasn’t been changed. We also have offsite and local copies of the full archive file system.

With the new system in place, we now have a more secure, less expensive, and easier to use system to handle our monthly backups.

1. Three, Eugene “Magnetic Media Restoration Company” URL: http://www.vidipax.com/ Under the menu option: magnetic tape preservation, tape storage.

2. Cook, Rick “check the life of 4-mm tapes in backup” 11 June 2002. URL: http://searchstorage.techtarget.com/tip/1,289483,sid5_gci832122,00.html

3. Kodak Corp. “Permanence, care, and handling of CD, Introduction” 14 Sept. 2001 URL: http://www.kodak.com/global/en/professional/products/storage/pcd/techInfo/permanence.jhtml

4. Nelson, Mark “File Verification using CRC” Dr Dobbs Journal May 1992 URL: http://marknelson.us/articles/crcman/crcman.htm

5. “Cryptography FAQ (7/10 Digital Signatures)” infomanage.com 10 October 1993 URL: http://www.infomanage.com/internet/hacking/faq/crypt7of10.html In the section about MD4 and MD5.