NAME: Emeka Michael Nzeopara
ADVANCE SECURITY LAB 1 : Threat Modeling

Task 1 - Decompose the Application

1. Describe entry points assets, and trust levels in forms of tables
2. Select at least 3 use cases that you think are the most interesting and prepare Data Flow Diagram (DFD) for them.

The application can be affected from and/or within the company. Because of this we can divide the threat into two places which can be internal and external.

In the case of the internal threat we should consider threats like developers, operations staff, DevOps team and their likes. For internal threats we will look within the code of application and infrastructure that is used to deploy the application

In the case of the external threat, we will consider the threat from the outside sources, more like users of the applications. Therefore, I may divide the possible trackers into external and internal. For external user entry points are the web application interface and its servers.

Entry Points:

Internal Threats

Name	**Description**	**Trust Level**
Hardware	The hardware part of the infrastructure	IT Support Staff (10) Developers(11) DevOps (12)
Infrastructure	The processes inside the company (CI/CD, etc.)	DevOps (12) Operations (13)
Code	Code of web application and necessary algorithms for video + Dependencies that can be considered as an external threat	Developers (11) DevOps (12)

External Threats

Name	Description	Trust Level
HTTPS Port	The web application will be available through the 443 port	Anonymous Web User (1) User with Valid Credentials (2) User with Invalid Credentials (3)
Login Page	Interface to the login function	Anonymous Web User (1) User with Valid Credentials (2) User with Invalid Credentials (3)
Login Function	Sign in and sign up	User with Valid Credentials (2) User with Invalid Credentials (3)
Content Page	Main page for all user interaction	Anonymous Web User (1) User with Valid Credentials (2) User with Invalid Credentials (3)
Search function	Query to the server and DB	Anonymous Web User (1) User with Valid Credentials (2) User with Invalid Credentials (3)
get content function	GET query to obtain content from a server	Anonymous Web User (1) User with Valid Credentials (2) User with Invalid Credentials (3)
Account page	Show information to user, web interface to manage account	User with Valid Credentials (2) Content maker (4)
Manage function	Delete, upload, and edit video. Show info, give access	Content maker (4)
Moderating function	Block content	Moderator (9)

To tabulate how the Assets would be arranged we can also divide them into three categories which could include the user, infrastructure and applications.

Users Assets

Name	Description	Trust Level
Personal Data	App stores some personal data	DB admin (5) Web server admin (6) Web server user process (7) DB user (8) Moderator (9)
Credentials	The user credentials to log into the app	User with Valid Credentials (2) DB admin (5) Web server user process (7) DB user (8)

Assets for the Infrastructure

Name	Description	Trust Level
Code execution	Ability to execute source code	Web server admin (6) Web server user process (7)
Access to DB	Ability to interact with databases information	DB admin (5) DB user (8)
Access to network	Ability to interact with infrastructure within the company	DevOps (12) Operations(13)
Hardware	Access to the hardware part of the infrastructure	Staff (10) Devs(11) DevOps (12)

Assets Related to Applications

Name	Description	Trust Level
Availability	The app should be available for users	DB admin (5) Web server admin (6)
Content (data)	Ability to manipulate the content of the app	Content maker (4) Moderator (9)
Code	Direct access to the source code	Devs (11) DevOps (12)
Session		User with Valid Credentials (2)

Trust Levels

ID	Name	Description
1	Anonymous Web User	A connected user without credentials
2	User with Valid Credentials	Connected user with valid credentials

3	User with Invalid Credentials	Connected user with invalid credentials
4	Content maker	User with publisher content
5	DB admin	User with read and write access to the databases
6	Web server admin	User with the ability to configure app servers
7	Web server user process	The process used by the app server to execute code
8	DB user	User with limited read and write permission
9	Moderator	User with the ability to block the open content
10	Staff	Persons within the company without direct access to the development
11	Developer	A person with direct access to the source code
12	DevOps Engineers	Engineers who manage the software platform

To describe the data flow of the application, I will be considering 3 cases to this process. The first case will be when a user want to get information about their account. This can be seen in the diagram below

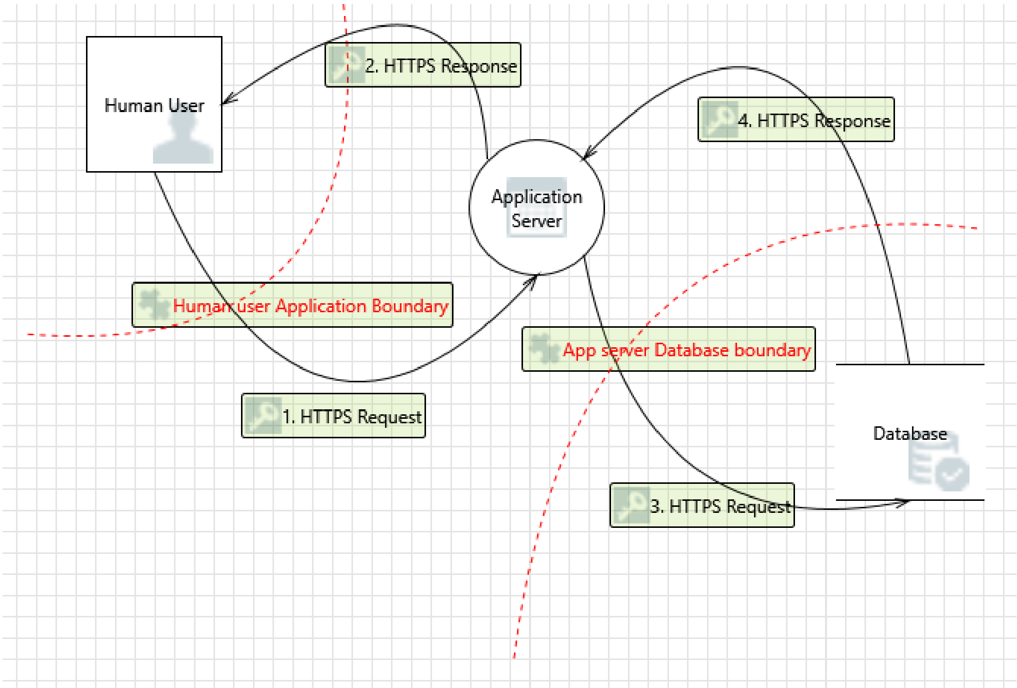


Figure 1: Get Information on Account and Video

This figure can be used for different scenarios, although we can use getting an information to explain it. It just states a scenario where the user makes a request to the application server and the server responds with some prompt authentication process. After that is verified, the application server will query the database and supply the information that is being requested.

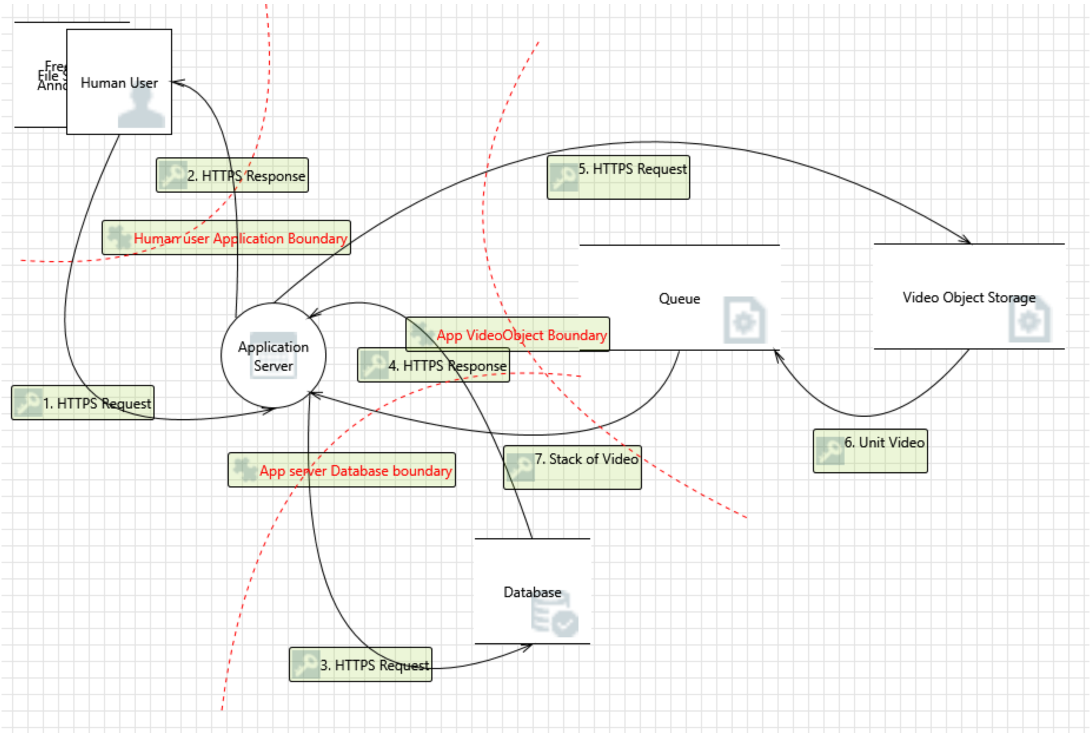


Figure 2: Get a Video

In the figure 2 above, it is like the extension of the figure 1, suppose there is need to obtain a video from the storage, lets assume the video requires some form of verification e.g a private video for instance. Firstly the steps that has been spoken of in the previous diagram will need to be passed. After this has successfully moved, the application server will make a request to the video storage which will subsequently respond by passing the video file to a queue and then the application server.

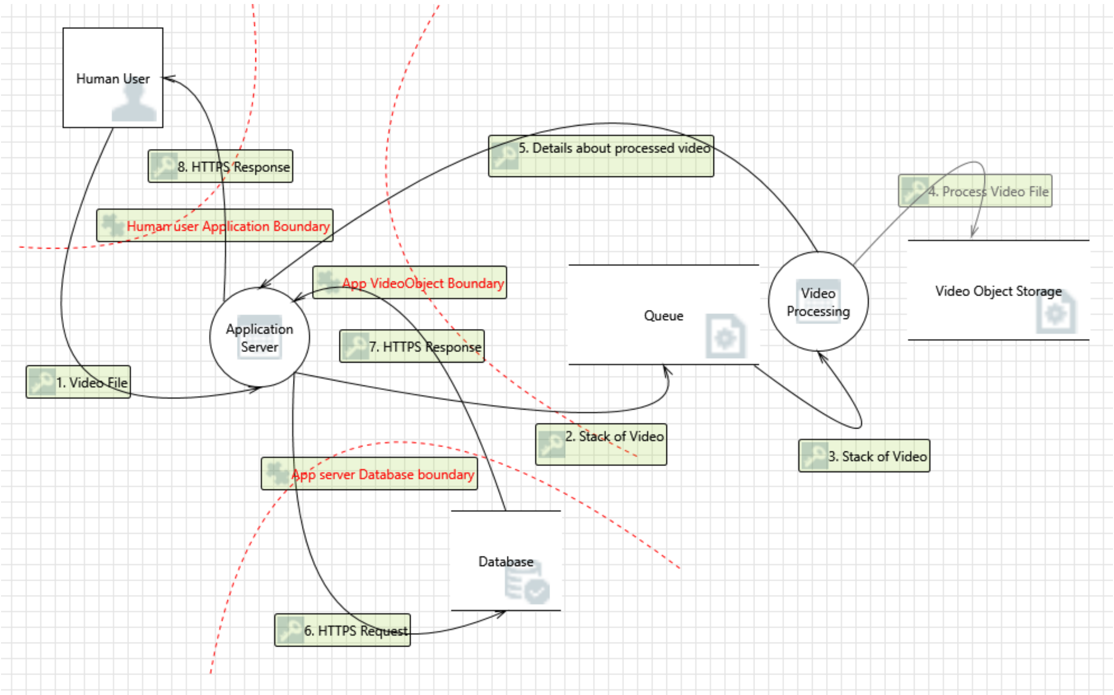
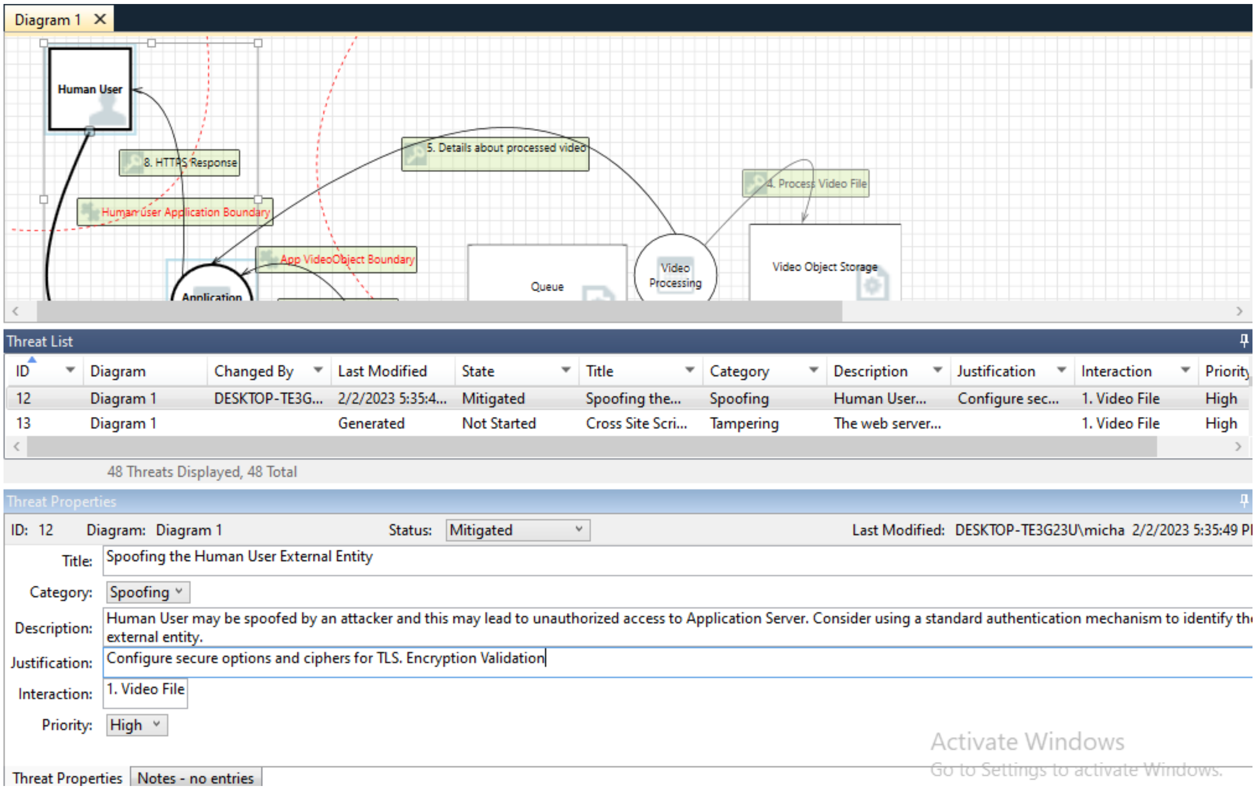


Figure 3: Video Uploading

Let us give a scenario of where we want to upload a file to the web application. A video file will be sent to the application server, the server will perform some validation and send the stack of videos to the video processing queue. Each video in the stack of videos goes through the video processing and is stored in the video object storage. After this has been completed, there would be need for the server to update its database and this is done by the server sending details about the video it has received to the application server and the application will update the database by sending a request, getting a response and finally updating the user about the completion of the whole process.

Since we are modeling the threat, there would be need for us to check how the threat can be watched from the microsoft tool. This can be done by checking the analysis of the infrastructure, a screenshot is given below.
Let us take for instance the HTTPS traffic flowing from the user to the Database, this traffic might be vulnerable to spoofing attack, which will be willing to collect the credentials of the user and by doing so log in as the user of the credentials.
This spoofing is known to be impersonation and could be mitigated by various forms of ways, of which encryption is a part of it.
By using the latest form of encryption, we could be able to work something out in resolving this problem. A simple snapshot is given below



Threat List

ID	Diagram	Changed By	Last Modified	State	Title	Category	Description	Justification	Interaction	Priority
12	Diagram 1	DESKTOP-TE3G...	2/2/2023 5:35:4...	Mitigated	Spoofing the...	Spoofing	Human User...	Configure sec...	1. Video File	High
13	Diagram 1		Generated	Not Started	Cross Site Scri...	Tampering	The web server...		1. Video File	High

48 Threats Displayed, 48 Total

Threat Properties

ID: 12 Diagram: Diagram 1 Status: Mitigated Last Modified: DESKTOP-TE3G23U\micha 2/2/2023 5:35:49 PI

Title: Spoofing the Human User External Entity

Category: Spoofing

Description: Human User may be spoofed by an attacker and this may lead to unauthorized access to Application Server. Consider using a standard authentication mechanism to identify th external entity.

Justification: Configure secure options and ciphers for TLS. Encryption Validation

Interaction: 1. Video File

Priority: High

Threat Properties Notes - no entries

Figure 3: Analysis of a possible Attack

Details of a structured way on how these attacks are possible and their respective mitigation is given in the next section below.

Task 2 - Determine Threats

Task: To apply STRIDE for each asset in the application and come up with a summary table with the following columns

As at this point, we have been able to decompose the system and discuss some possible scenarios of what can take place in the system. In this task we will like to determine the threat landscape. We will be using **STRIDE** to achieve this so that we can categorize the threats in a simple, clear and a repeatable manner.

Assets	Category	Threat	Vulnerability	Score	Countermeasure
Code execution	Elevation of Privilege	An attacker can execute commands	Lack of regular patching could lead to	9.6	Patching the vulnerability Principle of least privileges for processes Forced

			remote code execution Injection		validation for input Input filtering Encoding
Access to DB	Spoofing Tampering	Databases contain a lot of information that should be securely stored	Data can be modified by external user Data can be accessed by external user	9.6	Principle of least privileges for processes Validation Encryption HMAC
Code base	Tampering Information Disclosure	Contains known vulnerabilities. Its dependencies can also be affected by the attacker	Vulnerable dependencies exposed data in the codebase. This can lead to disclosure of sensitive information in error or logs - Non Repudiation SCA	9.3	Implement the use of DAST, SAST tools. Configure logging, secrets scanning, backup data, integrity check, Train information security developers.
Session	Spoofing Tampering	Session can be hijacked absence/weakness of identity token validation	It will lead to bypassing authorization	8.2	Configuring the appropriate safety options for token and authorization process Sensitive cookies are encrypted and expired after logout Extra authentication with important actions
Access to network		Infrastructure within the company can be exposed	Exposed port Misconfiguration Exposed environment Exposed confidential	8.0	ACL configuration for logs Configuration of environment Secret scanning Least privilege

			information into open sources Exposed services		
Credentials	Spoofing Tampering	The credentials can be spoofed by an attacker	The insecure transition of credentials Insecure storage of credentials Insecure credentials configuration	7.2	Configure secure options and ciphers for TLS. Encryption Validation
Availability	Denial of Service	An attacker can affect the work of the application	Network layer flood Domain hijacking	6.8	Applying restrictions depending on the frequency of requests
Malfunctioned Hardware	DoS Information Disclosure	The hardware part of the infrastructure can be affected	Hardware Damaged by person Stole Was modified Damaged by events (flood, earthquake, etc.)	6.8	Backup Mitigation Creation of protected premises Development of physical security procedures Distributed infrastructure
Content	Information Disclosure	Attacker can get access to confidential content	Saving content in the browser. Misconfiguration Spoofing the private content	5.8	Sensitive info does not store Applying appropriate configuration for exceptions and errors Encryption for content with safe ciphers.

Personal Data	Spoofing	Personal data can be obtained by the attacker	Personal information is stored openly and can be accessed	5.3	Hide confidential info about a person

Reference

- STRIDE Threat Modeling
- Microsoft Threat Modeling Introduction
- Security Modeling and Threat Modeling Resources
- CVSS Calculator
- OWASP Decompose the Application
- OWASP Determine and Rank Threats
- OWASP Determine Counter Measures and Mitigation
- Threat Modeling Cookbook