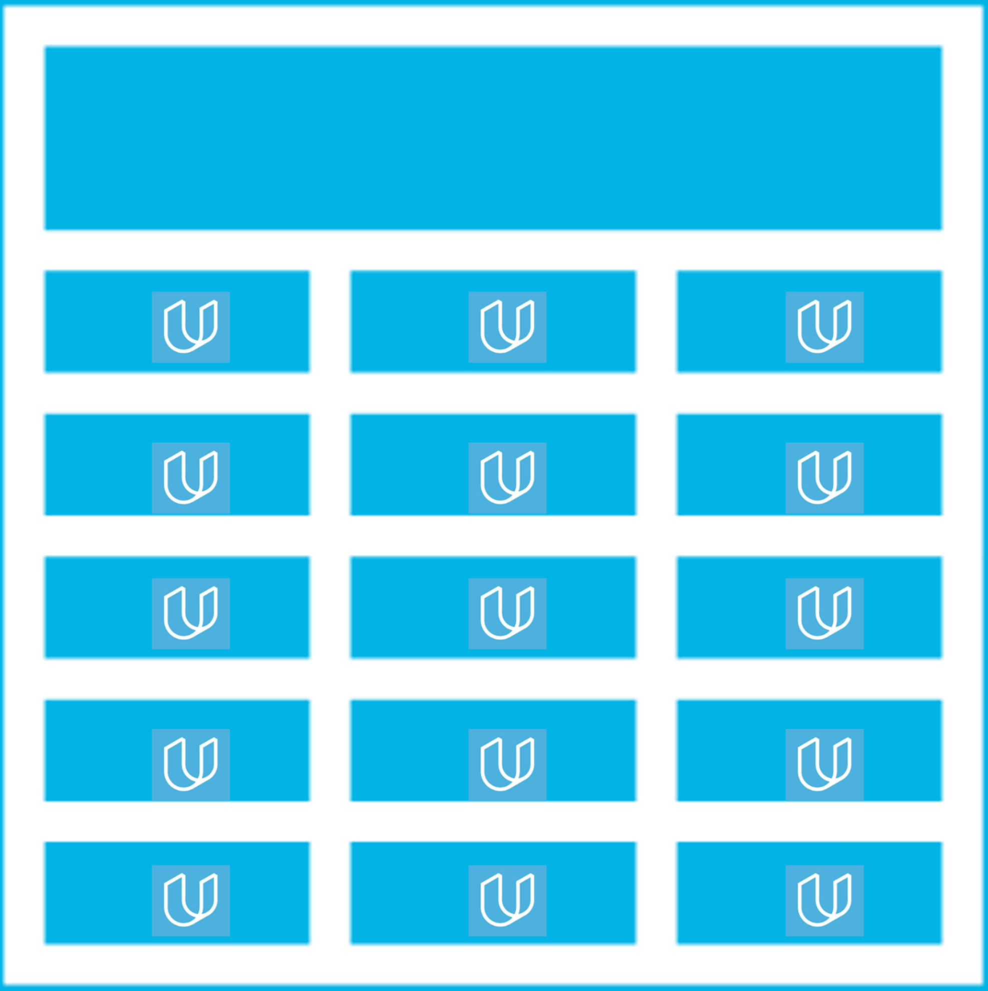
TimeSheets: Threat Report



Oluwamayowa Olawumi

27/06/2023

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

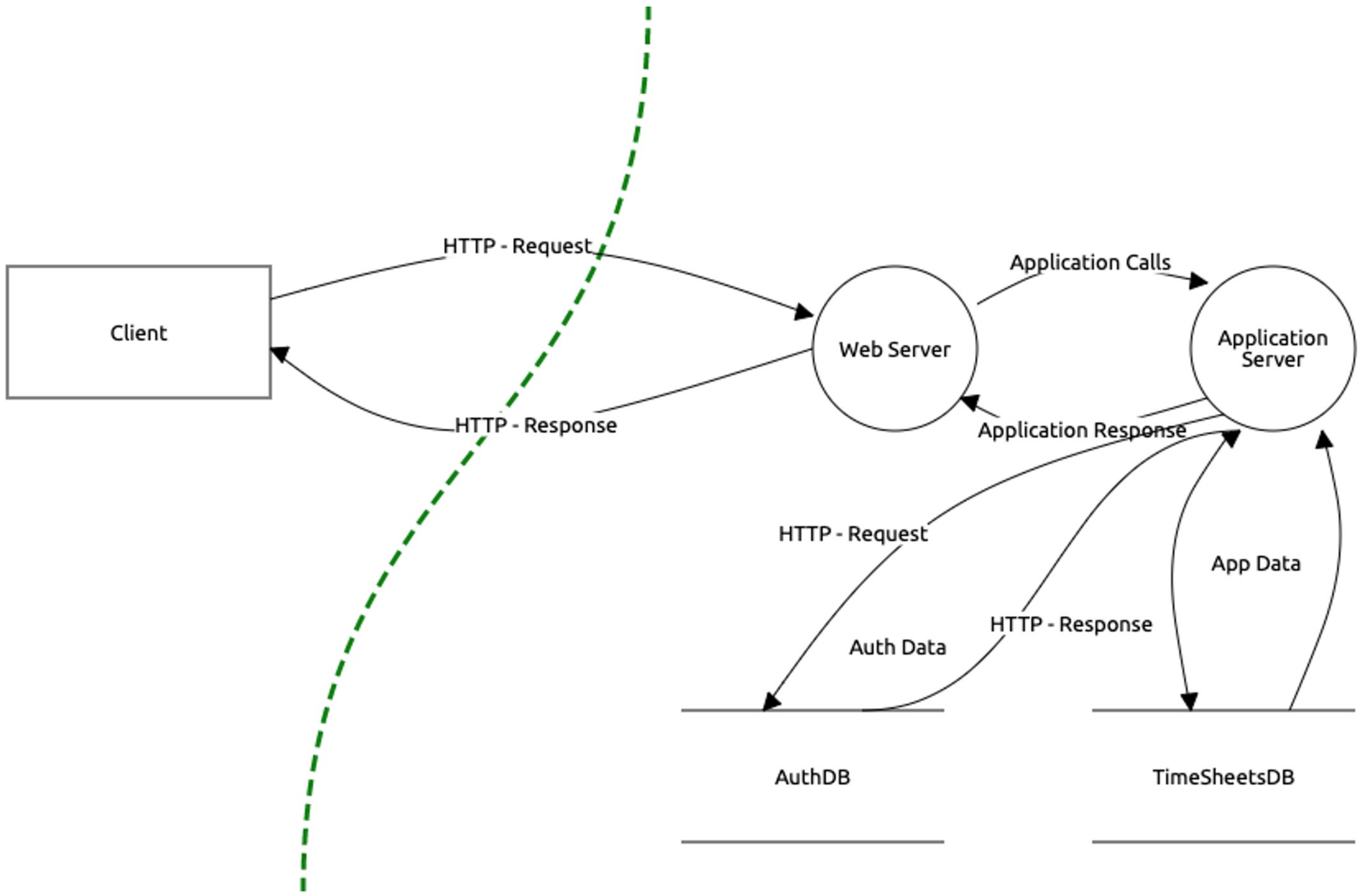
Request is generated from the client via the Internet. The request arrives at the TimeSheets web server which serves static content to the user (HTML, images, etc). Dynamic data is retrieved from the database and served to the client.

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?

Data at rest is a prime target for malicious actors. Unlike individual data packets in motion across a network, static data storage often exhibits a discernible structure and meaningful file names. Moreover, data at rest commonly encompasses an organization's most sensitive and valuable information, including:

- Financial documents (e.g., transaction records, bank account details, credit card numbers).
- Intellectual property (e.g., product information, business plans, schematics, code).
- Contact information.
- Marketing data (e.g., user interactions, strategies, directions, leads).
- Personal information of employees and customers.

Failure to encrypt such data grants hackers easy access to valuable assets.

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?

Storing encrypted passwords in a way that is reversible means that the encrypted passwords can be decrypted. A knowledgeable attacker who is able to break this encryption can then sign in to network resources by using the compromised account.

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?

In-transit data are more vulnerable than at-rest data as you cannot reliably prevent eavesdropping when sending messages over the Internet.

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?

It is fundamentally inadequate because its 56-bit key is too short. It is vulnerable to brute-force search of the whole key space, either by large collections of general-purpose machines or even more quickly by specialized hardware.

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?

The use of the HTTP request/response.

Absence of firewalls to shield internal network from incoming traffic.

HTTPS should be utilized. This protects against man-in-the-middle attacks and the confidentiality of data sent between the browser and the website.

Firewalls should be set up between the client and internal networks to shield our network from malicious external traffic.

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. *(Did you use a tool or defined risk scoring system?)*

Utilizing the risk assessment formula, where Risk = Likelihood x Impact, to evaluate each risk and determine the most concerning ones with the highest impact on the business.

Unencrypted in Transit: This risk holds the top ranking (number 1) due to the potential for unauthorized individuals to intercept and access data transmitted over unencrypted channels. This vulnerability can result in data breaches, unauthorized access to sensitive information, and the potential for data manipulation or misuse.

Unencrypted at Rest: Ranked as number 2, this risk acknowledges that while data at rest may not be as immediately susceptible to interception as data in transit, unencrypted data stored on physical or digital storage devices remains vulnerable to theft or unauthorized access. Should the data fall into the wrong hands, privacy breaches and significant harm could ensue.

Reversible Encryption: Ranked number 3, this risk considers the lessened security of reversible encryption, where the encryption algorithm used can be reversed or decrypted. If an attacker gains access to the encrypted data and successfully reverses the encryption, the sensitive information contained within could be compromised.

Outdated Algorithm: Ranked number 4, this risk acknowledges that using outdated algorithms may not directly expose data or communication channels but still poses security risks. Outdated algorithms are more susceptible to cryptographic attacks and vulnerabilities, thus diminishing the overall system security. It is advisable to employ up-to-date, widely recognized cryptographic algorithms that have undergone rigorous scrutiny and testing.

Section 4

Mitigation Plan

4.1 Employee Data Unencrypted at Rest

What is Your Recommended Mitigation Plan?
Apply encryption to sensitive data stored in the server. This can be achieved through disk encryption, database encryption, or file-level encryption.

Regularly review and audit data storage systems to identify and address any vulnerabilities or misconfigurations.

Why Did you Recommend This Course of Action?

Ensures privacy of critical data.

Limits the impact of data breaches.

Prevents ransomware blackmail.

4.2 Authentication Data Stored Using Reversible Encryption

What is Your Recommended Mitigation Plan?

Use hashing for encryption.

Adapt the use of salt in the hashing technique.

Conduct security assessments and penetration testing to identify any weaknesses in the encryption implementation.

Why Did you Recommend This Course of Action?

Hashing is practically irreversible.

Salting renders rainbow tables useless.

Penetration testing simulates real-world attack scenarios to evaluate the effectiveness of the encryption implementation. It helps uncover potential attack vectors and provides insights into how an attacker might exploit vulnerabilities in the system.

4.3 Authentication Requests are Not Encrypted in Transit

What is Your Recommended Mitigation Plan?

Implement Transport Layer Security (TLS) or Secure Sockets Layer (SSL) protocols to encrypt data during transit

Train employees on the importance of using encrypted communication channels and the potential risks associated with transmitting data over unsecured networks.

Why Did you Recommend This Course of Action?

Implementing TLS or SSL protocols enhances the security of data transmitted over networks by encrypting it, significantly impeding unauthorized individuals from intercepting or accessing the data. Encryption acts as a robust safeguard against data breaches and unauthorized access.

Raising employee awareness about the significance of using encrypted communication channels and the associated risks when transmitting data over unsecured networks fosters a culture of security consciousness. Employees gain a heightened understanding of the importance of prioritizing secure communication methods and refraining from transmitting sensitive information over unencrypted channels, thereby mitigating the risk of data compromise.

4.4 DES Algorithm in Use

What is Your Recommended Mitigation Plan?

Keep encryption systems, libraries, and software components up to date by adopting widely recognized and current cryptographic algorithms.

Stay knowledgeable about best cryptographic practices and emerging vulnerabilities.

Consistently evaluate and review the cryptographic algorithms employed within the organization to identify and replace any outdated or weak algorithms.

Collaborate with industry experts and cryptographic communities to ensure compliance with recommended cryptographic standards.

Why Did you Recommend This Course of Action?

Maintaining up-to-date encryption systems, libraries, and software components is crucial to employing the most secure and reliable cryptographic algorithms. These algorithms have undergone rigorous evaluation, testing, and scrutiny by the cryptographic community, minimizing their vulnerability to attacks and exploits.

Remaining informed about cryptographic best practices and emerging vulnerabilities enables organizations to proactively identify and mitigate potential weaknesses in their encryption systems.

Regularly reviewing and assessing the cryptographic algorithms used within the organization facilitates proactive risk management. Outdated or weak algorithms can be identified and replaced with stronger alternatives, reducing the likelihood of cryptographic attacks and data breaches.

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?

Develop an Audit Plan: Create a comprehensive audit plan detailing the audit's objectives, scope, and methodology. Emphasize verifying the implementation of recommended security measures, including encryption, robust algorithms, access controls, and regular reviews.

Conduct On-Site Assessments: Perform on-site assessments to directly evaluate organizational practices. Review documentation, interview relevant personnel, and assess the implementation of security measures across various systems and processes.

Review Security Policies and Procedures: Evaluate the organization's security policies and procedures to ensure alignment with recommended security measures. This entails scrutinizing encryption, access controls, key management, data storage, and audit-related policies.

Perform Technical Assessments: Execute technical assessments such as vulnerability scans and penetration testing to assess the effectiveness of implemented security controls.

Optional Task *(Continued)*:

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

- Apply Security Updates and Patches: Ensure regular updates and patches are applied to software, operating systems, and applications to guard against known vulnerabilities. Establish a systematic process for managing patches to promptly deploy security updates, reducing the risk of exploitation.
- Implement Multi-Factor Authentication (MFA): Utilize multi-factor authentication for user accounts, particularly those with privileged access or access to sensitive data. MFA strengthens security by requiring users to provide additional authentication factors in addition to their username and password.
- Enforce Least Privilege Principle (LPP): Restrict user access privileges to the minimum necessary for their assigned roles and responsibilities. Adhering to the principle of least privilege mitigates potential harm resulting from compromised accounts or internal threats.