Database Security

Catherine Boothman D12127081, Oisin De Conduin,
Eoin Gillen, Sean Nolan,
John Cahill D12127121, Máirtin Mac Colgain D12127151

October 20, 2013

Abstract

Database technology has changed over the years and with the advent of Cloud Computing and Web Based Storage small companies and organisations whose budgets would have precluded them from purchasing a Database Management System (DBMS) in the past now find that they can well afford to have one. As more and more of these businesses rely on database technology to store information critical to their day to day operation the security of their databases continues to become an increasingly more important and complex issue. The traditional problems associated with databases were the loss of data through physical damage to the on-site database server and the control of correct and appropriate employee access to sensitive data. There was minimal, although not negligible, risk of unauthorised access from non-company personnel and database damage as a result of internet connections. However as cloud based DBMS solutions are becoming more and more available, both as a backup to large companies and as an affordable option for others, much more needs to be done to keep data safe. In this paper we outline the common threats facing databases and, where possible, the measures that can be taken to safeguard them and mitigate any potential risk to data.

1 Introduction

Databases and database systems have become a major part of everyday life. Nearly everything we interact with and rely upon has some kind of dependence on a database, e.g. [1] withdrawing money from the bank, paying tax and ordering products on-line. For decades nearly every business has made use of some kind of database technology to manage employee records and customer, product and supplier details. These are vital business functions and yet they are the simplest and oldest uses of a database. With the advent of web based applications and information systems technology both the potential use of databases by businesses and their consequent reliance upon them have exploded. Never before has security been such a major issue. Cloud storage and applications are opening up affordable space and processing power to smaller businesses and social organisations who, because of their low turnover, could never have dreamed of being able to have a traditional DBMS housed on-site in the past. However with this technology a whole new array of security risks arise which have the potential to expose companies' private data to the full public view of the world wide web.

Security threats can be categorised into three main areas, viz: Unauthorised Data Observation, Incorrect Data Modification and Data Unavailability. All businesses and organisations may suffer heavy losses of finance and reputation due to unauthorised data observation. Incorrect modifications of data, either intentional or unintentional, result in an incorrect database state and the use of inexact data may in turn lead to heavy losses. When data is unavailable, crucial information for the proper functioning of a business or organisation is not readily accessible [2]. Ask anyone who had an Ulster bank account last year how

annoying frustrating this is!

In order for a database to be considered secure we must therefore ensure that the following requirements are met:

1. Data Privacy Protection against unauthorised disclosure (a.k.a. secrecy or confidentiality);

2. Data Integrity referring to the prevention of unauthorised and improper data modification;

3. Data Availability referring to the prevention of and recovery from hardware and software errors, physical damage to them and malicious data access denials creating database system unavailability.

These three requirements arise in almost all application environments [2]&[3] and will be dealt with in more detail later in this paper.

2 Threats To Database Security

2.1 Human Error

One of the biggest threats to database security is from authorised users who do not follow proper safety protocol because of carelessness or a lack of education in this area. This can lead to 'Authentication Errors' [3]. These user errors include:

- Leaving a computer logged in while unattended, e.g. going out for lunch;

- Weak password, e.g. common, simple, obvious passwords which hackers can easily work out;

- Visiting illegitimate websites;

- Downloading malicious attachments;

- Illegitimate disclosure of information to unauthorised personnel, e.g. password [14].

Email, instant messaging and Social Networks are the most common means of virus spread. Users who are easily fooled through clever social engineering to open emails, click on links or download attachments are unwittingly opening up databases to harm. Phishing is a form of email attack that uses social engineering to convince a user to hand over sensitive information, e.g. a password to a cracker who uses it to attack databases [14].

Non-repudiation methods should be used to track a user's access to and manipulation of a particular database so that if an error or attack occurs it can be traced back to the user's account [3].

2.2 Malware Attack

Although databases can be configured to allow local access only they are being increasingly placed on the cloud or on a specific server thus allowing access through the Internet or across a network. This openness creates possible security threats from planned computer attacks. As more important business, financial and personal information is uploaded into databases they become enticing and profitable targets for intruders [14].

Malware is malicious software. It comes in many forms, is designed by intruders to carry out unauthorised activity on a computer or network and can often damage or destroy a database [14]. There are over 600,000 malware programs at large on the Internet.

2.2.1 Viruses/Trojans/Worms/Spyware/Bots

Viruses are self-replicating pieces of software that load themselves onto a computer without the user's consent. Once installed they can often carry out harmful attacks on databases, e.g. access private information, log keystrokes and corrupt and manipulate data [14].

Trojans disguise themselves as legitimate but are malicious programs. Unlike viruses they do not replicate. They can be used to harm or manipulate databases and to carry out other

remote attacks [14].

A worm can be seen as a specialist version of a virus with the major added danger that it does not need user input to travel through a computer network. Worms are self-learning. They will use information on vulnerabilities that they find on one computer on a network to infect the next and can be programmed to harm databases [14].

Spyware is a type of virus that monitors a computer or database without the user's knowledge and steals information which is sent back to be used as the Spyware creator wishes [14].

Crackers are able to control thousands of computers by infecting them with Bots which are malicious self-installing programs that can be used for DoS (denial of service) attacks where they flood a CPU with requests, slowing it down until it becomes unusable. They can shut down networks and databases in the same way [14].

2.2.2 SQL Injection

SQL Injection is a particular danger to databases and other data driven systems. It involves a malicious SQL statement being inserted into the entry field of a database and then being executed. Hijacking website URLs and sending information through a web-browser is a common attack 'vector'. SQL Injection is used to reveal confidential database information to an attacker and to carry out unauthorised and harmful manipulation of databases [14].

2.3 Natural Disasters

Natural disasters including earthquakes, flooding, hurricanes and fires present an uncontrollable risk to data centres and databases. The threat is twofold. There is the immediate danger to on-site equipment but this could also lead to a second difficulty in terms of service downtime especially if the databases affected are critical to on-site and/or offsite services resulting in a possible loss of revenue and customer confidence.

Response time is crucial in preventing damage to database systems, e.g. fire which can spread quickly through a data centre endangering personnel, equipment and on-site backups. While natural disasters may not necessarily touch the data centre itself they can damage the infrastructure in the surrounding area such as power and communications in which case redundant services may also be needed to keep databases up and running.

During Hurricane Sandy's hit on New York in 2012 data centres in Manhattan were cut off from power and communications. The subsequent flooding led to damage of essential equipment and services which brought databases offline for a number of weeks until a restoration effort could be carried out. Data centres located in basements and lower floors and those dependent on basement backup power supply pumps were worst affected [8].

2.4 Hardware Failure

Given that most databases run on a continuous basis hardware failure must be factored into the successful operation of a data centre. The hard drive is the most common component to fail followed by memory, CPUs, motherboards and power supplies [13].

Mechanical storage is the most cost effective method for storing large volumes of data on a non-volatile medium[7]. However due to the drives' structure and moving parts they are prone to failure as they age.

2.5 Human element

Apart from the security risks discussed above data centres and databases can also undergo physical attacks from humans. These are either unintentional or malicious.

Encryption of data makes it difficult to glean any sensitive information from stolen hard drives. Databases and larger data centres now contain a myriad of high value equipment ranging from complex servers and network switches

to fibre cable and tools.

Equipment can also be damaged during break-ins as occurred in Vodafone's Basingstoke data centre in the United Kingdom [12]. Thieves caused damage to a number of servers resulting in a service outage to Vodafone's voice SMS and internet services.

It is also possible for an organisation to have its database threatened from within, be it database sabotage or data falsification, where employees maliciously cause service disruption and data loss integrity. These acts can be carried out for the purposes of committing fraud or be purely destructive and are only prevented by having an adequate security system in place including restricted access where only designated vetted staff are allowed in sensitive areas.

3 Security Measures

3.1 Physical Risks

3.1.1 RAID

One of the easiest and most common ways to protect a database against hard drive failure is to use RAID (redundant array of independent disks) [15]. This is an umbrella term used to describe data storage schemes that divide and replicate data across multiple hard drives in real time. This allows a server to use multiple hard drives at once both increasing performance and facilitating easy data recovery should a hard drive failure occur.

3.1.2 Off-site backups

In the event of a flood, fire or other disaster, an entire server could be destroyed. One of the many ways to secure a backup of a database system against these possible physical risks is to keep a secure off-site backup away from the main site. Thus it is possible to recover the data from the off-site backup even if the site of the main server is compromised.

3.1.3 Cloud storage

Storing a backup in the cloud allows for recovery similar to storage on an off-site location. Although it may be more convenient keeping sensitive data in the cloud may not be as secure as keeping it in a separate off-site location [16].

3.2 Human Error

3.2.1 Role-Based Access Control

The most straightforward way to protect a database against human error is to tightly control access to the database. Applying the principle of least privilege make sure that people only have access to the areas of the database that they need and nothing more. The challenges include handling multiple roles, conflicting roles and consistency of the access control rules.

3.2.2 Testing

The fault-tolerance computing community has come up with several algorithms for recovering databases and systems from failures and other problems . These techniques included acceptance testing and checkpointing[3]. Sometimes data is replicated as aforementioned so that there are backup copies. Therefore it is imperative to make sure that all code is checked and tested before it goes live. A single bug could be enough to take down an entire system so proper testing is needed to ensure that there are no problems that could compromise the integrity of the database. We also need flexible security policies as there may be conflicts between requirements such real time processing and security [3].

3.2.3 Risk Analysis

Before developing any computer system for a particular operation one needs to study the security risks involved. The goal is to mitigate the risks or at least limit and contain them if

the threats cannot be eliminated. The challenges include identifying all the threats that are inherent to a particular situation. This is especially useful for viruses. Once a virus starts spreading the challenge is how do you stop it or if you cannot stop it how do you contain it and limit the damage that is caused [3]? Although various antivirus software packages will possibly limit the virus from affecting the system or causing serious damage new viruses are being developed all of the time and therefore it would be prudent to stay one step ahead.

3.3 Malicious Risks

3.3.1 Firewalls

Various Organisations now have web infrastructure for internal and external use. To access the external infrastructure one has to go through the firewall. These firewalls examine the information that comes into and out of an organisation. Thus the internal assets are protected and inappropriate information may be prevented from coming into an organisation [3]. It is also possible to throttle connections through the firewall to increase security and reduce access to database servers.

3.3.2 Data Mining

Data mining is the process of posing queries and extracting patterns, often previously unknown, from large quantities of data using pattern matching and other reasoning techniques. Data mining may be used to detect and possibly prevent cyber-attacks. There are many examples. Anomaly detection techniques could be used to detect unusual patterns and behaviours. Link analysis could trace viruses to perpetrators. Classification may be used to group various cyber-attacks and then use the profiles to detect an attack when it occurs. Prediction could determine potential future attacks. Data mining can also be used for analysing web logs as well as analysing audit trails. Based on the results of the data-mining tool one can determine whether any unauthorised queries have been posed [3]. It is important also to note that data mining gives out information not previously known using various reasoning techniques such as statistical inference which exacerbates the inference problem whereby unauthorised information can be deduced from legitimate responses to queries posed. This needs to be considered and appropriate security and privacy constraints applied to all data sources.

3.3.3 Protecting information

While ideally the information in a database should never be compromised, sometimes there is no amount of preparation that can stop it. Even the biggest companies with high-end security can be breached, such as with Sony and the Playstation Network [17]. In cases like these, it is important that all sensitive data be encrypted in some way.

Passwords and other information can be encrypted using a hashing algorithm. This takes a block of data and returns a fixed-length string called the hash-value. Even a single character changing in the data block will return a completely different hash value. It is possible to store this hash value instead of the actual password so that even if a database is compromised all the attackers get is a list of encrypted passwords.

Let us look at other types of information. Using hash functions on a message creates a message digest. If appropriate functions are used each message will have a unique digest. Therefore even a small modification to the message will result in a completely different message digest. Thus integrity is maintained. Message digests together with cryptographic receipts, which are digitally signed, ensure the receiver knows the identity of the sender [3].

It is however possible to reverse engineer the passwords from the hashed values if the hashing algorithm is known. This can be done using a dictionary attack or something similar. To prevent this passwords can also be salted before

they are hashed by inserting random characters into them [18].

3.3.4 Database hardening

It is extremely important to keep up to date with patches as indicated by the SQL Slammer worm which affected circa 100,000 systems in January 2003 exploiting the buffer overflow problem. The worm was made possible by a software security vulnerability in SQL Server first reported by Microsoft on July 24, 2002 [19]. A patch had been available from Microsoft for six months prior to the worm's launch but many installations had not applied it.

Furthermore stored procedures and triggers can lead to privilege escalation and compromise and this should be considered prior to establishing databases.

Finally database processes should be run under dedicated non-privileged accounts through applications and unneeded components of said databases disabled.

References

[1] Ramez Elmasri and Shamkant B. Navathe, *Fundementals of Database Systems, 4th Edition*, Pearson (2004)

[2] Elisa Bertino and Ravi Sandhu, *Database Security - Concepts, Approaches and Challenges* IEEE Transactions on Dependable and Secure Computing **2**, (2005)

[3] Bhavani Thuraisingham, *Database and Applications Security*, Auerbach (2005)

[4] Amichai Shulman, *Top Ten Database Security Threats: How to Mitigate the Most Significant Database Vulnerabilities*, Imperva Website

[5] Sumit Jeloka, Don Gosselin and Richard Smith, *Oracle®Database Security Guide, Release 2*, Oracle and/or its affiliates (2012)

[6] `http://www.cableorganizer.com/articles/how-to-fireproof-your-server-room.html`

[7] `http://research.microsoft.com/en-us/um/people/navendu/papers/sigcomm11netwiser.pdf`

[8] `http://arstechnica.com/information-technology/2012/10/hurricane-sandy-takes-data-centers-offline-with-flooding-power-outages/`

[9] `http://arstechnica.com/information-technology/2012/11/how-one-nyc-data-center-survived-hurricane-sandy/`

[10] `http://www.techrepublic.com/article/disaster-recovery-how-to-protect-your-systems-from-flood-threats/`

[11] `http://www.sans.org/reading-room/whitepapers/awareness/data-center-physical-security-checklist-416?show=data-center-physical-security-checklist-416\&cat=awareness`

[12] `http://www.computerworlduk.com/news/mobile-wireless/3262920/vodafone-in-network-outage-following-datacentre-break-in/`

[13] `http://www.datacenterknowledge.com/archives/2008/05/30/failure-rates-in-google-data-centers/`

[14] Alfred Basta and Melissa Zgola, *Database Security*, Delmar Cengage Learning (2011)

[15] Donald, L. (2003). MCSA/MCSE 2006 JumpStart Computer and Network Basics (2nd ed.). Glasgow: SYBEX.

[16] Winkler, Vic. "Cloud Computing: Virtual Cloud Security Concerns". Technet Magazine, Microsoft. Retrieved 12 February 2012.

[17] `http://www.cbc.ca/news/technology/playstation-data-breach-deemed-in-top-5-ever-1.1059548`

[18] http://www.dshield.org/diary.html?storyid=1111

[19] http://www.giac.org/paper/gsec/3091/ms-sql-slammer-sapphire-worm/105136