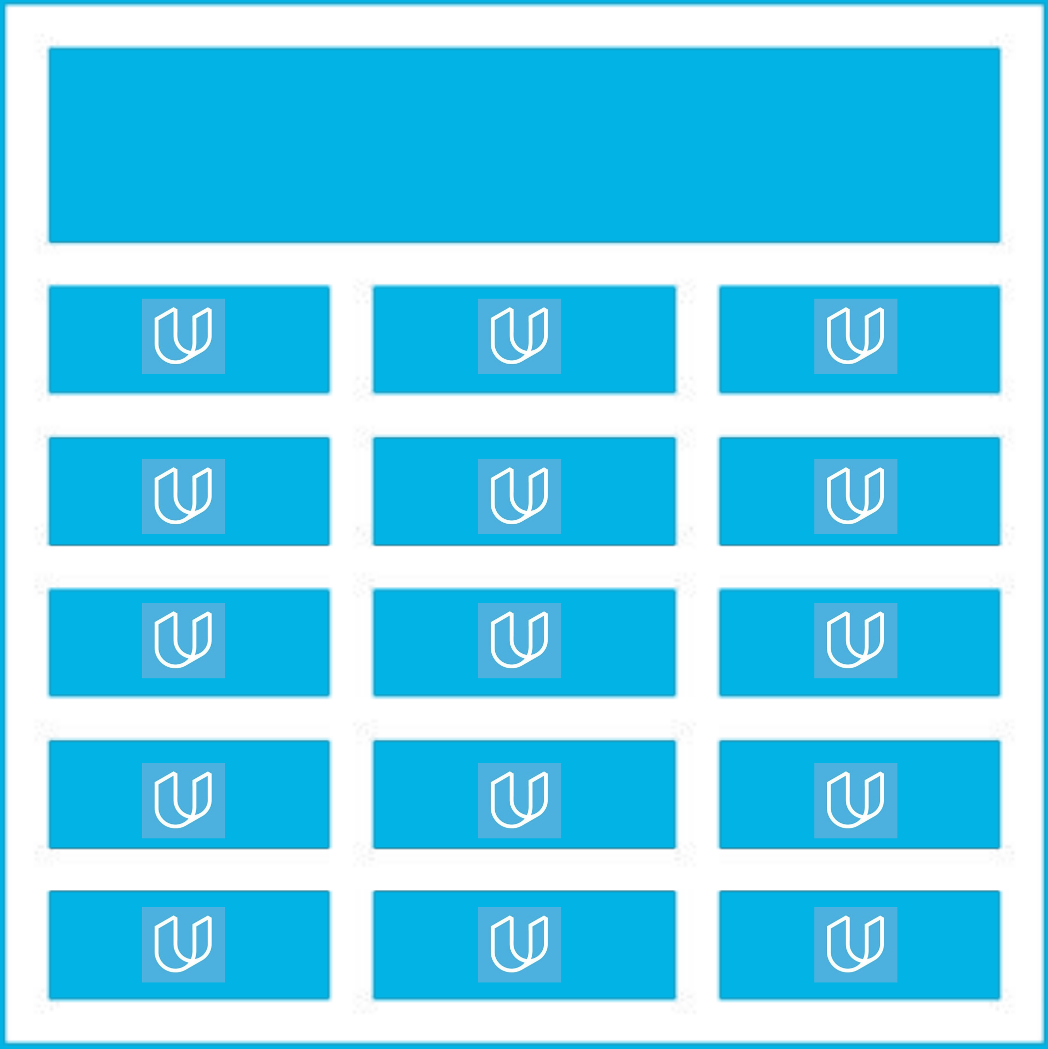
TimeSheets:

Threat Report



Ahsan Khan

02/19/2021

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

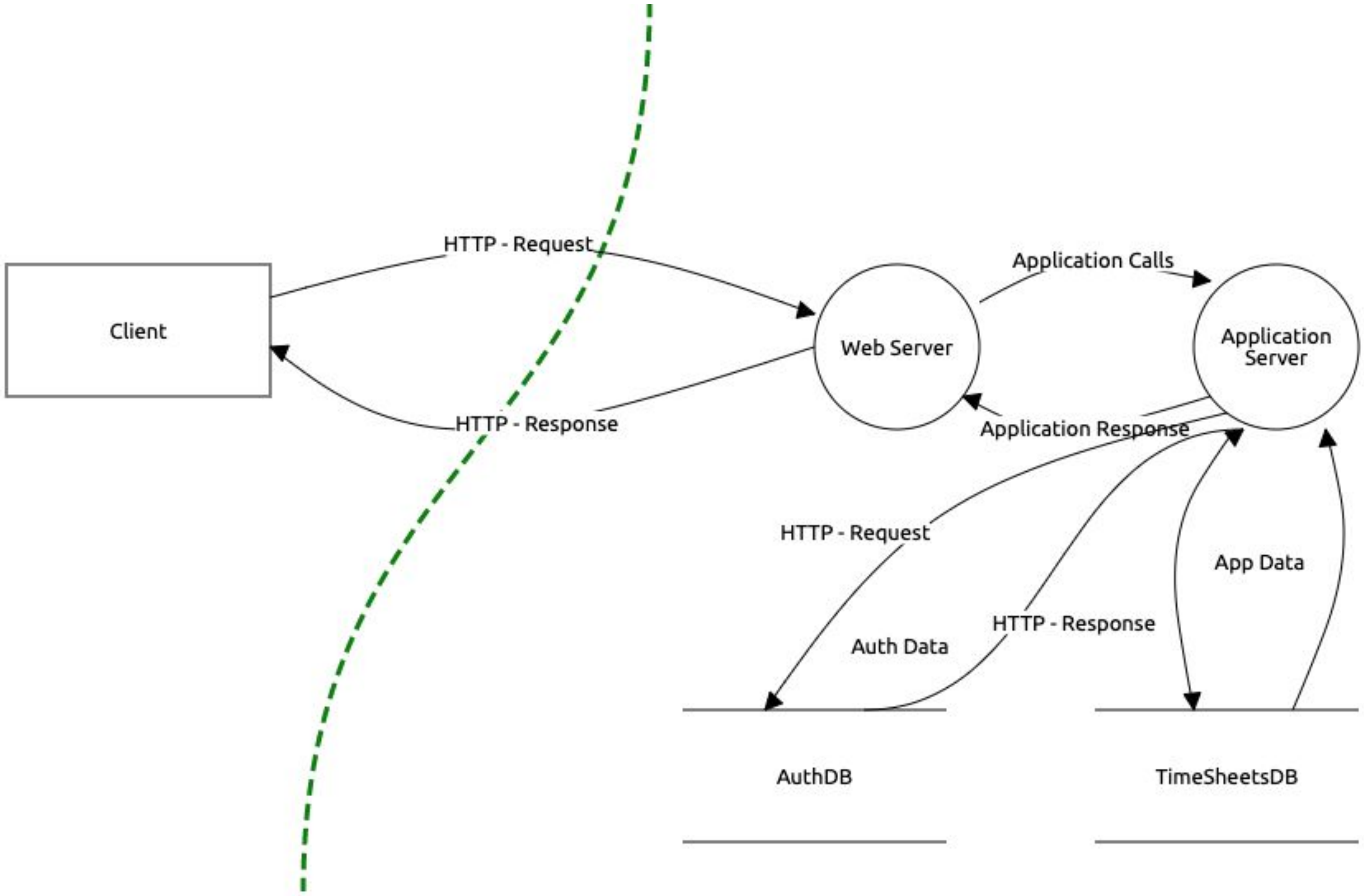
Request is generated from the client via the Internet. The request arrives at the TimeSheets web server which serves static content to the user (HTML, images, etc). Dynamic data is retrieved from the database and served to the client.

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 encrypted at Rest using AES-256 encryption
- Authentication data is using HASHING protocol
- Authentication requests are encrypted in transit
- Sensitive data is encrypted using AES-256 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?

To understand the issue better, it would be helpful to define the meaning of *‘encryption at rest’,* first. **Encryption at rest** means that the data the application stores on its local storage is encrypted, so that an attacker who can access the storage but not the application itself can’t read the data.

It is very understandable now that why we consider this an issue. If your system gets compromised, the least you can do to keep the user detail and credential safe is by encrypting the data that is being stored on the database. If the data is not encrypted, anybody who can access the database, can either steal the data or manipulate it for their good.

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?

According to Microsoft, storing encrypted authentication data in a way that is reversible means that the encrypted passwords can be decrypted. A knowledgeable attack who is able to break this encryption can then log on to network resources by using the compromised accounts.

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?

MAN IN THE MIDDLE ATTACK: It is an attack when an attacker intercepts communications between two parties either to secretly eavesdrop or modify traffic traveling between the two. Attackers may use this attack to steal login credentials or personal information, spy on the victim, or sabotage communications or corrupt data. With that being said, if the authentication requests are unencrypted in transit, an attacker using the method of MITM attack can easily see your login credentials in the plaintext, and that can be used to log in to your accounts or steal your 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?

DES algorithm has a block size of 58 bits (+8 parity bits). It is proven by experts and professionals that 58 bits is a very short key, which can be cracked in a matter of days or months or a year. There is no flaw in the algorithm, but the advancement in technology has shown that DES algorithm can be cracked very easily and that's why it is not considered safe anymore.

Optional Task:

Examine the threat model diagram from Section 1 and answer:

What non-encryption issues can you identify?

- Employee Data Unencrypted at Rest

- Authentication requests are not encrypted in transit

What recommendation would you give to solve those issues?

I would suggest the company to AES algorithm to encrypt the data that is being rested on the database servers, this way if the networks get compromised, at least the data on their database would be safe and encrypted. Secondly, I would suggest the company to encrypt the credentials using hash methodology, this way the man in the middle attacker won't be able to see the credentials in the plaintext.

Why do you recommend those solutions?

AES algorithm is the strongest as of now, it may take decades to break this encryption, that's why it is a good practice to encrypt the data using AES. Hashing turns your password (or any other piece of data) into a short string of letters and/or numbers using an encryption algorithm. If a website is hacked, the hackers don't get access to your password. Instead, they just get access to the encrypted "hash" created by your password

Section 3

Risk Analysis

3.1 Scoring Risks

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

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?)*

According to CVSS(Common Vulnerability Scoring System), to describe and rank the risk, we will need to keep some aspects in our mind (CIA triad). For example, availability(if the system is up and running), confidentiality(if the data/personal information is secured), integrity (accuracy of the data being requested).

Unencrypted in Transit: Unencrypted data in transit is not keeping our data confidential, it is visible in the plaintext to the man-in-the-middle attacker. The attacker has the potential to modify the data we are requesting from the server, this way we do not get the expected results. It also has the potential to make the system unavailable to us, by manipulating with our authentication credentials. That's why, this is the most dangerous out of all 4 of the risk.

Unencrypted at Rest: Unencrypted data at rest breaks the law of confidentiality and integrity. An attacker can steal our data and also also manipulate with our information.

Outdated Algorithm & Reversible Encryption: Both of these risks compromises the law of confidentiality. An outdated algorithm (like DES) can easily be breakable by a knowledgeable attacker, and can compromise the user's data by stealing it. Same goes for the reversible encryption; if it is being used for the authentication security, then a compromised key can easily decrypt our authentication credentials, and can reveal the login info in the plaintext, to the attacker.

Section 4

Mitigation Plan

4.1 Employee Data Unencrypted at Rest

What is Your Recommended Mitigation Plan?

Use AES-256 encryption to encrypt the unencrypted data at Rest

Why Did you Recommend This Course of Action?

According to NIST (Cybersecurity Framework), AES-256 is the best encryption method to encrypt the data. AES uses symmetric key encryption, which involves the use of only one secret key to cipher and decipher information. AES is the first and only publicly accessible cipher approved by the US National Security Agency for protecting top secret information. Leaving data unencrypted at rest is just like depositing gold in the security locker and not taking the key out of the locker. Encrypting data at rest protects the organization from physical theft of the file system storage devices.

Most modern operating systems (like Linux or Windows Server) provide the capability to encrypt their disks in their entirety. This is accomplished with symmetric encryption whereby there is a key or passphrase that a computer operator has to enter when the disks are encrypted and when the system boots to allow access to the data.
Encrypting data at rest can protect the organization from unauthorized access to data when computer hardware is sent for repair or discarded. Encrypting data at rest can help to satisfy information security or regulatory requirements such as the Payment Card Industry Data Security Standard (PCI DSS) or the Health Insurance Portability and Accountability Act (HIPAA).

4.2 Authentication Data Stored Using Reversible Encryption

What is Your Recommended Mitigation Plan?

Use HASHING SHA-1 and use salt with the hashing methods.

Why Did you Recommend This Course of Action?

Encryption is reversible, which means if a bad guy finds out the key that was used to encrypt the password, that bad guy can easily decrypt the authentication credentials. The best industry practice is to use SHA-1 Hashing with salt, to secure the stored authentication data. Hashing is irreversible, which means, even if you find out the hash value, it is computationally impossible to crack the hash value and get the password in plaintext. A thing called rainbow table is a precomputed database of hash values and their corresponding plaintext passwords. Anyone who has access to this rainbow table, they can easily crack your hashed password. That's why it is recommended to use salt in your hashed value. Salt adds a unique value to the beginning or end of a value you intend to hash, with that being said, it gets impossible for the rainbow database to find the corresponding plaintext if the hashed is being encrypted with salt.

4.3 Authentication Requests are Not Encrypted in Transit

What is Your Recommended Mitigation Plan?

Enable Secure Sockets Layer (SSL)

Website certificate

Why Did you Recommend This Course of Action?

It uses a hashing algorithm to ensure data is not altered in transit. This can help defeat man-in-the-middle attacks, as they act of decrypting and re-encrypting data allows an attacker to alter the signature but not change the key data. A hacker can act like man in the middle by connecting to your wireless network. We call it man in the middle attack, because they are sitting between you and the server. Whatever request you send to the server, the man in the middle will get to see it first, same goes for the response from the server to your network.

Do not use a website that is not HTTP Secured. If you see your gmail account showing http, instead of https, then just kill the session. Do not pass any login credentials.

4.4 DES Algorithm in Use

What is Your Recommended Mitigation Plan?

Start using AES algorithm

Why Did you Recommend This Course of Action?

DES algorithm is outdated, which means that it can be cracked by the hackers. With the growth of technology and the computational power, it has become possible to crack DES algorithm using brute force attack and other hacking methodologies. AES algorithm is the most secured at the moment, and according to NIST, the US government is also using AES encryption to keep the States data secured.

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?

STEPS:

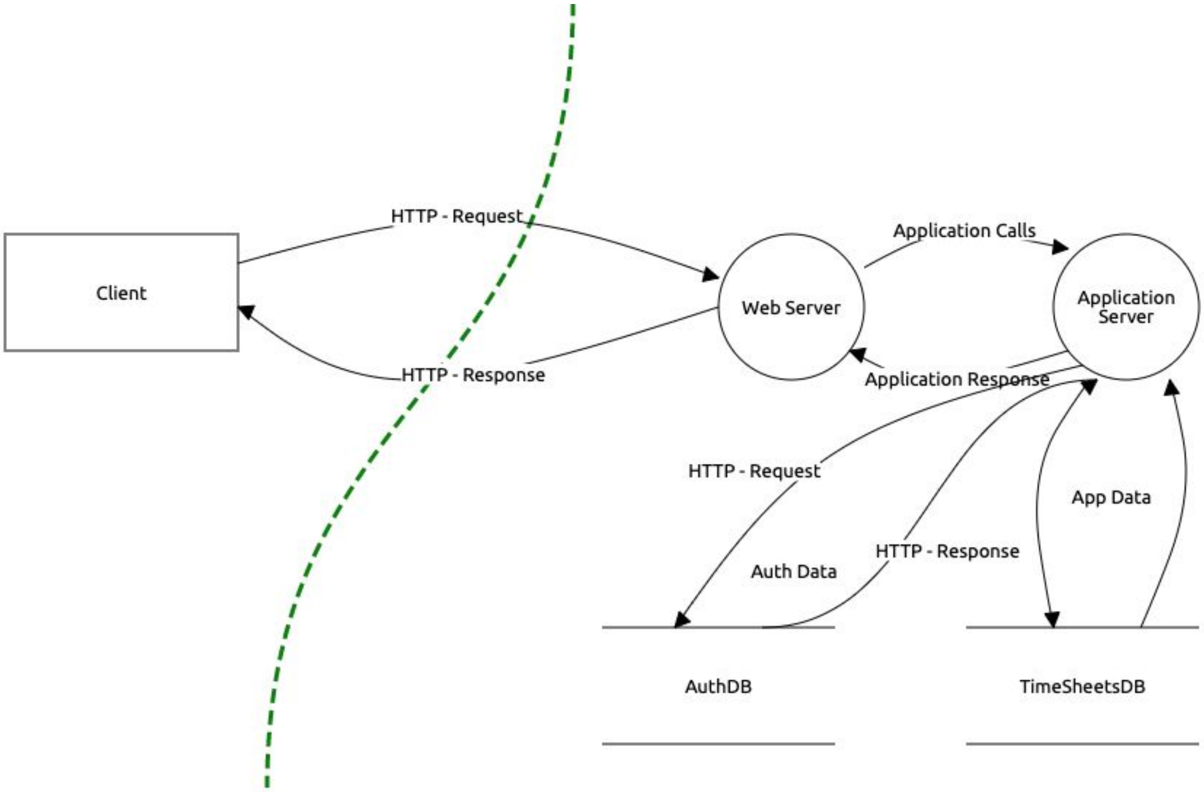
It is highly recommended to work the with cybersecurity framework checklist. Depending on the type of company, you can use any of the followings: NIST (Data Security), HIPAA (Health Security), PCI-DSS (Credit Card security), etc.

1. Update all the outdated encryption.
2. Secure the data at rest with AES-256 encryption.
3. Secure the saved authentication credential with HASHING protocol.
4. Use HASHING, SSL or certificate to secure the login/authentication credentials that are in transit.
5. Document all current security policies and procedures for easy access.
6. Evaluate activity logs to determine if all IT staff have performed the necessary safety policies and procedures.
7. Analyze your security patches to ensure everything is up to date.
8. Implement all encryption best practices where appropriate.
9. Double check exactly who has access to sensitive data and where that data is stored within your network.

Optional Task:

Create an architecture diagram of a secure system.

Image of your secure architecture:



- Employee Data encrypted at Rest using AES-256 encryption
- Authentication data is using HASHING protocol
- Authentication requests are encrypted in transit
- Sensitive data is encrypted using AES-256 algorithm

Optional Task *(Continued)*:

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

1. Educate employees on social engineering attacks and phishing.
2. Verify the security of every one of your wireless networks.
3. Regularly review event logs to keep human error at a minimum.