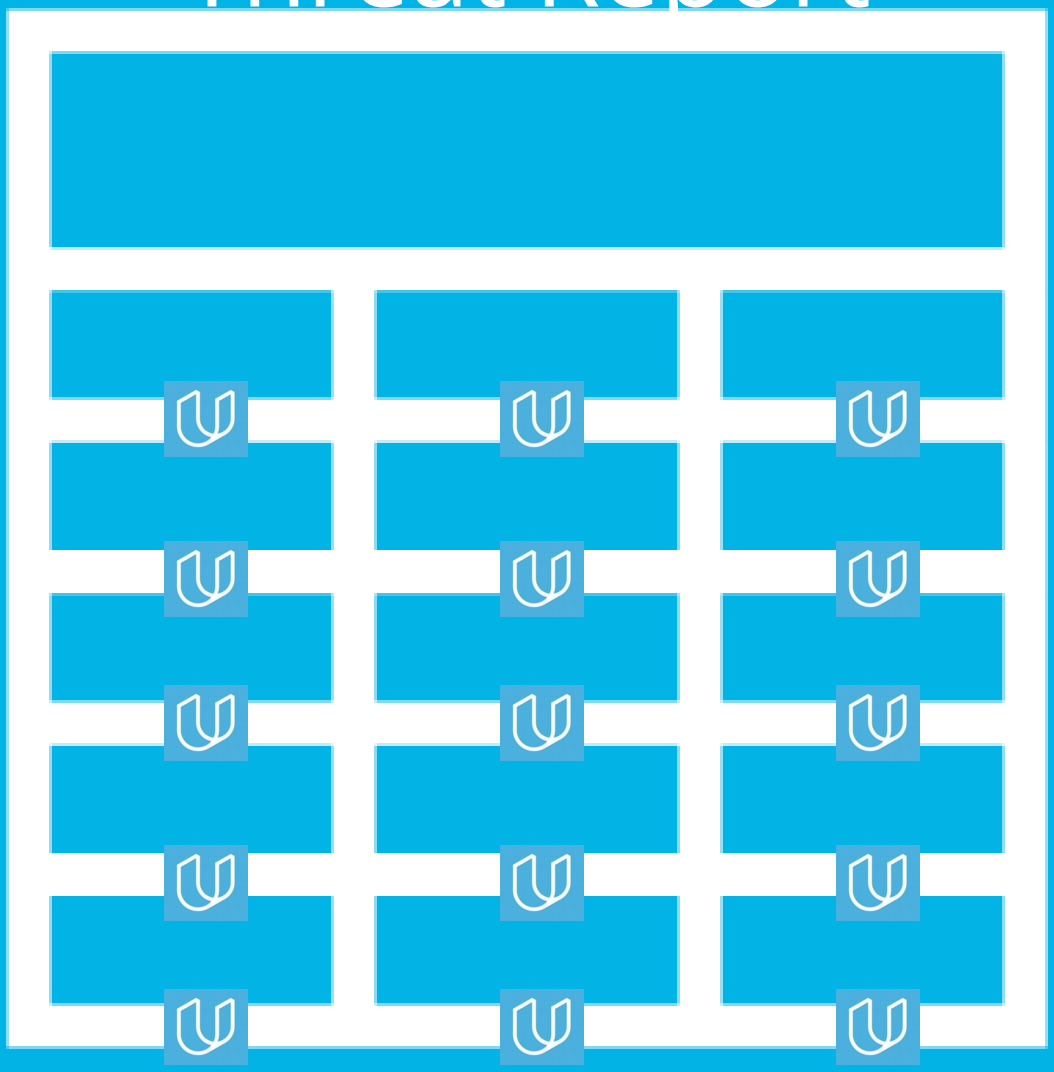
TimeSheets:
Threat Report



Sandra Emma
13th November 2020

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

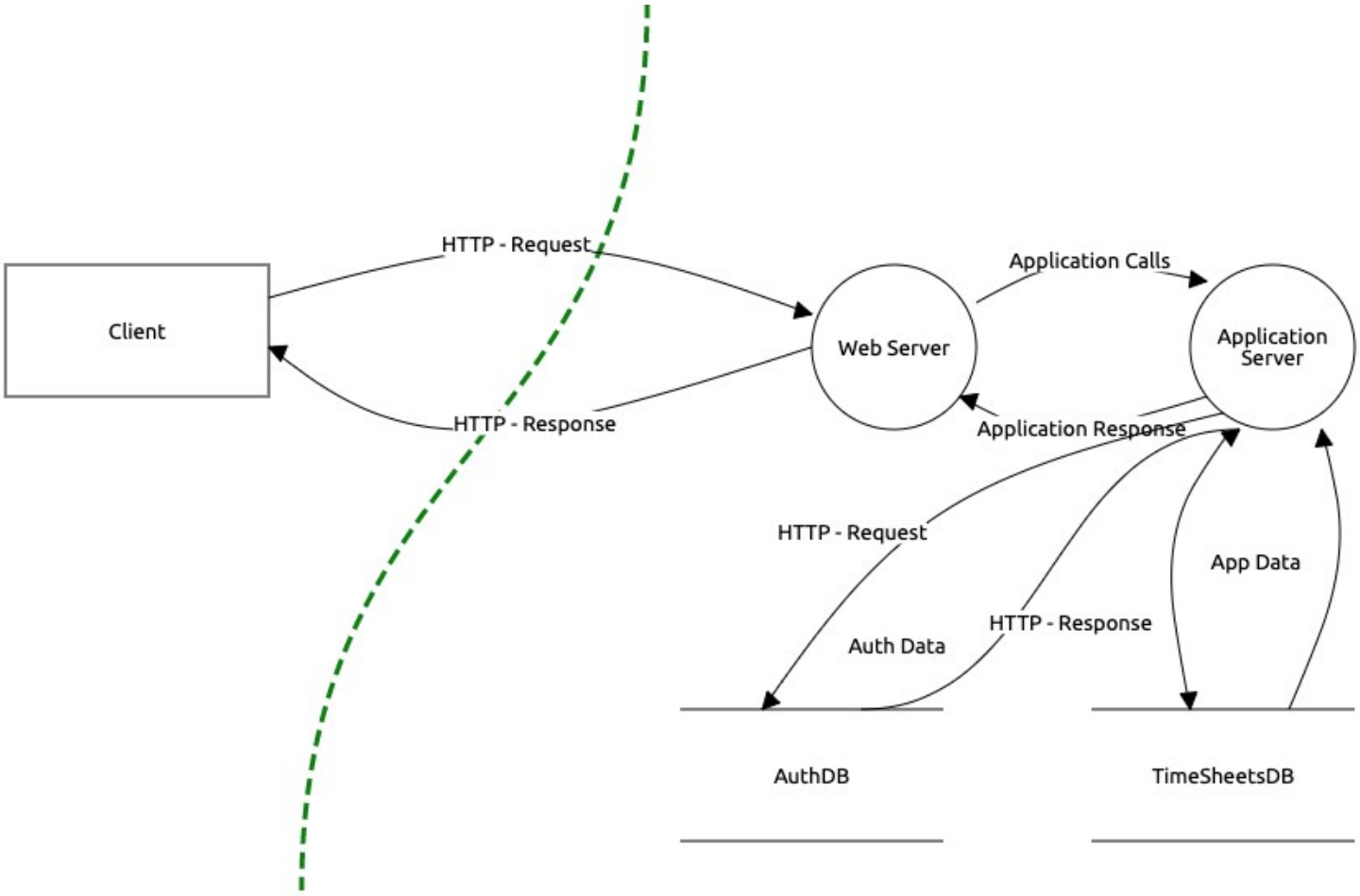
Request is generated from the client via the Internet. The request arrives at the TimeSheets web server which serves static content to the user (HTML, images, etc). Dynamic data is retrieved from the database and served to the client.

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 database server stores employee data and will be queried from the application server. Now, storing sensitive information such as employee data on the database server should be encrypted for security reasons. Data at rest is subject to threats from hackers and other malicious threats. Having Employee Data Unencrypted at Rest is an issue because:

Data breaches of unencrypted sensitive information occur often, and many are highly publicized. Businesses are thrust into the spotlight and scrutinized for scandalous lack of oversight and accountability around data security.

Unprotected sensitive data leads to identity theft, fraud, and theft of financial resources from employees and customers. There might be data leakage, because information such as employee data is very trivial to an organization. Hence, the information is no secured, which makes it vulnerable to hack attacks. This information could be accessed by a third party. Confidential information could be accessed by other employees too there by breaching confidentiality contracts.

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?

Authenticating data using reversible encryption policy setting provides support for applications that use protocols that require the user's password for authentication. Storing encrypted passwords in a way that is reversible means that the encrypted passwords can be decrypted. This becomes an issue because a knowledgeable attacker who is able to break this encryption can then log on to network resources by using the compromised account.

Based on my research, storing password using reversible encryption is essentially equal to storing plaintext versions of the passwords. This is disadvantageous in that, authentication information is exposed.
So, it is not recommended.
Reversible encryption should not be used for authentication because the specific requirements and parameters of authentication are incompatible with the weakness of reversible encryption. The primary weakness of reversible encryption is simple: if the key is compromised, the encrypted data is compromised.

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?

During data transfer, many communication channels can be "sniffed" by attackers during data transmission. For example, network traffic can often be sniffed by any attacker who has access to a network interface. This significantly lowers the difficulty of exploitation by attackers. The lack of proper data encryption passes up the guarantees of confidentiality, integrity, and accountability that properly implemented encryption conveys. If the authentication information is transmitted in plaintext, it is possible for an attacker with access to the network traffic to sniff in and get the information. Omitting the use of encryption in any program which transfers data over a network of any kind should be considered on par with delivering the data sent to each user on the local networks of both the sender and receiver. Worse, this omission allows for the injection of data into a stream of communication between two parties -- with no means for the victims to separate valid data from invalid. In this day of widespread network attacks and password collection sniffers, it is an unnecessary risk to omit encryption from the design of any system which might benefit from it.

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, the Data Encryption Standard, can no longer be considered secure. While no major flaws in its innards are known, it is fundamentally inadequate because its 56-bit key is too short. These sizes are typically not large enough for today's uses. 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.In short, it is now absolutely clear that DES is not secure against any well-funded opponent, any opponent with access (even stolen access) to enough general purpose computers

Now, using DES algorithm on sensitive data is not secured because it can easily be attacked by simple brute force which makes it vulnerable to attacks.

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?)*

I have made use of the Common Vulnerability Scoring System Version 3.1 Calculator(CVSS)

Risk	CVSS score
● Employee Data Unencrypted at Rest - CVSS score	8.5
● Authentication data is using reversible encryption	7.5
● Authentication requests are not encrypted in transit	9
● Sensitive data is encrypted using DES algorithm	5

3.2.1 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?)*

I made use of on an online CVSS calculator and inserted the various risks into the system and obtained the values shown in the table above. This helped me make my decision on the ranking system explained below. From the ranking, we discover that Unencrypted data in transit rest has the highest severity risk and the highest value, reason why it is the most dangerous. Next we have Unencrypted data at REST. Followed we have reversible encryption which poses some problem and least dangerous among the 4 was sensitive data using DES.

- *Unencrypted data in Transit: Scored 9 and is ranked first since a malicious user can intercept plaintext data transmitting across unencrypted network and gain unauthorized access to that jeopardize sensitive data.*
- *Unencrypted at Rest: Scored 8.5 and is ranked second. When sensitive data is at REST, it can easily be accessed by unauthorized users. once a malicious users gets access to the data, all information is made available to them.*

3.2.1 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?)*

- *Reversible Encryption: Scored 7.5 and is ranked third. We know that storing encrypted passwords in a way that is reversible means that the encrypted passwords can be decrypted. This becomes an issue because a knowledgeable attacker who is able to break this encryption can then log on to network resources by using the compromised account.*

- *Outdated Algorithm: Scored 5 and is ranked fourth and least severe. Data Encryption Standard, can no longer be considered secure. While no major flaws in its innards are known, it is fundamentally inadequate because its 56-bit key is too short. Now, using DES algorithm on sensitive data is not secured because it can easily be attacked by simple brute force which makes it vulnerable to attacks.*

Section 4

Mitigation Plan

4.1 Employee Data Unencrypted at Rest

What is Your Recommended Mitigation Plan?

Data encryption, which prevents data visibility in the event of its unauthorized access or theft, is commonly used to protect data in motion and increasingly promoted for protecting data at rest.

The encryption of data at rest should only include strong encryption methods such as AES or RSA. Encrypted data should remain encrypted when access controls such as usernames and password fail. Increasing encryption on multiple levels is recommended. Cryptography can be implemented on the database housing the data and on the physical storage where the databases are stored. Data encryption keys should be updated on a regular basis. Encryption keys should be stored separately from the data. Encryption also enables crypto-shredding at the end of the data or hardware lifecycle. Periodic auditing of sensitive data should be part of policy and should occur on scheduled occurrences. Finally, only store the minimum possible amount of sensitive data.

4.1 Employee Data Unencrypted at Rest

Why Did you Recommend This Course of Action?

AES stands for Advanced Encryption Standard and is in wide use around the world. It falls into a class of encryption methods called "symmetric" encryption. That is, the same secret (an encryption key) is used to encrypt the data, and also used to decrypt the data. AES encryption is probably the most widely used encryption method for protecting ***data at rest.*** *You will find it used in self-encrypting disk drives, database encryption, storage encryption, and so forth. It's been around since about 2002, and it is an international standard. Roughly speaking, when you encrypt with AES you put data and the secret encryption key into software that implements AES encryption, and out comes the encrypted data. When you want to use that data you put the encrypted data and the same encryption key into the software, and out comes the original data that you can use.*

AES encryption is great when we have a constrained environment. For example, if we encrypt data in a database, we will decrypt data when we need to access the database. Another example is hard drive encryption - we encrypt the data written to the disk, and decrypt it when we read from the disk. Encryption and decryption will take place on the same platform and in the same context. AES encryption is great for this particular use case. That is why it is commonly used for protecting ***data at rest****.*

4.2 Authentication Data Stored Using Reversible Encryption

What is Your Recommended Mitigation Plan?

We can make use of the Secure Hash Algorithm (SHA-3), instead rather than using reversible encryption.

Why Did you Recommend This Course of Action?

Hashing is a one way process. It ensures that the plain text password can never be recovered even if the hash is stolen or revealed from the computer. This makes it easy to manage password storage and retrieval. this is also one of the major differences between hashing and encryption. We cannot revert a hash digest back into its readable form even if we know about the algorithm being used to generate the hash. In case of encryption, we can get back the readable text by decrypting the data back.

The Secure Hash Algorithm 3 (SHA-3) is a computer security cryptographic algorithm. It was created by the US National Security Agency (NSA) in collaboration with the National Institute of Science and Technology (NIST) as an enhancement to the SHA-2 algorithm. SHA-2 has six different variants, which differ in proportion with the bit size used for encrypting data. SHA-3 is the newest member of the Secure Hash Algorithm family, but it is built quite differently from its predecessors. At this stage, it has not yet replaced SHA-2, but simply gives cryptographers another option that can provide improved security in certain situations.

4.3 Authentication Requests are Not Encrypted in Transit

What is Your Recommended Mitigation Plan?

Encrypt the data before transmission .

Why Did you Recommend This Course of Action?

The purpose of data encryption is to protect digital data confidentiality as it is stored on computer systems and transmitted using the internet or other computer networks. The outdated data encryption standard (DES) has been replaced by modern encryption algorithms that play a critical role in the security of IT systems and communications.

These algorithms provide confidentiality and drive key security initiatives including authentication, integrity, and non-repudiation. Authentication allows for the verification of a message's origin, and integrity provides proof that a message's contents have not changed since it was sent. Additionally, non-repudiation ensures that a message sender cannot deny sending the message.

4.4 DES Algorithm in Use

What is Your Recommended Mitigation Plan?

Implement stronger algorithm like triple DES

Why Did you Recommend This Course of Action?

Triple DES was designed to replace the original Data Encryption Standard (DES) algorithm, which hackers eventually learned to defeat with relative ease. At one time, Triple DES was the recommended standard and the most widely used symmetric algorithm in the industry.

Triple DES uses three individual keys with 56 bits each. The total key length adds up to 168 bits, but experts would argue that 112-bits in key strength is more like it. In 3DES, data is run through the DES algorithm three times instead of just once, which makes it 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?

- *Communicate new policies to employees. Change is inevitable, when business policies and procedures change, communicating these with employees is essential to avoid costly mistakes and errors. Depending on the nature of the policy or procedure that's being changed, there could be legal and financial consequences if the organization does not comply,*

- *Make sure all procedures are well documented: Recording internal procedures is crucial. In an audit, you can review these procedures to know how people are interacting with the systems. These procedures can also be analyzed in order to find systematic faults in how a company interacts with its network.*
- *Make sure the written version of the policy can be understood by the total population of the workforce;*
- *Check the penetration testing process and policy: Penetration testing is one of the key methods of locating vulnerability within a network. Review the current pen-testing methods and assess the process in which they're employed.*

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?

- *Make sure sensitive data is properly is stored separately: Social security numbers or medical records should be stored in a different location with differing levels of access to other less personal data.*
- *Encrypt company laptop hard disks: Sensitive data should ideally never be stored on a laptop. However, often laptops are the focus on many people's work lives so it is important to be able to account for them.*

  Less sensitive data which may be stored on a laptop can be encrypted to provide increased security.

- *Check wireless networks are secured: It is important to try to use up to date technology to secure your networks, otherwise, you leave them vulnerable. Avoid WEP or WPA and make sure networks are using WPA2.*

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?

- *By regularly performing an internal audit, you can ensure compliance with any and all relevant laws and regulations. Internal auditing programs are critical for monitoring and assuring that all of your business assets have been properly secured and safeguarded from threats. It is also important for verifying that your business processes reflect your documented policies and procedures.*
- *Test software which deals with sensitive information*
- *Look for holes in the firewall or intrusion prevention systems: Assess the effectiveness of your firewall by reviewing the rules and permissions you currently have set. Often, holes in a firewall are intentionally created for a reasonable purpose - people just forget to close them back up again afterward.*