














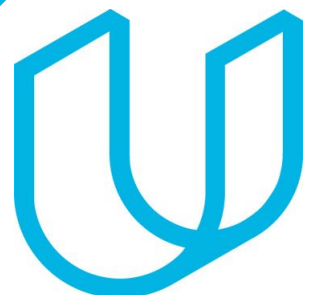


TimeSheets: Threat Report

YOUR NAME:

DATE



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

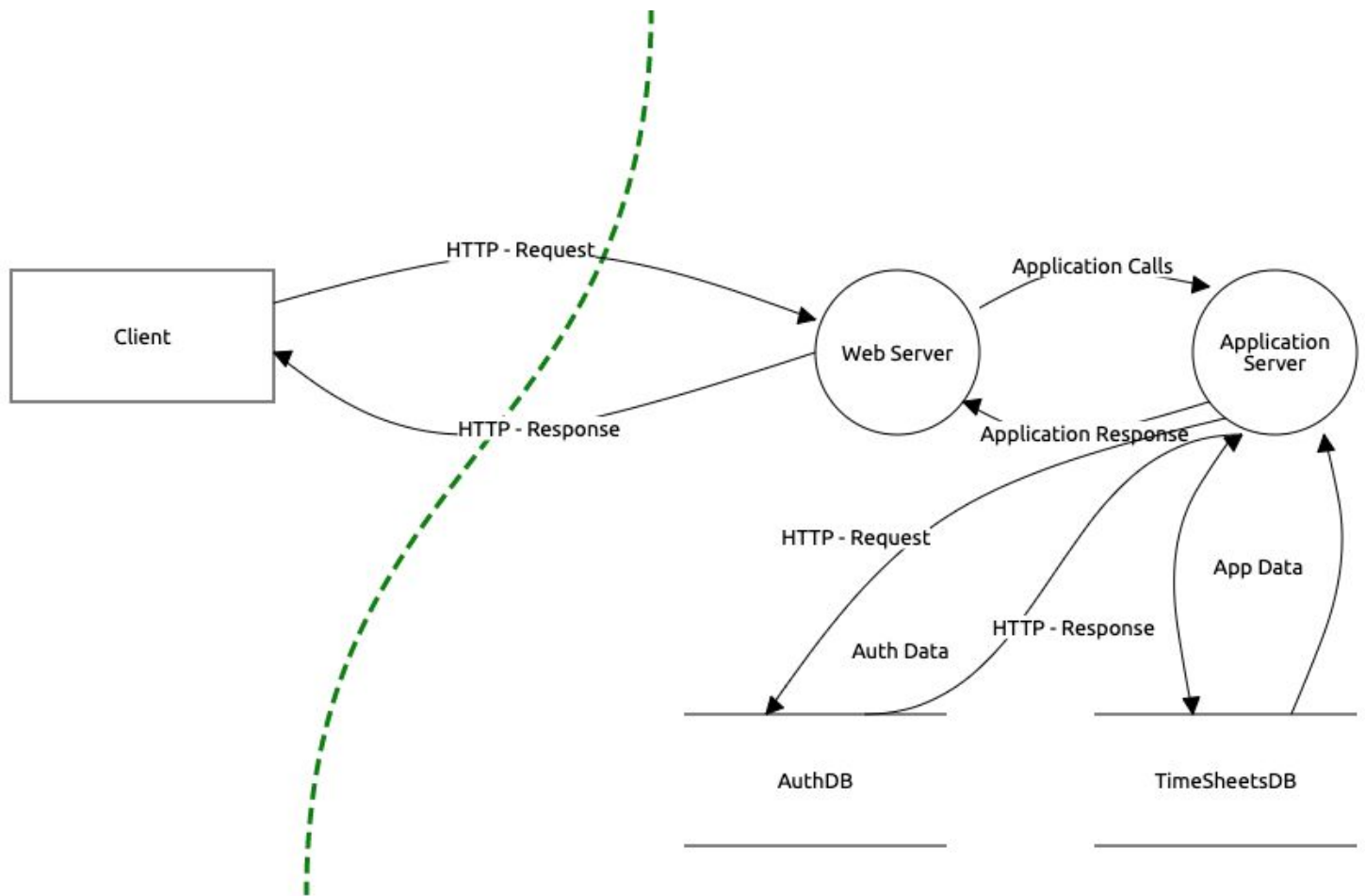
Request is generated from the client via the Internet. The request arrives at the TimeSheets web server which serves static content to the user (HTML, images, etc). Dynamic data is retrieved from the database and served to the client.

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?

If the database is not encrypted, hackers can exploit the server. Furthermore, any cleartext passwords stored on the database server will be compromised.

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?

If the credentials are stored in cleartext, they can be read by humans, allowing hackers to read login credentials and use them to gain access to the rest of the system infrastructure. Those credentials must be hashed with a salt and rainbow tabletop mixture.

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?

Plaintext is visible to the naked eye. Attackers can intercept and steal your login credentials if credentials are stored and transmitted in plaintext. One of remediators that can help encrypt authentication requests are using encrypted protocols ports with hashing combination.

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 is considered insecure because it is 56-bit size keys. It is not long enough and modern computers can bruteforce those keys in short amount of time.

Optional Task:

Examine the threat model diagram from Section 1 and answer:

What non-encryption issues can you identify?

1. HTTP protocol
2. Auth DB and Timesheet DB is not encrypted.
3. Multiple login sessions to Database from App Server.

What recommendation would you give to solve those issues?

1. Using HTTPS protocol
2. Encrypted login credentials that stored in database using Salt + Rainbow Hashing encryption.
3. Use certificate to verify server communications.
4. Limit or don't login to database server directly.

Why do you recommend those solutions?

1. Establish HTTPS, VPNs, certificates and/or Asymmetric Encryption communication internally and externally.
2. Create Database admin login. Then, create a jumpbox server internally that can communicate with DB directly.



Section 3

Risk Analysis

3.1 Scoring Risks

Risk	Score <i>(1 is most dangerous, 4 is least dangerous)</i>
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?)*

Threat x Vulnerability = Likelihood

Risk = Likelihood x Impact

Unencrypted during transit is Threat meaning any hacker that know how to sniff network traffic can read data during transmission. This is why i rate this as 1.

Unencrypted at Rest is vulnerability. The hacker has to have access to database or hard drive data in order to be compromise. If the hacker got information about the database or hard drive during network transmission, this become a likelihood compromise. This is why i rate this as 2.

Likelihood is hacker attempted to decrypter database. The database is encrypted but using old or outdated **Algorithm**. This is why i rate this as 3.

Reversible Encryption is rated as 4 because we do not know if this encryption is up to date or not. In addition, the data is encrypted so the attacker has to spend some time to decrypt the data.



Section 4

Mitigation Plan

4.1 Employee Data Unencrypted at Rest

What is Your Recommended Mitigation Plan?

My recommendation for mitigation plan for Employee Data unencrypted at Rest using Salt + Rainbow combination, encrypted database and limit to no direct access to database.

Why Did you Recommend This Course of Action?

The reason I have recommend these mitigation plan because the data must be encrypted. In addition, the database must be protects from attackers to gain access. Encrypted the Database is the first step, second steps is protecting the data within the database which the Salt Hash + Rainbow encryption combination will protect the data. Lastly, making the database not access directly instead creating a jumpbox server internally. That jumpbox server can only communicate with the database.

4.2 Authentication Data Stored Using Reversible Encryption

What is Your Recommended Mitigation Plan?

Using AES-256 bit encryption. If there bigger budget available, I would recommend AES-512 encryption

Why Did you Recommend This Course of Action?

I have recommend AES-256 keys encryption because it harder to decrypt by hackers. Even the supercomputers may take years in order to crack the encryption.

4.3 Authentication Requests are Not Encrypted in Transit

What is Your Recommended Mitigation Plan?

Using encryption protocol like HTTPS, SSL, VPN etc with combination of certificate signing and PKI.

Why Did you Recommend This Course of Action?

I recommend this course of action because HTTPS prevent hackers from read network traffic in transit. Certificate signing and PKI provide asymmetric key to show authenticity of servers. VPN provide extra security layer on top current security measures.

4.4 DES Algorithm in Use

What is Your Recommended Mitigation Plan?

Using AES 128, 192 or 256 bit keys encryption.

Why Did you Recommend This Course of Action?

I recommend this course of action because DES algorithm can be crack by modern computers. Also, it is short 56 bit key length. AES provide strong key length and it is harder to crack.

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?

I would recommend implement these standards across the board. Then, I would perform penetration testing against these standards to reinforce this infrastructure. Then, provide user training to employees on security awareness using company equipment. Furthermore, send out security awareness to all the employees of the company about how attackers can use social engineering techniques to gain access internally. Lastly, hire third party penetration company to test our current security infrastructure in place including the users.

During new hire onboarding process, all employees must sign company equipment use agreement when using company equipment. Then, i would have security audit team to perform audits within the company including IT department making sure they are upholding security measure in place.

Optional Task:

Create an architecture diagram of a secure system.

Image of your secure architecture:

1. I would put web server in DMZ.
2. Then, put IDS/IPS in between external network and internal network. Web Server would be in external network and internal network will include Application and Timesheet and Auth databases.
3. Implement HTTPS, certificate and PKI in the webserver and Application Server.
4. The application server is jumpbox which only can communication with AuthDB and TimesheetDB so i would put load balancer. If we are using cloud provider, I would use auto scaling if the server is unhealthy.
5. Create honeypot outside the firewall to trick hackers.
6. I would limit number of login sessions communications between application server and both database.
7. I would encrypt web server, application server and both databases.
8. I would use AES-256 bit key encryption with Salt + rainbow encryption on both databases.

Optional Task (*Continued*):

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

- 1. I would encrypt web server, application server and both databases.**
- 2. I would use AES-256 bit key encryption with Salt + rainbow encryption on both databases.**
- 3. Create honeypot outside the firewall to trick hackers.**
- 4. Implement HTTPS, certificate and PKI in the webserver and Application Server.**