Your Independent HP Business Technology Community

The Connection

A Journal for the HP Business Technology Community

INSIDE:

Focus: Best of '09

10
Year in Review

19
NonStop Operations:
Event Analysis & Management

27
Community Voice: SIG "Top 3" Lists for 2009

Will Volume Level Encryption Keep My Data Safe?

Ron LaPedis, CSE, MBCP, MBCI, CISSP-ISSAP, ISSMP
Principal
Seacliff Partners International, LLC
San Bruno, California

Ron LaPedis is principal advisor for Seacliff Partners International, LLC. He spent 25 years with Hewlett Packard becoming a business continuity and security specialist and marketer, consulting with key customers and partners. Before leaving HP, LaPedis filled a dual role as the NonStop Division's senior product manager for security and business continuity products and the headquarters marketing manager for the Asia Pacific region.

There are many questions surrounding the topic of protecting sensitive information, and volume-level or full-disk encryption seem to come up quite often as the answer. This article will discuss the issues related to the use of encryption for protecting information residing on disk storage from theft or modification by touching on these topics:

- Types of information
- Rules and Regulations
- The Objective Defines the Solution
- Encryption Injection Points, Benefits and Tradeoffs
- Key Management

Types of Data

To set a baseline, I would like to classify the information that we are about to discuss. The types of data you usually hear about are Information at Rest, Information in Transit, Public Information, and Private Information. Let me add one more type of data, Information in Use.

Information at Rest refers to that which is being stored, whether online or offline. Examples of storage media would be disk, tape, CD-ROM, paper, and more. Information in Transit is that which is moving from one place to another. It can be moving from disk to memory, disk to tape, one computer to another, or from one data center to another. It should not matter if Public Information is shared in any manner as it is public, while Private Information needs to be restricted to a specific set of individuals, whether this includes everyone in your company, your executive staff, or perhaps information limited to your ethics committee and kept from your executive staff. Ron's rule of information is that anything kept on an Internet-facing Web server is de facto public information.

The last type of information is Information in Use. This information currently is being used, either by an application or a human. At the time this article is being written, it needs to be in cleartext (unencrypted) to be of use. However, stay tuned, since an IBM researcher has solved a thorny mathematical problem called 'privacy homomorphism,' or 'fully homomorphic encryption,' which allows unlimited analysis of encrypted information — data that has been intentionally scrambled — without sacrificing confidentiality.

Rules and Regulations

Depending on your industry vertical and geographic location, you might be subject to rules and regulations covering control over information.

For example, Payment Card Industry (PCI) Requirement 3 states that you must protect stored cardholder data through the use of Strong cryptography with associated key-management processes and procedures. It further states that encryption keys must be changed at least annually.

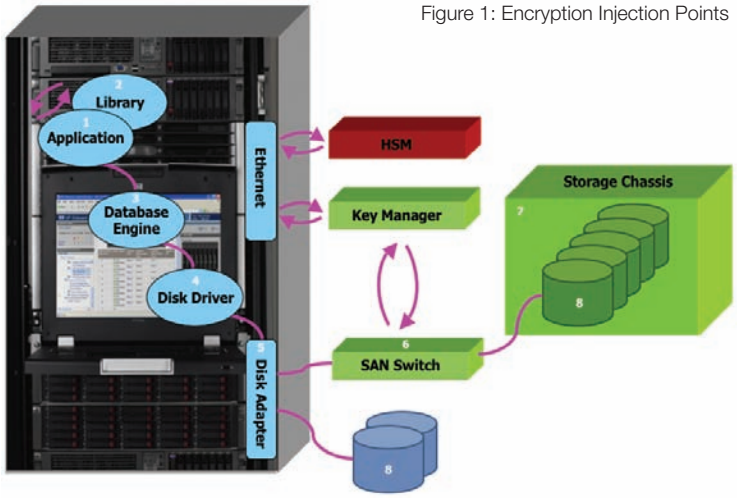
California's SB 1386 law applies to any company doing business in that state no matter where they are located. It states that any breach of the security of the data in the system shall be disclosed…if "unencrypted personal information was, or is reasonably believed to have been, acquired by an unauthorized person…"

The Objective Defines the Solution

Why are you reading this article? Do you have an interest in encryption technologies, do you need to 'check the boxes' on an audit document, or are you sincerely interested in protecting your company's information assets? What are you trying to protect them from – insiders, an external threat, someone who sees your storage area network (SAN), operating system, database, or application as an easy target? Perhaps you are thinking about physical theft of a corporate asset such as a magnetic tape, disk drive, laptop PC, or a server. And what about information stored on media that needs to be returned to the manufacturer when it is replaced as part of a repair or upgrade?

It is hard to come up with an answer unless you are asking the right question. For the remainder of this article, we'll assume that you are sincerely interested in protecting

Figure 1: Encryption Injection Points



your company's information assets. Further, you want to ensure that you have an appropriate amount of protection to mitigate each identified risk.

But what is an appropriate amount of protection? The amount of protection afforded an object should be proportional to its value. That is, you shouldn't spend thousands of dollars to protect an asset worth 20 Euros. Neither should you expect to justify your security spending with a return-on-investment analysis. Security is like insurance in that the amount spent on it derives from a risk-analysis and business-impact analysis.In fact, insurance can be used to back up your security controls to limit losses under unforeseen circumstances.

Encryption Injection Points, Benefits and Tradeoffs

This section will use Figure 1 to describe the various points where data can be encrypted and decrypted. These points include the following:

1. Application
2. Library that is bound into your application
3. Database engine (Oracle, DB2, SQL Server, etc.)
4. Disk driver
5. Disk adaptor or interface
6. Storage area network (SAN) switch
7. Storage array (StorageWorks, EVA, etc.)
8. Disk drive (Seagate, Western Digital, etc.)

The illustration also shows a hardware security module (HSM) connected to the system via an Ethernet interface along with a key manager connected to the SAN switch and the system.

All things being equal, the closer to the application your injection point, the longer your information stays protected. To put this another way, consider what happens if you implement volume level encryption on the disk drive (No. 8). When information enters the disk it is encrypted, and when it exits the disk it is decrypted. If someone steals the physical drive, your information is protected. However, if someone can tap into the storage chassis, SAN, or log in to the system, your information is not protected.

Application or Library

For what is arguably the best protection, you can buy third-party applications that have encryption functionality built into them. If you use a home-grown application, you will need to provide your own functionality by coding it yourself or by buying a source or object code library that can be compiled in or bound to your application.

If you are not able to modify your application for any reason, you might be able to use a library that will intercept file or database calls and perform encryption functions as the data is being accessed. You may need to take this approach if you have lost your source code or the third-party vendor does not want to or cannot add encryption functionality.

In any case, ensure that the algorithm being used is publicly vetted, such as the Advanced Encryption Standard (AES), Blowfish, Triple Data Encryption Algorithm (Triple-DES), or Truecrypt. The Data Encryption Standard (DES) has been broken and is now considered to be insecure for many applications.

If you are approached by a company offering up a secret encryption algorithm, run the other way quickly. It is not the algorithm that must be secret, it is the key. By using a publicly vetted algorithm, you are assured that many experts have evaluated it and proven it effective. In many cases you will want to use a Federal Information Processing Standardization 140 approved algorithm and/or library.

You need to be concerned with encrypting fields that need to be sorted, searched, or used for joins, because they will need to be decrypted to be accessed which could use a large number of CPU cycles. While encrypted data usually cannot be used as a primary key, there is a workaround.

An online retailer that stores a credit card number for convenience does not need to use the credit card number as the primary key – the customer's account number will be used instead. This means that the credit card number can be held in the database in encrypted form and only decrypted when it is needed for a purchase.

Conversely, a credit card issuer will use the credit card number as the customer's primary key. While you must decrypt every key while doing a partial key search, this is not the case when searching for a specific key. The trick is that rather than decrypting every key as you search for the specific record that you want, you encrypt the information for which you are searching then perform the key position. For example, when a customer calls for service, the account number that they provide is encrypted and the encrypted value is used to perform the key lookup.

If you find that you perform specific partial key searches often, you can still use the encrypt-then-position trick by hashing the partial key and storing it in its own field.

With this approach your application needs to do the heavy lifting, and when encryption algorithms are updated your application needs to be updated as well. You also may be using expensive CPU cycles that are better spent processing transactions. However, you are able to control what is and is not encrypted at the most granular level.

You need to ensure that you have a competent development and QA staff well-versed in secure-programming methods, but you are assured that your data is safe unless someone hacks into the application itself.

Database Engine

Many database engines support the ability to encrypt the entire database, selected tables, or columns. By allowing the database engine to control the encryption, you are relying on the database vendor's data security experts instead of your own. You might have less granularity than if your application were doing the encryption. However, because the development is spread across a much larger customer base, the vendor can put more resources into ensuring that the coding is solid and the algorithm is fast.

Even when the database engine is doing the encryption, you still need to deal with the same concerns about encrypting fields that need to be sorted, searched, used for joins, or as primary keys that are listed above, and you can't use the encrypt-then-position trick unless the database knows how to do it. While the database is doing the work so that you don't have to, the same expensive CPU cycles will be used.

Whether it is encrypted in the application or the database engine, the downside of using an encrypted primary key is that it is hard to predict the key space, and therefore properly partition the database for space or speed. Let's assume that your primary keys can range from 0000 to 9999, you currently have 2,500 accounts and the account numbers are issued randomly. If your database is spread across four volumes then using ranges 0000-2500, 2501-5000, 5001-7500, and 7501-9999 would be a good start to balance the load. However, when your account numbers are encrypted, there is no way of telling what values will be created and theoretically they could all fall into the 0000-2500 range. As many storage arrays can repartition databases dynamically, this may not be a concern.

Because the database automatically decrypts data, anyone who can access the database in any manner can access it. This means that management tools and reporting software can be used to gain access to private information.

Disk Driver

At this low level, the disk driver may not understand application- or database-level data structures but encrypts entire disk blocks. Because it is not concerned with interpreting data or making decisions about what to encrypt, it can be made very efficient. This is the first place where you can implement volume level encryption. However, as the data is decrypted almost as soon as it touches the system, private data can be accessed through disk utilities and backup software. And you are still using expensive CPU cycles to do the work.

Disk Adaptor or SAN Switch

The major benefit of performing encryption services at level five and above is that you are no longer using CPU cycles but are using hardware that was designed for this purpose. The major downside is that you can use lower level utilities to get to the data such as hardware debugging aids. If you encrypt within a disk adaptor you cannot connect another system to your SAN and get to the data as you can if you are encrypting at the SAN switch. This is another location for volume level encryption.

Storage Array or Disk Drive

As with a disk adaptor or SAN switch, the encryption is being performed by hardware that was designed for this purpose. Because the data is decrypted as soon as it leaves the disk or the array, anyone who can access the SAN can get at your private data. This placement of volume level encryption is as close to the disk as you can get.

Key Management

How many applications, systems, storage arrays, tapes, and disk drives do you have? How many of them need to be encrypted? How often to you need to change your encryption keys? How long do you need to archive information? The point that I am trying to make is that you are going to be juggling a lot of keys. Not only keys that are in current use, but also keys that are needed to access archived information. In the United States, you must have financial information available for at least seven years. Manually managing seven years of encryption keys across thousands of tapes and servers and hundreds of applications and databases will quickly get out of hand.

A handful of manufacturers sell automated-key-management systems that will store, track, and distribute keys as needed. Unfortunately there are no key management standards yet, but Hewlett-Packard, IBM, EMC Corp./RSA Security, and Thales Group led a coalition of vendors that submitted a standard for interoperability between key management systems and encryption devices to the Organization for the Advancement of Structured Information Standards (OASIS).

To Wrap up

I've given you a lot of food for thought, but I really haven't answered the question asked in the title of this article. Will volume level encryption keep my data safe? So far we have only talked about data that exists in your data center. Because it is very rare for someone to back a truck up to your data center and steal your disk drives, the answer no, it cannot reliably protect your data from most of the risks that your company might face. But on the other hand, I heartily recommend that you use volume level encryption for every desktop and laptop computer for which you have responsibility – especially laptops.

According to a Ponemon Institute (www.ponemon.org) report dated June 30, 2008, companies lose a lot more laptops than they disclose - approximately 12,000 laptops a week in U.S. airports. And 40 percent of those lost are accidentally left behind at security checkpoints.

Of the laptops that are found, just 33 percent are reclaimed by their owner. The rest are sold off, leaving "potentially millions of files containing sensitive or confidential data that may be accessible to a large number of airport employees and contractors," said Larry Ponemon, chairman and founder of the Ponemon Institute. "It's staggering to learn that up to 600,000 laptops are lost in U.S. airports annually, many containing sensitive information that companies must account for. IT departments must re-evaluate the steps they're taking to protect mobile professionals, the laptops they carry, and company data stored on mobile devices."

So in this case, volume level encryption will keep your data safe. Protecting the information on your servers will take a little more work and the first steps are to determine your objectives and the resources that you can bring to bear to meet them.