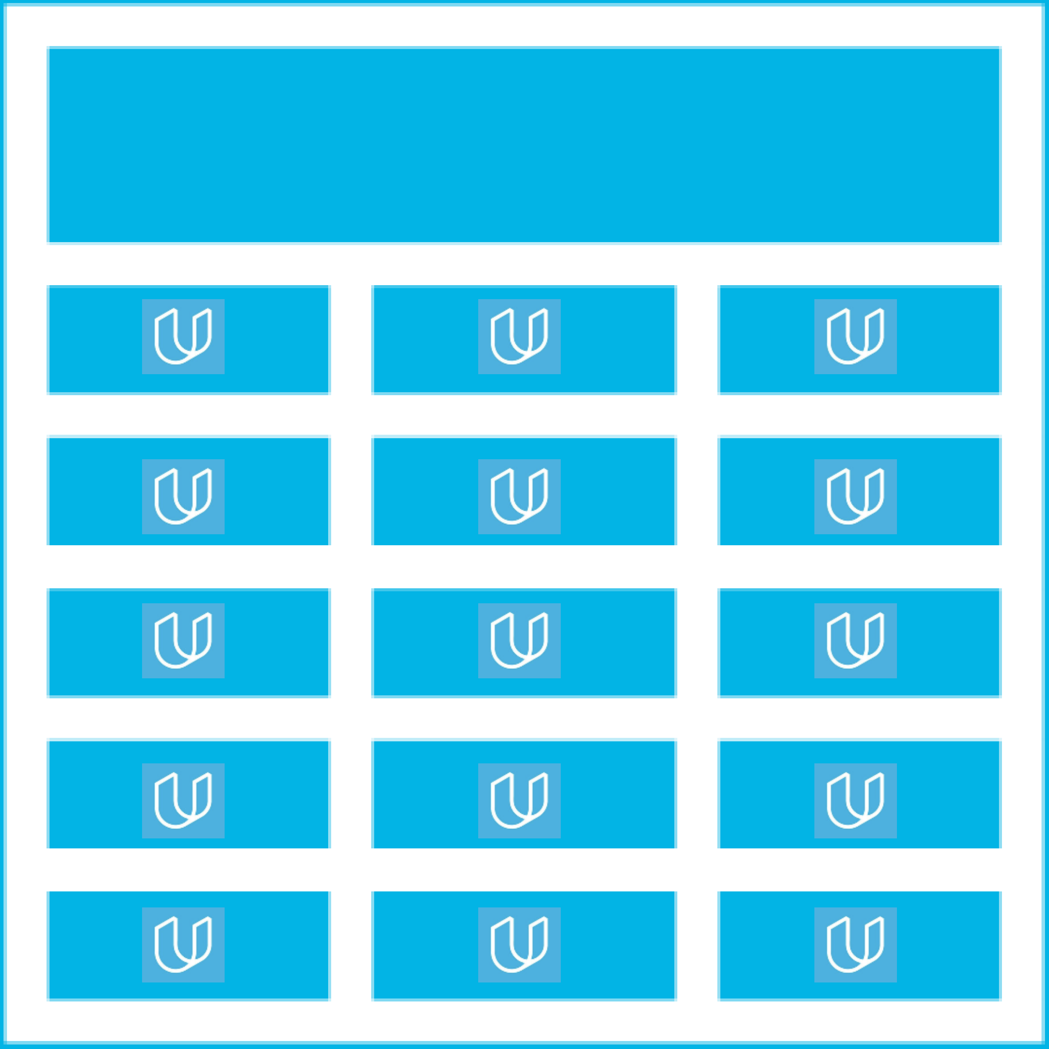
TimeSheets:

Threat Report



Mostafa Saber

23/10/2022

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

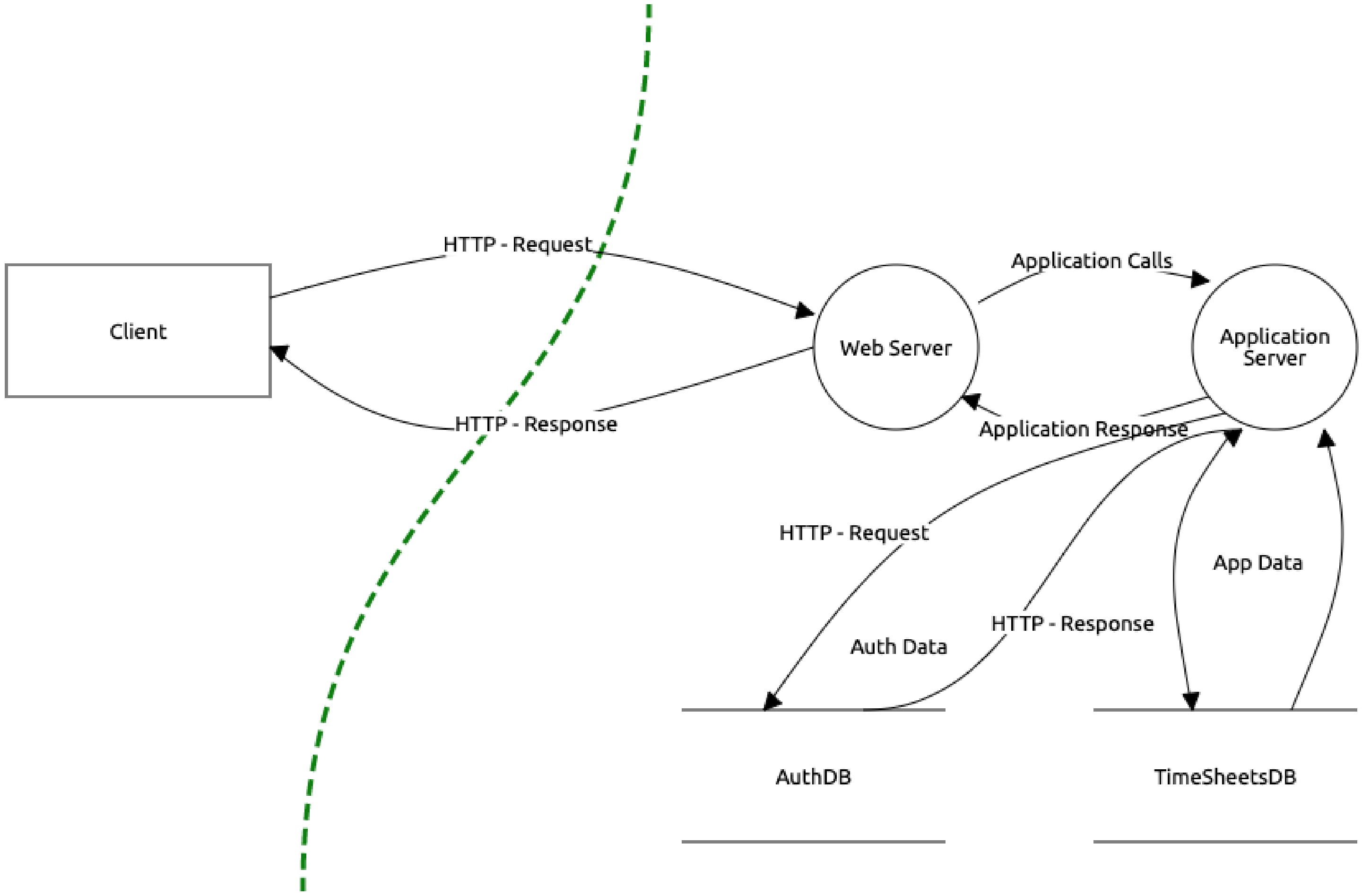
Request is generated from the client via the Internet. The request arrives at the TimeSheets web server which serves static content to the user (HTML, images, etc). Dynamic data is retrieved from the database and served to the client.

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 CIA (Confidentiality Integrity Availability) triad is the foundational principle for the cybersecurity world, and breaking at least one of these principles can damage the whole security of the system.

The broken pillar here is the Confidentiality. Confidentiality can be achieved through encryption of the data. When data is not private anymore it opens a door for a vulnerability that the threat actor can exploit. Because if the data got breached by any means it will be visible as a plain text for the threat actor to exploit.

So by following the security principle of confidentiality and encrypting the data, the potential impact will be low, as the threat actor wont be able to use the data since it is encrypted even after a breach.

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?

Authentication is the first line of defense against the threat actors, therefore its authentication data (credentials) are extremely sensitive and important, and using encryption opens a vulnerability here.

Because if the DB got compromised and since the encryption is a two-way method it can get decrypted if the threat actor got his hands on the encryption key.

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?

Data should never be transmited externally or even in some cases internaly as a plain-text without encryption as this violates the confidentaility principle and opens a vulnerability for threat actors to exploit.

Since the data is not encrypted at transit, a MitM(Man in the Middle) attack can exploit such vulnerability by sniffing the inbound and outbound traffic of the system, and thus, gain authority of a very sensitive data.

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?

According to the NIST SP 800-175B Cryptographic Standards and guidelines, the best strong symmetric algorithms to use right now is the AES (Advanced Encryption Standard) and TDEA (Triple Data Encryption Algorithm). So using an outdated encryption algorithm like the DES (Data Encryption Standards) to encrypt sensitive data definitely opens a vulnerability for the threat actors to exploit.

If the data got breached or compromised, it can be decrypted since it's a weak encryption algorithm and the data can get exploited and utilized by the threat actors.

"due to advances in computer power and speeds, the strength of the DES algorithm is no longer sufficient to adequately protect Federal Government information. Therefore, DES was withdrawn as an approved algorithm in 2005 (i.e., the use of DES is no longer approved for encryption or otherwise applying cryptographic protection)" [NIST SP 800-17B]

Optional Task:

Examine the threat model diagram from Section 1 and answer:

What non-encryption issues can you identify?

- No Filtering of Traffic at the system edge.
- No Input verification and Output sanitization
- No Backup server
- No implementation of SSL/TLS
- No authorization service for clients and admins

What recommendation would you give to solve those issues?

- WAF (Web Application Firewall)
- Apply input verification and output sanitization
- Add a backup server, and backup regularly
- Implement the PKI and certificates
- Implement authorization service

Optional Task *(Continued)*:

Why do you recommend those solutions?

- WAF will help filter traffic from suspicions locations and IPs
- Input verification is important to avoid web application vulnerabilities like SQL Injection and XSS, also sanitizing outputs from the system to avoid giving sensitive information for threat actors to use
- Having a backup server will help in achieving Availability for the system, so that if the system got compromised, the data is still available and safe.
- Implementing the SSL/TLS protocol avoids falling victim to spoofing attack and trust the communicating entity.
- Implementing authorization service to separate user privilege from admin privilege

Section 3

Risk Analysis

3.1 Scoring Risks

Risk	**Score** ***(1 is most dangerous, 4 is least dangerous)***
Unencrypted at Rest	4
Reversible Encryption	3
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.

RISK = LIKELIHOOD X IMPACT is being used as the methodology.

- **Unencrypted in Transit –**

  - **Risk:** 1

  - **Method:** HIGH X HIGH = HIGH

  - **Reason:** This risk is the most severe because data being transferred unencrypted can make it fall victim to sniffing, and the threat actor will be able to see the sensitive data in plain text and utilize it. This data can be the credentials of a user or even an admin, and getting these credentials will make the threat actor have access to the whole system where he then can breach the rest of the data and even modify its content.

  - **Comparison:** For the other risks, the threat actor first needs to breach the data which is not easy and then try to unencrypt the data, which is a long process that the security team might have the time to respond to the incident and protect the data.

3.2 Risk Rationale *(Continued)*:

- **Outdated Algorithm -**
  - **Risk: 2**
  - **Method:** MEDIUM X HIGH = HIGH
  - **Reason:** The **likelihood** of this risk is not high, because the threat actor first needs to compromise the system and leak out the sensitive data before decrypting it. Though its **impact** is extremely high, because once the threat actor decrypts the sensitive data (Due to the weak outdated algorithm), the threat actor will have access to systems credentials and utilizes it.
  - **Comparison:** Compared to the **Reversable Encryption Risk,** the **Outdated Algorithm Risk** is actually more severe, because even if the sensitive data got breached, if the sensitive data was using a high encryption algorithm the threat actor wont be able to utilize it and it will be useless to him. Additionally, compared to the **Unencrypted at Rest Risk**, although it depends on the situation and the value of the data, but in our case, the data at rest is not as sensitive as the login credentials that the threat actor will target, the threat actor will most likely target the first line of defense which is the authentication data, therefore its more severe.

3.2 Risk Rationale *(Continued)*:

- **Reversible Encryption -**
  - **Risk: 3**
  - **Method:** LOW X HIGH = MEDIUM
  - **Reason:** Its not easy to breach data from the internal system and even if the data got breached if the algorithm being used is strong, it will be impossible to reverse it that's why the **Likelihood** is low. On the other hand, if the threat actor was able to breach the data and gain the key to reverse encrypt it, the **Impact** will be extremely high as the sensitive data will be utilized by the threat actor.
  - **Comparison:** In comparison with the **Unencrypted at rest Risk**, as said before it depends on the type of system and data, and in our case the data at rest is not as sensitive as the authentication data, its just merely the recorded attendance of the employees, which makes it lower in risk than **the Reversible Encryption Risk** of a sensitive data.

3.2 Risk Rationale *(Continued)*:

- **Unencrypted at rest -**
    - **Risk: 4**
    - **Method:** LOW X LOW = LOW
    - **Reason:** The reason the **Likelihood** is low is that the data might not be of great value to the threat actor. Additionally, its hard to breach and compromise server and database without prior knowledge of the type of database or server used. Moreover, this data is not as active and regularly being transmitted outside the system like the authentication data. The **Impact** is low because even if the data got breached, the threat actor will find it useless as its of low value.
    - **Comparison:** The comparison has been mentioned in the other risks, but essentially, the data here is of low value and the breach is not likely to happen as its not easy, therefore the other risks hold more severity.

Section 4

Mitigation Plan

4.1 Employee Data Unencrypted at Rest

What is Your Recommended Mitigation Plan?

Encrypt the Data.

Why Did you Recommend This Course of Action?

Data must be encrypted so that in the future even if a breach would occur, the threat actor wont be able to utilize the data as it will be encrypted.

Additionally, In order to achieve confidentiality, the data must be encrypted to gain privacy.

4.2 Authentication Data Stored Using Reversible Encryption

What is Your Recommended Mitigation Plan?

Store using Hash algorithms and add salt.

Why Did you Recommend This Course of Action?

Hash is a one-way function that is unreversible unlike encryption, so if the threat actor would breach the data he wont be able to reverse it to know its content.

Additionaly, adding salt will prevent the threat actor from using the Rainbow table to know the plain text of the hash.

4.3 Authentication Requests are Not Encrypted in Transit

What is Your Recommended Mitigation Plan?

Apply the SSL/TLS protocol, HTTPS, certificates.

Why Did you Recommend This Course of Action?

When data is encrypted in transit, even if the threat actor would sniff the data he wont be able to utilize it since its encrypted.

Additionally, applying the certificates feature will not only allow you to trust the entity you are communicated with, but will also make the MitM attack uneffective as the only way for the threat actor to hijak the session is for him to gain the Public key and change it with his own public key which Is not possible with out the private key of the CA, therefore, the threat actor wont be able to spoof the system or the client, making the transmittion safe and private.

4.4 DES Algorithm in Use

What is Your Recommended Mitigation Plan?

Use the AES (Advanced Encryption Standard) or TDEA (Triple Data Encryption Algorithm).

Why Did you Recommend This Course of Action?

The AES (Advanced Encryption Standard) and TDEA (Triple Data Encryption Algorithm) are the best algorithms to use for encryption according to NIST. So even if the data got breached, it will be impossible for the threat actor to decrypt it as the computational power and speed is not advanced enough to do that.

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?

- A set of controls will be provided to the auditors to perform activities on it.

- Proof for the set of controls will be provided by the policies and standards.

  - If the policy does not exit, CISO will choose a framework suitable for the system and the data being used and handled.

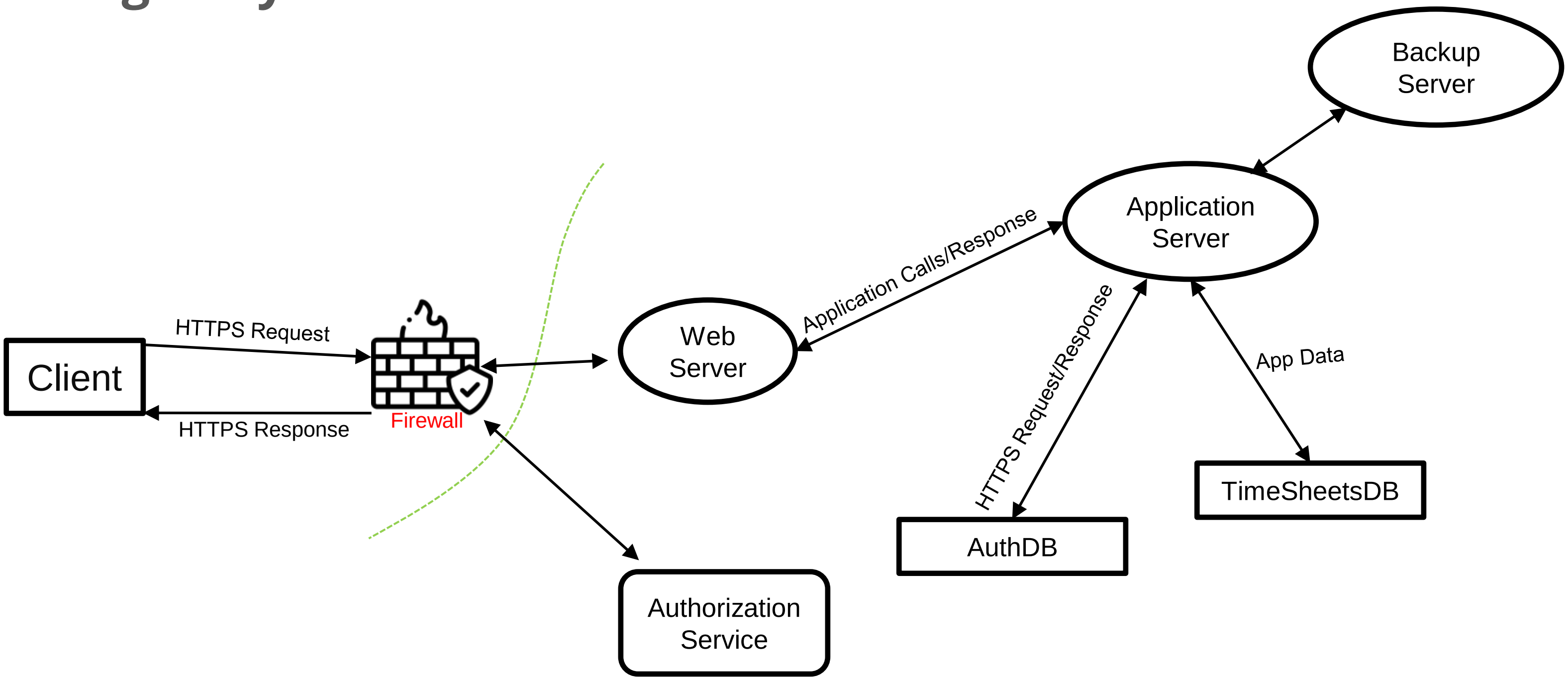
- Demonstration to the auditors will be done by providing documentation of how the encryption is done and what type is being used and how it is complied.

- Auditors can take random samples of the data to make sure its encrypted properly and make test to the system to ensure the data is being encrypted in transit using the appropriate tools.

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:

- Add IDS (Intrusion Detection System) and IPS (Intrusion Prevention System)

- Implement Zero Trust Policy

- Add IOCs (Indicator of Compromise) to the security appliances and up to date from the Threat Intelligence parties.

- Add multifactor authentication

- Regular vulnerability assessment to be made

- Patching up to date

- Add AAA server

- Control Audits to be done to ensure compliance of policy