Sensitive Data Exposure

Project Report

by

Sarik Koirala(2014UCP1012)
Pratyush Pratap Singh(2014UCP1013)

Under the supervision of

Professor Dr. R.B.Battula

Computer Science Engineering Department
Malaviya National Institute of Technology, Jaipur
May 2017

1 Introduction

Data loss or exposure data is transmitted and unauthorized access to data from an organization's or an external organization's database. During the management of any company or organization, sometimes sensitive data must be handled with care and should stay away from illegal / unauthorized, filtered, making business as well as vulnerable data. This type of exposure can be a serious threat to the privacy of any organization or personal data of each user. According to 2013, they reported that more than 800 million sensitive disks were exposed by 2,000 incidents. While many of these losses are due to attacks, there are many losses that are caused by unintentional errors of employees in system design or proprietary data. Sensitive data exposure may occur at any time when an application does not protect the confidential data being revealed to attackers. For many applications, these data can store a lot of important information such as passwords, credit card data, session tokens, or other authentication credentials. The main objective of this project is to study how these sensitive data can be exposed and ways to prevent attackers from accessing this sensitive information.

2 Sensitive Data Exposure

Sensitive data exposure Sensitive data exposure is a security incident in which stolen, seen, or used by an unauthorized malicious user of sensitive and sensitive data is stolen. Private data may include financial information such as credit card or bank, personal health information (PHI), personal information (PII), business secret secrets. Most of them involve vulnerable files unstructured data or other sensitive information. Usually, these are accidents such as theft or loss of digital media such as laptops containing information stored in clear, publish such information on the World Wide Web or on a computer accessible from the Internet, without adequate security measures, a system Which is not completely open but is not properly or formally accredited for security at the approved level, such as clear e-mail or the transfer of such information to information systems of a possibly hostile body such as a competitor or a foreign nation , Where it could be exposed to more intense decryption techniques.

The 5 major ways sensitive data can be exposed is:

1. SQL injection 2. Cross-site request forgery 3. Man in the middle attack 4. Session Hijacking 5. ID Spoofing

3 SQL injection

SQL injections are used by aggressors to attack data-based applications, including malicious SQL statements that are placed in an input field for its execution (for example, to download the attacker's content). Explore security vulnerabilities in an application interface. It can be used to attach SQL data in various ways.

This threat is serious because 65 percent of organizations in this world have experienced a SQL injection attack that has succeeded. Nearly half the respondents state the threat of SQL injection in front of your company is very important. On average, respondents believe that 42 percent of all data breaches are due, at least in part, to SQL injections. SQL injection (SQLI) is considered one of the 10 most vulnerable web application vulnerabilities 2007 and 2010 by the security project's open web applications. In 2013, SQLI was rated as the number one attack in the top 10 OWASP.

Example:-

select * from user where username= or 1=1; – and

The above SQL statement is a tautology example. Single quotation marks in the sentence after "username =" and "- -" balance the quotation reference and double hyphen makes a comment about the rest of the query. The result is that the password value field becomes irrelevant and can be sent to any chain. As a result, the previous statement runs as an SQL query on the database that exposes all information about all users in the database table

3.1 Counter Measure against SQL injection

Following are the counter measures : 1. You must test each SQL statement should have access to the database. Any activity that leads to a deviation from the behavior that is assumed to have marked as security events. Behavioral analysis provides immediate protection against these threats. Many organizations are adopting this method. 2. Check and clean the input data. There is a well known legal control validating data type, length, format, and range. 3. Use security parameters type. Use these parameters with strings built dynamically with SQL commands. Collection parameters such as SqlParameterCollection provide control of the type and length of validation. If you use a set of parameters, input is treated as a literal value and SQL Server is not executable code. An additional advantage of using a set of parameters that can reinforce type and length controls. Values outside the field to trigger an exception. 4. Limit

permissions to the database. Ideally, you should only grant execution permissions for selected stored procedures in the database and does not provide direct access to the table. 5. Avoid displaying error information in the database. In case of errors in the database, be sure not to reveal the detailed error messages of the user.

4 Cross Site Request Forgery(CSRF)

It is a vulnerability exploited on the website. In this, the attacker forks a cross-site request of any type of form. This allows a user's browser to perform a desirable action on a trusted site.

These attacks were called the "sleepy giant" of web-based vulnerabilities, as many websites failed to protect them and because they were largely ignored by web development and security communities. This attack may allow a user to transfer money to bank accounts and collect email addresses of users who violate the privacy accounts of compromised users and users.

Server-side modifications must be applied to protect a system from such attacks.

4.1 Counter measures against CSRF

Here are the countermeasures: 1. Using cards. Tokens are cryptographic values. These are generated when a user session starts and is destroyed at the end of the user session. This symbol should be included in each request, which will be used by the server side. For a malicious user to forge an HTTP request, the victim's special session verification tokens must be met. 2. Exit when you end up using your account. (This is a method that users can use to prevent this type of attack) 3. Make sure you do not save access credentials in your browser and use secure browser extensions. (This is a method that users can use to prevent this type of attack)

5 Man in the middle attack

A man in the middle attack (MITM) is when an attacker is placed in a conversation between a user and an application, making it appear as a regular exchange of information is in progress. The purpose of this attack is to steal personal information as account details. The goals are usually users of financial applications, e-commerce sites, and other websites where access is required. Information obtained

during an attack may be used for various purposes, including identity theft or an illegal change password. The (and simplest) way most common to do this type of attack is where an attacker makes them hotspot free of charge accessible and harmful to the public. Usually the name in a way that matches your location is not password protected. Once a victim is connected to an access point, the attacker gains full visibility to any online data exchange. Example:

5.1 Counter measures for MITM

For users: Avoid WiFi connections that are not password protected. Pay attention to browser notifications to indicate that a website is not secured. Close the application session immediately when not in use. Do not use public networks (for example, canteens, hotels) while performing sensitive operations. For web site operators, Safe communication protocols, between TLS and HTTPS, counter spoofing attacks by encrypting the data transmitted. So intercepting site traffic is avoided and decryption of sensitive data blocks, such as authentication tokens. Applications use SSL / TLS to protect each page of your site.

6 Session Hijacking

When any user gets logged-in on to a web server, a session is created. This session keeps track of user information, including a session ID in order to authenticate requests for data. When the user logs out, the session is ended. Session data is normally stored within a cookie or as parameters in the URL.

Session hijacking involves the attacker taking over a session by getting a valid session ID. The attacker can then pretend to be the authorized user and make requests as that user.

In cloud data services, this could lead to a breach of the victims information, making cloud services particularly vulnerable.

Every analysis of security in the cloud lists session hijacking as one of the top three threats to it. Because of this, we will be taking a more in depth look at session hijacking and some of the defenses.

Also known as TCP session hijacking.

6.1 Counter measures for Session Hijacking

The best way to prevent session hijacking is to allow client-side protection. We recommend taking preventive measures to hijack the client-side session. Users must have effective antivirus, anti-malware software, and should keep the software up to date.

Although there are some of these that are used to prevent this type of attack:

1. HTTPS is a combination of HTTP and SSL (Secure Sockets Layer) standards, providing secure encryption communication over the Internet. HTTPS can be used from one end to the next, throughout the session, to prevent Sniffers attackers packets to get a valid session ID. 2. OTC prevents attacks by signing each user request with a secret session that is stored safely in the user browser. Protecting every user request prevents a malicious user simply frustring by forcing a session ID to any desired request as a user. 3. SessionShield is a proxy outside of the browser that inspects all incoming / outgoing requests. It will help protect the user from session hijacking attacks that manifest themselves as XSS, even if the web application does not properly cushion the potential issues.

7 Sniffer Attack

A sucker is an application that can capture network packets. Sniffers are also known as network protocol analyzers. While protocol analyzers are really tools for network troubleshooting, they are also used by hackers to enter the network. If network packets are not encrypted, the data inside the network packet can be read with a suction cup. Sniffing refers to the process used by hackers to capture network traffic with a sniffer. Once the package is captured using a suction cup, the contents of the package can be scanned. Sniffers are used by hackers to capture sensitive network information, such as passwords, account information, etc.

Types of Sniffer Attachment Sniffing LAN: In this case, the sniffer software is installed on the internal LAN to thoroughly analyze the entire network. This helps to provide more inventory server information, Living Owners, Open Doors, etc. Once all the details are collected, the hacker can launch a specific attack on the port. Sniffing Protocol: This method involves creating separate sniffer for attacks on different network protocols. For example, if a hacker sees UDP packets on a network, a standalone sniffer begins to acquire information. ARP Sniffing: Hackers steal all important information about IP addresses and associated MAC addresses. This data is also used to launch packet spoofing attacks, or ARP poisoning attacks exploit network routing vulnerabilities. TCP Sniffing session: This is a sniffer base attack where the traffic hackers between the source IP address and the destination stop. Target details such as service types, port numbers, and TCP sequence numbers to create and control a TCP session made. Sniffing Network Password: These sniffer attacks, hackers penetrate HTTP sessions that do not use se-

cure encryption. Then, user IDs and passwords can be stolen and used for harmful purposes.

7.1 Counter measures for Sniffer attack

1. Enable a WPA or WPA2 for the router. Also, be sure to change the default password to restrict access to the network. Use a long, secure password consisting of numbers, uppercase letters, lowercase letters, and symbols. 2. Use MAC filtering on the network. It should only allow MAC addresses to access the private VPN by reducing the chances that a tracker light on the network. 3. Make sure that the important sites I use, especially those that involve financial transactions, have Secure Socket Layer (SSL) encryption. If a site is SSL enabled, you will have a URL that starts with HTTPS instead of HTTP.