1 Database and Cloud Security

1.1 Need for Database security

Firstly, why do we need database security? How do we implement it? DBMS systems can be quite complex already, and implementing security measures on top of that can increase that complexity. However, due to our reliance on databases, we should really invest in it.

There are several reasons why securty is not a more major focus already. First, the users sometimes don't know how to implement it, since they might only know how to interact with the database system and not how to implement security.

Firstly, what is a database? THey are a collection of data stored by an application or service. They are a relation between the user (or the object about which they store information), and the items or fields which they try to implement. The user or process can use a querying language to access the database.

Databases can require extra security measures not provided by the OS, maybe some read and write operations. Then there are relational databases, where items are stored in a big table, where each column is called a field and each row is a record. Each column contains the specific field for the user.

These tables are linked by identifiers and common elements, and the querying language can retrieve the desired info. A ***primary key*** is an element that by itself can uniquely identify a row.

1.2 SQL Injection Attack

This is one of the most prevalent and dangerous types of attack (major services get attacked all the time).

This type of attack is designed to exploit the natore of the web and web apps. The attack then tries to send a request to the database that extracts large amounts of data. Can launch all sorts of attacks depending on the types of requests sent.

How do we perform these attacks? If you recall, comments in SQL start with the – characters. Then we might try to prematurely terminate a string and inject some other command. A clever user might try to implement a carefully crafted input string to take advantage of this fact.

The categories for SQL attacks might be the following:

1. **In-band attacks**: uses the sam ecommunication channel for injecting SQL codes and retrieving results. So an attacker on the web might use a malicious query, but still carry it out on the web.:

   These types of attacks can be classified by three kinds. *Tautologies* insert code in condtitional statements so that they always come out to be true. In *End-of-line comment* type attacks, we can insert some code into the field and then nullify whatever comes after it by using the comment characters.

Finally, in *Piggybacked queries*, the attacker can add additional queries, thus retrieving information not meant for him/her.

2. **Out-of-Band attacks**: These attacks might use a different channel to carry out the deed, like HTTPS, DNS resolutions, email, etc....

   Clever designers might want to place limitations on the types of queries we answer, and the channels on which they are sent. If we vary the channels on which we perform the attack, then we might be able to get around the limitation.

3. **Inferential Attacks**: By sending particular requests, we can infer characteristics of the database without carrying out an actual breach.

   With inferential attacks, we can gather information about the database, usually in a correct manner, by analyzing overly-descriptive error messages on purposefully wrong queries. Thus we can infer what kind of data the database holds. A blind SQL injection then refers to asking true/false questions to the database, because we can then know about certain types of info present in the queries.

Then how can we counteract these queries? There are defensive practices the admins might take to ensure that the items in the database are not compromised.

1.3 Database Access Control

We have previously assumed that the users have been authenticated. The database admin might want users to be able to freely grant access rights to other users. The users might also not be able to access the database, since the info there is sensitive.

Typical grants gives to a user might be the ability to insert or extract info into the database (`INSERT` and `SELECT`, for example).

cascading rights: Whenever we want to grant or revoke the access rights to a user, any user that has been authenticated by that user will also have their rights revoked.

1.4 Role-Based Access Control

With roles assigned to the users, we can have increased securtiy and less burden. A database with this ability needs to be able to interact with all these roles; add, delete, move, etc... Three types of users: *application owner*, *end user*, *administrator*. The roles that these users have over the database dictates what information they can access or what access rights they possses. The owner owns the database; the end user operates on it but typically does not know how it works; the administrator has the responsiblity for members of the database.

1.5 Inference

Inference is the process of making actual valid queries to a database and deducing other inforation from that. This could come from error messages or other types of ouptut, such as true/false questions. This might be done by taking advantage of *metadata*, which is the information *about* the information present on the system. The *inference channel* is the means by which the information is obtained.

How do we detect inference?

There are two main approaches. We could try to prevent it form happening during the design of the database system, or we could try to detect it at the moment the query is made. However, we need an algorithm to detect what actually constitutes an inference type approach.

We could also create many more databases and have a finer grain ocntrol of those databases. This way, accessing all the information of a particular user is preferred.

1.6 Database Encryption

Because a database consists entirely of information, it can be the most important part of any business. Due to this, there are several different layers of security present in a database, to prevent any sort of unawanted access. Where in the database do we apply the encryption though? We could do it on all rows, or all columns, or each individual field.

The disadbantages of encrypting the database are: inflexibilty a the moment of searching for an index. The user must have the key for the field that they want ot obtaain; the key gives them the permission to do that.

To ensure at least a level of security, we can maybe encrypt any query that is going to the database.

1.7 cloud computing

Our goal with cloud computing is to deliver availability of a resource to our users. This can be done to support our *Software as a Service* model, where the user does not pay to use the software, and we offer ongoing support for our service.

We can abuse the resources provided by these services. Signing up for these services is free, which could open them for a plethora of attacks.

The APIs used to connect witht hem can also be exploited. Ensuring a strong authentication system is paramount. Some of the elements of the cloud service weren't designed to be isolated.

2 Chapter 6: Malicious Software

What is malware? Malware is a program that is inserted into another computer that inhibits the normal operation of a computer system.

2.1 Classification

One way to separate different kinds of malware is to see how it spreads. A program like this is designed to wreak havoc on a user's pc and then move on to the next.

Another way to classify malware is by seeing what kind of payload is being carried. Are we stealing the user's info? Are we encrypting it only? If we encrypt it, we probably will charge the user some amount of money to return the decryption key.

Is the piece of code independent? Does it need a host program? Programs like worms, trojans, and bots are independent.

Does the malware replicate? If so, it might be a worm, else, it might just be a trojan or spam email.

The market for malware and the information obtained from it is great, and it has to be done mostly on the black market.

2.2 Advanced Persitent Threats

With these types of threats, they are persistent and stealthy, so the user might not have any idea what is happening behind the scenes. Over an extended period of time, the chances of success are greatly increased. The goal of APTs is to infect the user computer with malware.

Yet anoher way to classify viruses is by their target. Which area of memory does it affect? There are some that affect the boot sector of the drive, which will cause the virus to spreak on boot. It can also infect other executable files where the shell or OS thinks that it is a regular exe file.

Yet even another way of classifyin viruses is by their *concealment strategy*. How are they keeping themselves hidden? There are some that will encrypt themselves, will have a part of their script that generates a encryption key. Another type called *stealth virus* will alter a piece of code or use other editing techniques to conceal itself from antivirus software that might be installed on the computer. Yet another type of virus called *metamorphic* viruses can change with every infection; rewrites itself every iteration.

In the 90's, there was a common type of atack called the macro attack, where various scipts could be injected into user files, such as PDFs, which could carry with them embedded JavaScript. The 'Melissa' virus, popular for MicroSoft prooducts, sent an email to the first 54 contacts if the email client was set as 'Outlook'.

2.3 Worms

A worm is a program that actively seeks out other machines to infect; it seeks to spread itself to other users. Can (and has to for the homework) infect other machines through secure means such as SSH.

It can use email or other messaging services to spread itself. Then, it creates an autorun script on the host machine to call it after a certain time has passed.

If the worm has access to the password and knows how to crack the password of a victim, it can login remotely and even transfer files..

The first mainstream computer worm was the Morris Worm. This worm was designed to infect UNIX systems, by taking advantage of the pasword file and cracking it.

2.4 state of Worm technology

Today there are many scripting languages that are supported by modern OS systems like MacOs and Linux. Today ther are also many networking apps that may be infected, more than in the 90's. The spreading of these worms is of the topmost priority, since youwant to locate as many vulnerable machines in the shortest amount of time.

These worms can easily create copies of themselves, by allowgin for *polymorphic behavior*, in this case rewriting itself in a way that is functionally equivalent but implemented differently.

Nowadays many viruses can spread through the web, by Java applets or scripts embedded into webpages. User's information can then be tracked easily with software. Modern worms can even infect phones. Java plugins and Adobe Flash have been exploited through the years.

2.5 clickjacking

With this type of attack, multiple opaque layers are applied to the webpage, and the user's information is actually being given to the attacker when he or she thinks it's being used for legitimate purposes.

2.6 payload

A popular type of virus encrypts, corrupts, or simply destroys data on the unsuspecting user's computer. Sometimes the code is dormant and will only execute when certain conditions are met.

botnet: A collection of bots that can be used in a coordinated manner (DDOS, sniffing, keylogging, etc...)

Another type of attack just steals information from the user. These attacks can track whatever the user enters into his/her machine, or has a set of keyworkds that it is looking for. This type of atack is related to the concept of spyware, where the victim's machine is able to be tracked and monitord.

phishing: refers to the use of social engineering to trick the user into thinking it's the legitimate service it claims to be. Ususally can be triggered by clicking a fishy link from an email.

After all this, how do we stop malware from infecting our machines? Just don't install it bro. Only use trusted services and don't open weird links. When we want to counteract these threats, we need to consider that the user shouldn't notice any adverse effects in daily operation; i.e. the program used to resolve it cannot be too heavy as to render the machine useless.