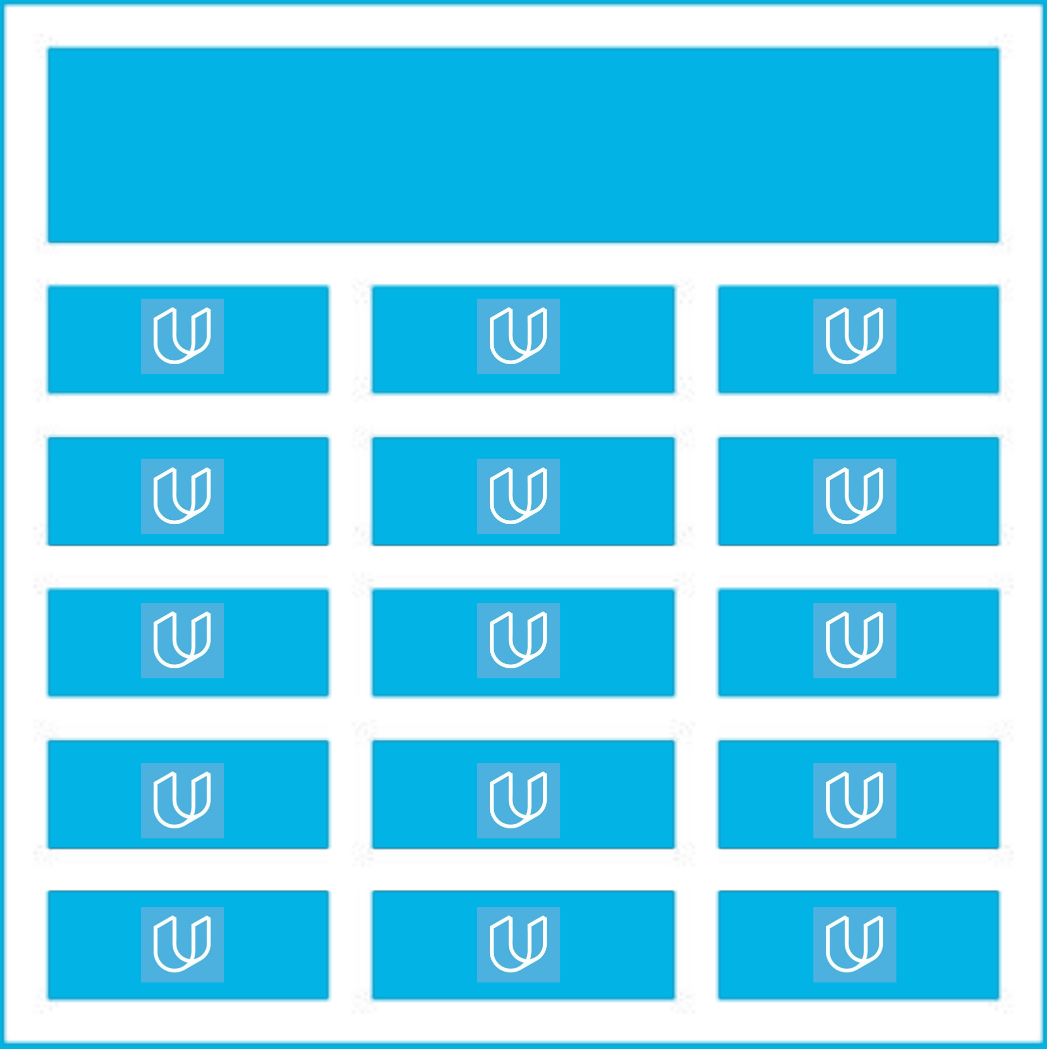
TimeSheets:
Threat Report



Zeina ElSharkawy

27/2/2025

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

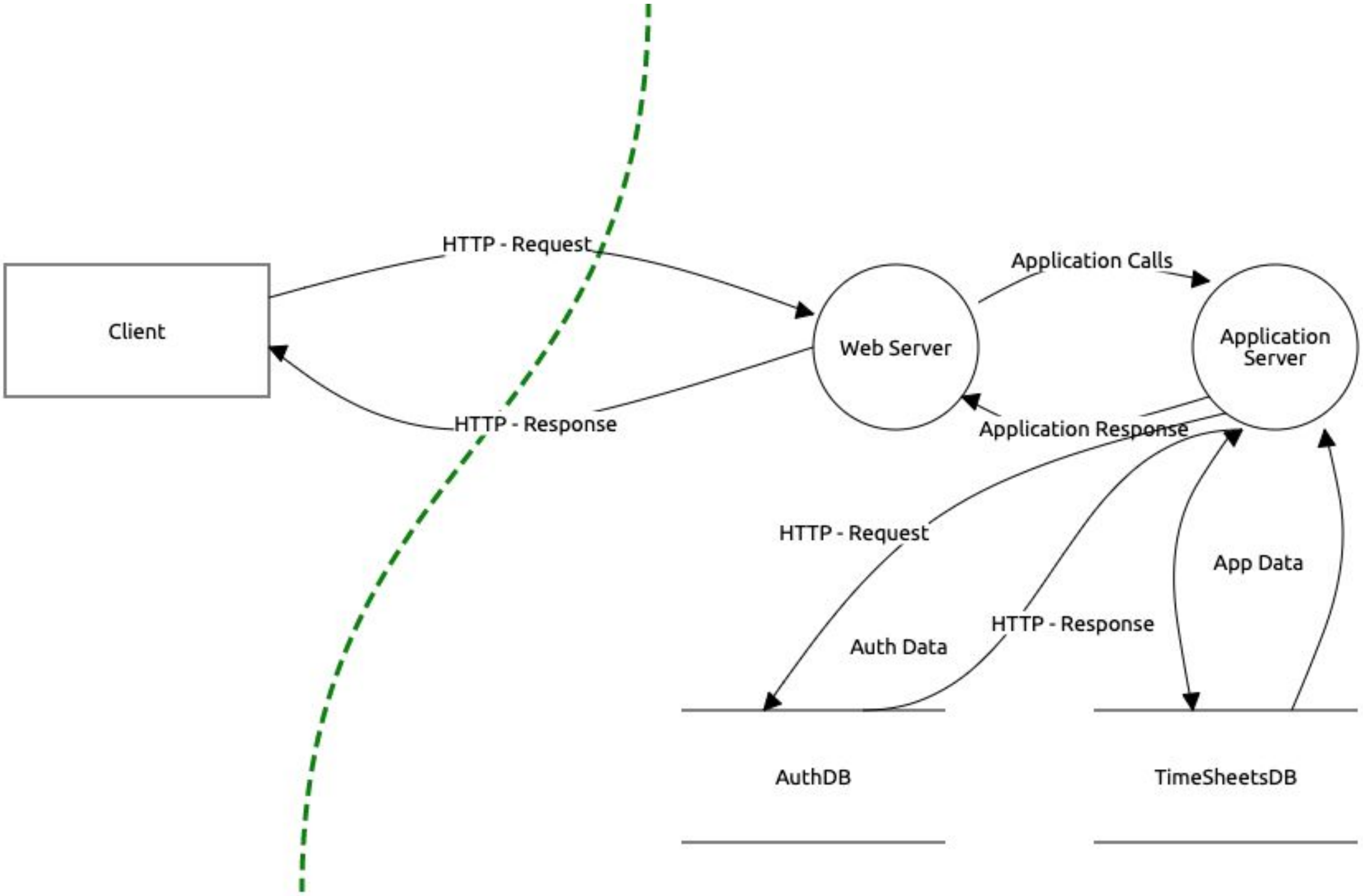
Request is generated from the client via the Internet. The request arrives at the TimeSheets web server which serves static content to the user (HTML, images, etc). Dynamic data is retrieved from the database and served to the client.

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 threat modeling, the Site Reliability Engineering (SRE) team confirmed that the database is on a server that does not have encryption at rest.

Why is this an issue?

[Your answer here]

The database on the server is considered data at rest which is valuable and can be exposed by hackers who might gain unauthorized access to this server so encryption at rest ensures that this data is protected against risks like data breach. Without encryption hackers can easily read and tamper this sensitive information leading to potential reputational (lack of trust) & financial damage to the TIMESHEETS organisation

2.2 Authentication Data Stored Using Reversible Encryption

Discovery:

During threat modeling, the Database Administrators (DBA) team confirmed that the database is storing authentication data (credentials) encrypted.

Why is this an issue?

[Your answer here]

Authentication data (Aka. credentials) must be deeply protected whereas encryption can be decrypted (reversible) so this exposes the data to hackers (not safe) who are able to decrypt this data and gain access to these credentials

2.3 Authentication Requests are Unencrypted in Transit

Discovery:

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

Why is this an issue?

[Your answer here]

This exposes the users’ authentication data (usernames & passwords) where hackers can intercept the request transmission and steal the data (MITM man in the middle attack/ eavesdropping)

2.4 DES Algorithm in Use

Discovery:

While conducting threat modeling, the security team identified sensitive data being stored using the Data Encryption Standard (DES) algorithm.

Why is this an issue?

[Your answer here]

DES is an encryption algorithm that became outdated with the improvement of computational power needed to break encryption algorithms as it became vulnerable to brute-force attacks(56-bit key is easily broken) so the sensitive data can be easily compromised and fall in the wrong hands affecting TIMESHEETS organisation reputation and finances.

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?

- *Weak or Missing Authentication/Authorization controls as there is no even obvious multifactor authentication*
- *No logging or monitoring (SIEM solution) so without logs detecting suspicious activity, investigating breaches, and performing forensic analysis becomes very difficult.*

Section 3

Risk Analysis

3.1 Scoring Risks

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

3.2 Risk Rationale

Why did you choose that ranking? Make sure to include your risk ranking methodology. Your explanations must be based on the learnings from this course as well as observations from the initial report. *(Did you use a tool or defined risk scoring system?)*

Likelihood x Impact = Risk Score

1. Unencrypted in Transit:

> Likelihood: High

Data in transit is exposed to interception (ex.man-in-the-middle attacks) as it is moving across networks.

> Impact: High

If it got intercepted sensitive information can be immediately compromised (accessed) and misused

3.2 Risk Rationale

Why did you choose that ranking? Make sure to include your risk ranking methodology. Your explanations must be based on the learnings from this course as well as observations from the initial report. *(Did you use a tool or defined risk scoring system?)*

Likelihood x Impact = Risk Score

2. Unencrypted at Rest:

> Likelihood: Moderate

Stored data is a bit protected/guarded by network & physical security measures so exploitation often requires more breach steps.

> Impact: High

However if accessed the damage can be strong as sensitive information may be exposed.

3. Reversible Encryption:

> Likelihood: Moderate

There is defense in depth provided by encryption but if the keys or the encryption mechanism are compromised the data might be revealed.

> Impact: Moderate

The data is somewhat protected until the reversible nature is exploited.

4. Outdated Algorithm:

> Likelihood: Lower

While outdated algorithms may have known weaknesses, they often still serve as a barrier unless there is a specific known exploit

> Impact: Lower

The damage isn't that strong compared to other vulnerabilities especially if there is fail safe/compensating measures.

Section 4

Mitigation Plan

4.1 Employee Data unencrypted at Rest

What is Your Recommended Mitigation Plan?

[Your recommended plan here]

Implement encryption:

- Encrypt the database and backups(if there are any)AES is the most widely adopted symmetric encryption algorithm for securing data at rest and uses keys of 128, 192, or 256 (strongest) bits

Key Management:

- Use a secure key management service to manage encryption keys ensuring they are rotated regularly and stored separately from the encrypted data.

Why Did you Recommend This Course of Action?

[Your justification here]

Encrypting data at rest protects sensitive information from unauthorized access and meeting compliance requirements helps avoid legal penalties and reputational damage. Also, it reduces the risk of data breaches and the potential consequences associated with them

4.2 Authentication Data Stored using Reversible Encryption

What is Your Recommended Mitigation Plan?

[Your recommended plan here]

Use One-way Hashing:

- use one-way hashing (e.g., bcrypt, scrypt, or Argon2).

Unique Salt for Each Password:

- Generate a unique salt per password to ensure that even identical passwords produce different hash outputs

Why Did you Recommend This Course of Action?

[Your justification here]

Hashing prevents the hacker from recovering of the original password in plaintext.Also, adding a unique salt to each password further defends against precomputed rainbow table attacks.This means that even if the hash is compromised, it is computationally impractical to reverse-engineer the original password.Overall, this reduces the risk of exposing sensitive credentials and aligns with modern security practices

4.3 Authentication Requests are Not Encrypted in Transit

What is Your Recommended Mitigation Plan?

[Your recommended plan here]

Implement TLS

- ensures that data sent between clients and servers is encrypted and secure from interception by:
  - Handshake Process(client and server agree on encryption parameters)
  - Symmetric Encryption
  - Ensures data integrity

Use HTTPS instead of HTTP

Why Did you Recommend This Course of Action?

[Your justification here]

Enforcing TLS encrypts the entire authentication transaction ensuring that credentials remain confidential and that the data cannot be altered in transit. Also, it mitigates the risk of eavesdropping and man-in-the-middle attacks by ensuring that even if network traffic is intercepted the data remains encrypted

4.4 DES Algorithm in Use

What is Your Recommended Mitigation Plan?

[Your recommended plan here]

Update to a stronger algorithm (AES is widely accepted and recommended by global standards and frameworks)

Why Did you Recommend This Course of Action?

[Your justification here]

AES has longer key lengths which makes it more resistant to brute-force attacks compared to DES which uses a 56-bit key which is far too short by today's standards against the current computational power

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?

[Your answer here]

Create a framework:

- Write Guidelines: Create clear guidelines highlighting the necessary actions for implementing encryption , TLS & hashing following best practice standards

Schedule meetings:

- Organise follow-up meetings to review progress on implementing recommendations

Perform Testing and Continuous Monitoring:

- Penetration testing & vulnerability scanning to assess current security practices & mitigation plan

Training and Awareness:

- Organize training sessions regularly on new security practices and the importance of compliance to ensure they understand and are adhering to the updated security standards.

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: