
















TimeSheets: Threat Report

YOUR NAME:

Tadhi AlAli

DATE: 15 -11 -2022



How to use this Template

- Make a copy of this Google Slide deck.
- We have provided these slides as a guide to ensure that you submit all the required components to successfully complete your project.
- When presenting your project, please only think of this as a guide. We encouraged you to use creative freedom when making changes as long as the required information is present.
- **Remember to delete this and all** of the other example slides before you submit your project.
- **Remember to add your name and the date** to the cover slide

Reference slide remove
before you submit

Purpose of this Report:

This is a threat model report for **TimeSheets**. The report will describe the threats facing TimeSheets. The model will cover the following:

- Threat Assessment
 - Scoping out Asset Inventory
 - Architecture Audit
 - Threat Model Diagram
 - Threats to the Organization
 - Identifying Threat Actors
- Vulnerability Analysis
- Risk Analysis
- Mitigation Plan



Section 1

Initial Threat Assessment

Completed Asset Inventory

Components and Functions

- ***TimeSheets Web Server:*** The web server's primary role is to serve static content to a requesting client through the http protocol.
- ***TimeSheets Application Server:*** The application server handles all the business logic process and serves dynamic content.
- ***TimeSheetsDB:*** The database server stores employee data and will be queried from the application server.
- ***AuthDB:*** Stores user authentication data (credentials) and will be queried from the application server.

Completed Asset Inventory

Overview of Application Functionality

TimeSheets is used by employees to track their hours worked. Users will login to the TimeSheets application from their device.

Data Flow

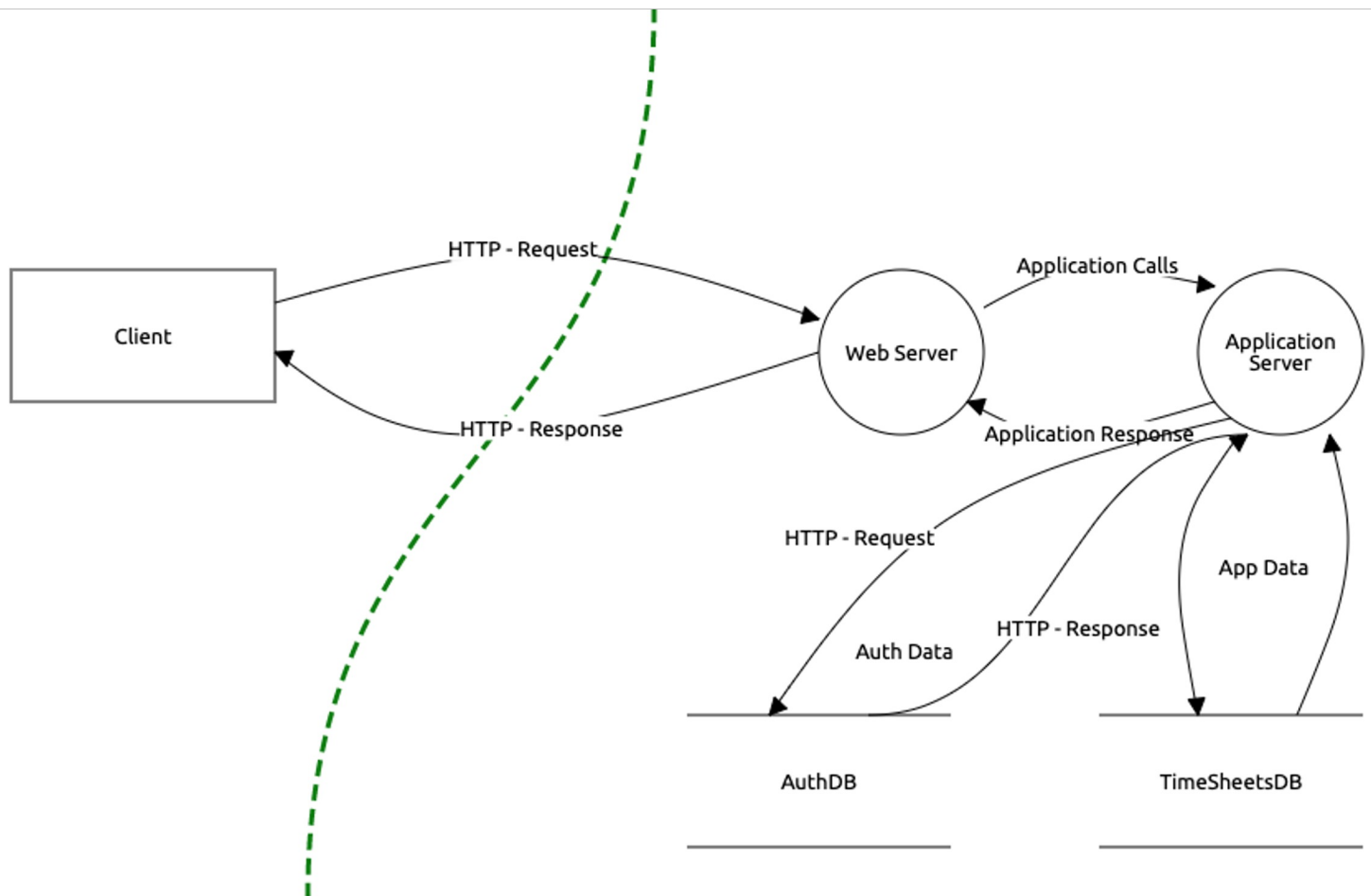
Request is generated from the client via the Internet. The request arrives at the TimeSheets web server which serves static content to the user (HTML, images, etc). Dynamic data is retrieved from the database and served to the client.

Completed Architecture Audit

Flaws

- *There is a lack of encryption at rest - database servers are storing data on unencrypted disks.*
- *There is lack of redundancy.*
- *There is no firewall that is filtering traffic coming from the Internet*

Completed Threat Model



- Employee Data Unencrypted at Rest
- Authentication data is using reversible encryption
- Authentication requests are not encrypted in transit
- Sensitive data is encrypted using DES algorithm

Completed Threat Analysis

What Type of Attack Caused the Login Alerts?

Man in the Middle (MitM)

What Proves Your Theory?

There is lack of encryption between the client and the application. A malicious actor is sniffing traffic and intercepting the requests with a valid username/password in the request. Additionally, the logs show successful login attempts from the expected IP, but also a different location at the same time.

Completed Threat Actor Analysis

Who is the Most Likely Threat Actor?

Internal User

What Proves Your Theory?

The IP address of the unexpected login matches that of an internal user. Additionally, there was no data leaked from the company, and no data changed. Just data accessed that the legitimate user typically doesn't access.



Section 2

Vulnerability Analysis

2.1 Employee Data Unencrypted at Rest

Discovery:

During the threat model the SRE team confirmed that the database is on a server that does not have encryption at rest.

Why is this an issue?

The database company include employee data which are the personal and sensitive information.

If the database dose not have encryption at rest. The hackers can easily get access into server by knowing the sensitive information such as employee login name and password. So, the hackers will know all the company information.

It's will break the privacy policy of the company. By failed to protect employee data.

Also, data breach will lead to affect the company costs.

2.2 Authentication Data Stored Using Reversible Encryption

Discovery:

During the threat model the DBA team confirmed that the database is storing authentication data (credentials) encrypted.

Why is this an issue?

when using reversible encryption in the storing authentication data it's easy to hackers to decrypt.

By that, the hackers can get access to all authentication credentials. They will get access to the company system and data. Meaning they can edit, add, delete company data.

2.3 Authentication Requests are Unencrypted in Transit

Discovery:

During the threat model the security team confirmed that authentication requests are being transmitted in plaintext.

Why is this an issue?

The examples of the risks due to unencrypted authentication requests in transit are Man in the middle and eavesdropping.

The hacker can get access to all the request, authentication factors and communication between devices.

2.DES Algorithm in Use

Discovery:

During the threat model the security team identified sensitive data being stored using the DES algorithm.

Why is this an issue?

With help of brute force attack, the DES algorithm has 64 bit key. By that the number of combination is small and a simple person computer can break it.

The same output will be created when two input given to s box.

Optional Task:

Examine the threat model diagram from Section 1 and answer:

What non-encryption issues can you identify?

What recommendation would you give to solve those issues?

Why do you recommend those solutions?

- *[Issue 1 Here]*
- *[Issue 2 Here]*
- *[Add more issues as necessary]*



Section 3

Risk Analysis

3.1 Scoring Risks

Risk	Score <i>(1 is most dangerous, 4 is least dangerous)</i>
Unencrypted at Rest	3
Reversible Encryption	4
Unencrypted in Transit	1
Outdated Algorithm	2

3.2 Risk Rationale

Why Did You Choose That Ranking? Make sure to include your risk ranking methodology. *(Did you use a tool or defined risk scoring system?)*

- Reversible encryption ranked as 4 due to that attacker finding out the reversible encryption is low.
- Unencrypted at Rest ranked as 3 due to possibility of the impact of risk will be less than others.
- Outdated Algorithm ranked as 2 due to the possibility of the impact of risk is high and by that the attacker can get access to the systems.
- Unencrypt in Transit ranked as 1 due to the attacker can get all the sensitive information such as, username and password.

All the ranking score were based on google search.

The risk ranking methodology I have used is => Risk = Likelihood * Impact



Section 4

Mitigation Plan

4.1 Employee Data Unencrypted at Rest

What is Your Recommended Mitigation Plan?

The plan to be recommended is to use transport data encryption.

Why Did you Recommend This Course of Action?

Transport data encryption use to encrypt SQL server and Azure SQL database data and files at Rest. So it's performe a real time input and output encryption and decryption to protect data at Rest.

4.2 Authentication Data Stored Using Reversible Encryption

What is Your Recommended Mitigation Plan?

The plan to be recommended is to use is Set store password using reversible encryption to disable and use salted hash.

Why Did you Recommend This Course of Action?

The Store password using reversible encryption provides support for applications that use protocols that require the user's password for authentication. Storing encrypted passwords in a way that is reversible means that the encrypted passwords can be decrypted. A knowledgeable attacker who able to break this encryption can then sign into network resources by using the compromised account. For this reason, never enable Store password using reversible encryption for all users in the domain unless application requirements outweigh the need to protect password information.

Using normal hashes could let the content vulnerable to dictionary attacks. On the other hand, salted hash can be used to set authentication data to protect the content from dictionary.

To use the salted hash, take password and a salt of your choice and append the salt to the password. by that, hash the salted password using hashing algorithm.

4.3 Authentication Requests are Not Encrypted in Transit

What is Your Recommended Mitigation Plan?

The plan to be recommended is to use End to end encryption.

Why Did you Recommend This Course of Action?

End to end encryption ensure the data is protected, not editable by the receiver. The data in transit is encoded to prevent others from read it. The data will be encrypted until reach its destination.

4.4 DES Algorithm in Use

What is Your Recommended Mitigation Plan?

*The plan to be recommended is to use **Advanced encryption standard (AES)**.*

Why Did you Recommend This Course of Action?

AES is based on the principle of substitution and permutation and use plain text of 128 to 256 bits., on other hand data encryption standard (DES) use plain text of 64-bits.

4.5 Security Audit

The audit team has been made aware of the systemic issue and wants to ensure your recommendations are followed. What steps can the audit team take?

- Create a policy that include establishment of transparent data encryption on the employee at Rest, set store password using reversible encryption, use salted hash, end to end encryption and use AES algorithm.

Optional Task:

Create an architecture diagram of a secure system.

Image of your secure architecture:

Optional Task (*Continued*):

Additional Steps Would You Recommend to Prevent the Attack as well as Future Issues: