The University of Melbourne

SWEN90010: High Integrity Systems Engineering

Assignment 1

Due Date: 11:59pm (Melbourne time), Monday 31st March, 2025

This assignment is worth 10% of your total mark.

In this assignment, you will apply security engineering principles to analyse and enhance the security of a system in its design phase. Given the requirements of the system, you will undertake a structured security analysis that includes the following key activities:

a) Draw a **block diagram** of the system architecture, identifying its **main components** and **communication channels**.

b) Apply the **STRIDE methodology** to systematically **enumerate potential security threats** to the system.

c) Assess the **potential impacts** of these threats on users, organisations, and society.

d) Propose **security requirements** that mitigate the identified threats, ensuring a more secure system design.

Your assignment solution will consist of a written report that answers the questions and carries out the tasks listed in Section 5. You will work in pairs for the assignment with your registered partner on Canvas. Each pair will submit only one solution, produced jointly by both partners. Working in pairs is important since a significant part of the assignment is brainstorming security threats to a system, using the STRIDE methodology discussed in lectures. As with other brainstorming activities, security threat enumeration is an inherently creative process that will benefit from being performed by a pair rather than by a single individual.

1 Background

A technology firm specialising in digital content integrity is in the process of designing an AI-based *deepfake detection* and *media verification* system called **VeriLens**. The firm aims to provide a reliable tool that helps users determine whether digital media has been manipulated or generated by AI, thereby enhancing trust in online content. VeriLens derives its name from a combination of *Verification* and *Lens*, symbolising its role in scrutinising and validating digital media authenticity.

The system is designed to assist journalists, social media platforms, government agencies, and businesses that rely on accurate visual content for news verification and fraud prevention. VeriLens will analyse media files submitted by users and provide a *trust score*, indicating the likelihood that the content is real or altered. This functionality is critical in combating misinformation, verifying media sources, and ensuring compliance with regulatory frameworks.

To illustrate the system's capabilities, consider the following example:

Example Input and Output

- **Input:** A journalist uploads an image from an online news article, seeking verification of whether it has been digitally manipulated.
- **Processing:** VeriLens scans the image for signs of digital tampering, checks inconsistencies in metadata, and analyses traces of AI-generated elements.
- **Output:** VeriLens assigns a **trust score** of 35% and flags potential signs of deepfake manipulation, advising further verification.

Figure 1: Example of VeriLens analysing an image for authenticity.

As part of its development, the firm is carefully considering security risks and potential threats to ensure the system remains robust against adversarial attacks and misuse. The design phase includes defining security requirements, assessing threats using the **STRIDE** methodology, and proposing mitigation strategies to inform the system design.

The following sections will outline the business model, security requirements, and system architecture, providing further context for your security analysis.

2 Business Model

VeriLens' business model consists of two service tiers:

Free Tier:

- Users can submit a limited number of media files per day for verification.
- Results provide a basic trust score indicating the likelihood of manipulation.
- Processing is queued, leading to longer wait times during high-demand periods.
- Users must create an account, and their verification history is stored for future reference.

Paid Tier:

- Users receive priority processing for faster verification results.
- Access to detailed analysis, including metadata validation and AI confidence breakdowns.
- Option to request human review for cases where automated verification is inconclusive.
- Support for batch processing, allowing media agencies to verify multiple files simultaneously.
- Enterprise-level API access for integration into third-party fact-checking platforms.

3 Security Requirements

VeriLens processes and stores various types of sensitive data to provide its verification services. This includes:

- **User-provided content:** Media files uploaded for verification.

- **System-generated verification data:** Information produced by the AI model, including indicators of potential manipulation.
- **User interaction records:** Data related to users' interactions with the system.
- **Authentication and access credentials:** Information used to manage user accounts and permissions.
- **Billing information:** Payment details for users subscribed to premium services.

VeriLens must ensure that its verification services operate securely and reliably, preventing harm to users and society. Specifically, the system should:

- Protect the confidentiality and integrity of user-submitted media, verification results, and related metadata.
- Ensure that verification outcomes remain trustworthy and are not improperly influenced.
- Prevent misuse of the system that could contribute to the spread of misinformation or harm individuals.
- Avoid exposing users, particularly high-risk individuals, to undue risks related to their use of the service.

4 System Overview

The VeriLens system is composed of multiple cloud-based components, each with a distinct role in delivering media verification services. The architecture separates the user-facing web platform from the back-end AI verification engine, and it incorporates third-party services for authentication and payment. Below is an overview of the key components and their responsibilities:

4.1 User Access and Web Interface

Web Application and CDN: Users interact with VeriLens through a web application accessible via standard browsers. This web interface is hosted on cloud infrastructure (augmented by a Content Delivery Network for fast global access) under the control of the VeriLens company. It is responsible for rendering the user portal where media files are uploaded for analysis, verification results are displayed, and user account settings (like profile and preferences) are managed. The web interface also provides session management – it keeps track of user login status and maintains user sessions so that interaction histories and results can be retrieved. For paid subscribers, the web platform additionally exposes an API endpoint that allows programmatic access to the verification service, enabling integration into third-party tools and automated workflows. The web interface layer enforces authorisation controls by checking user roles/tier (free vs. paid) and permissions before allowing certain actions (for example, only premium users can access advanced features or higher usage limits).

Session Management and Usage Tracking: The web interface handles session management after users authenticate. This includes generating or storing session identifiers (or tokens from the Identity Provider) and tying them to user accounts. It also monitors each user's usage. Free-tier users have a limited number of verification queries per day, so the system counts each submission and ensures the daily limit is not exceeded. These usage counters are typically stored in the system's database as part of the user's record. For paid users, the interface tracks usage as well, which is later used for billing calculations.

4.2 Authentication and Identity Provider Integration

User Authentication (IdP): VeriLens offloads authentication to a trusted third-party Identity Provider (IdP) (e.g. a service like Google Identity or Okta). When a user registers or logs in, the web interface redirects the process to this external IdP, which handles verifying user credentials. Upon successful login, the IdP issues an authentication token back to the VeriLens web application. All user accounts (free and paid) go through this authentication flow, ensuring that only authenticated users can submit media for verification or view results.

4.3 Verification Backend (AI Processing Engine)

Deepfake Detection Engine: The core verification logic of VeriLens resides in an AI-powered backend that processes the uploaded media. When a user submits an image or video through the web interface, the file is forwarded to this verification processing backend. This component runs advanced deepfake detection and media forensics algorithms to evaluate the content's authenticity. It performs tasks like scanning for digital manipulation traces, analysing metadata inconsistencies, applying statistical models, and cross-referencing the media against known authentic content. The verification backend is a cloud-hosted service running on specialised servers (e.g. GPU-equipped instances) to handle the computational intensity of AI analysis. While these servers are provided by a third-party cloud provider (for scalability and performance), the VeriLens team controls the deployed models and code.

Queued Processing and Tier Prioritisation: To handle potentially high demand, VeriLens uses a job queue for verification requests. The web interface places each submitted media file into a processing queue that the backend service works through, paid users receive priority and/or batch processing option in this queue. The backend processes each queued job and returns the verification result (such as a trust score and any associated analysis details) back to the web interface once complete.

4.4 Data Storage and Verification History

Secure Database: VeriLens maintains a central database for persisting data needed by the system. This database is under the company's control (hosted in a secure cloud environment). It stores user information and non-sensitive account details (e.g. the user's profile and service tier). It also stores the history of media submissions and their verification results for each user. When a verification job is completed by the backend, the results (trust scores, flags, timestamps, etc.) are recorded in the database, tied to the user's account. Free users' and paid users' usage metrics are likewise logged here — for example, the number of queries a free user has made today, or the cumulative number of verifications a paid user has performed this billing cycle.

History Management: Because results are stored, the system can provide a "verification history" feature. Users can view past verification outcomes through the web interface, and they have the option to manage this history (for example, deleting records or anonymising past submissions if privacy is a concern).

4.5 Billing and Usage Tracking

Billing System Integration: For paid-tier users, VeriLens integrates with an external Billing Service to handle payments. When a user upgrades to the paid tier or during account creation for a premium account, the web interface collects billing details (e.g. credit card information). Instead of storing this sensitive financial data in the VeriLens database, it securely transmits the information to the third-party billing provider. The billing provider (which could be a payment platform like Stripe or a dedicated billing API) stores the payment details and manages charges. The web interface communicates with the billing service at defined points – for example, to create or update a customer record when the user provides payment info, and to trigger usage billing.

Usage Tracking for Billing: Throughout each billing cycle, VeriLens tracks the paid user's usage (number of verifications, or possibly the resources consumed by those verifications) in its database. At the end of the cycle (e.g. monthly), the web interface will generate a usage report or tally for each paid user. It then communicates with the billing provider to process payments. In a typical flow, VeriLens might send the total usage count or amount due to the billing service, which then charges the user's stored payment method and confirms the transaction. Because free-tier users are not charged, this integration is inactive for them; however, their usage counts are still tracked internally for rate limiting. The **authorisation** logic in the web interface also uses account tier information — for instance, to decide if an account should be allowed to continue making requests or needs to upgrade after reaching a free tier limit.

4.6 System Administration

VeriLens is managed by a team of system administrators within the company. These administrators are responsible for deploying and maintaining the web interface servers, the database, and the AI verification backend. Even though the web front-end might be running on a CDN's machines and the AI engine on a cloud provider's hardware, the **VeriLens administrators maintain control over the software** on those machines. They use administrative workstations on the company's internal network to perform updates, apply security patches, and configure the services. Regular maintenance includes tasks such as updating the deepfake detection model with new training data (to improve accuracy against emerging manipulation techniques) and scaling the infrastructure as user demand grows.

5 Your Tasks

1. [**1 mark**] Draw a block diagram illustrating the system's architecture, including its **main components** and the **legitimate communication** channels [1] between them. Label *each communication channel* with the types of sensitive information (see section 3) that flow through it. Additionally, label each *component* that stores sensitive data with the types of sensitive information it holds.

[1] Legitimate communication channels are the approved and intended pathways for data exchange between system components. These channels exist for the system to function as designed, enabling users, administrators, and third-party services to interact securely. Unauthorised or adversarial channels—such as security breaches, side-channel leaks, or unintended data exposure—are not legitimate communication channels.

For each *component*, provide a brief description (no more than a few sentences) addressing the following:

(a) Who has ultimate control over the component?

(b) What is its role in the system, and how does it interact with other components?

Indicate the **trust boundaries** within the system on your diagram. For each *trust boundary*, specify who has ultimate control over the *components within that boundary.*

Hint: Trust boundaries exist only *between* components, not *within* them. Each component must reside within a single trust boundary. If you find that a trust boundary needs to pass through a component, this suggests that the component should be divided into multiple components, each assigned to a separate trust boundary.

2. [**4 marks**] Use the **STRIDE** methodology to enumerate potential **security threats** to the system. For each threat that you identify you should document:

   (a) **Who is the potential attacker(s)** that might try to exploit this threat?

   (b) **What is the security goal(s)** (e.g. confidentiality, integrity, availability etc) that the attack or threat would violate if it were successful?

   (c) How might the attacker(s) **exploit this threat?**

   (d) Which **system components and trust boundaries** would be compromised?

   Note on ambiguity: Importantly, your report should *document and justify any assumptions you make while carrying out your analysis.* The system description provided above is intentionally ambiguous. You might therefore need to make certain assumptions when carrying out your analysis. You should make sure that your assumptions are reasonable, by including with each a brief justification.

   When documenting how an attacker might exploit a particular threat, try to be as specific as you can and to draw on past incidents or vulnerabilities to justify your choice.

   For example, the threat that an attacker might try to impersonate another user is a bit vague. How might they try to impersonate them and for what purpose? Have such attacks been carried out in the past and if so, how?

   For this question we are expecting you to find a range of potential threats, from relatively simple threats to sophisticated ones. There is no fixed number of threats you are aiming to find. Instead, we want to see that you have been *methodical* and *creative*: if you are methodical in how you apply STRIDE, you should be unlikely to miss simple threats. If you are creative and do your research (see next paragraph) you will find some interesting potential threats in this system. As a rough guide: around five is far too few threats. 40 is likely to be excessive (by that point you may be splitting hairs).

   Note: To get full marks for this part of the assignment you should expect to have to do some research on past vulnerabilities and attacks on similar systems. Remember from the lectures on safety engineering the importance of learning from past incidents.

   If you believe a certain threat might be realistic but cannot find evidence of it being exploited in the past, justify why you think it is realistic and any assumptions you are making when drawing that conclusion. This can include citing research papers in which certain threats have been theorised about.

3. [**2 marks**] Your next task is to think about the **security-related impacts** of each of the threats you identified. To work this out, for each threat, describe the **potential harms** that could arise if it were successfully exploited. When thinking about impacts, you should consider impacts on individuals, the various companies involved in this system, plus the government and society (if applicable), and VeriLens as a business, with appropriate justification for your reasoning. You do not need to assess how likely the threat or exploitation is - only its **impact if it were to occur**.

   Again, you should try to refer to documented instances of your threats that have occurred in the past and the impacts that those attacks had, to justify your reasoning where such examples exist.

4. [**3 marks**] Based on the assessment of the impact of each threat, derive a corresponding set of **security requirements** for the system that would address or mitigate that threat, if that threat can be mitigated. Any threats that you believe **cannot be reasonably mitigated** should be explained clearly. This includes detailing why certain mitigation strategies are **not applicable** based on the assumptions you are making (which should be documented). For example, **insider threats due to human intent cannot be fully eliminated** — even with strict access controls and monitoring, a determined insider may still find ways to misuse their privileges.

   List for each threat the security requirements that are needed to mitigate it. Identify which trust boundary will be strengthened as a result of implementing this security requirement. As before, try to be specific. If you think authentication needs to be employed in a certain part of the system, what kind of authentication and why? To work that out it may help to look at your other threats too—perhaps by employing a certain kind of authentication you can mitigate multiple threats at once.

   Number each of your security requirements that you derive. That way, if one security requirement helps to address multiple threats, you don't need to repeat it.

   Note To get full marks in this part of the assignment you should expect to have to do some research on appropriate security mitigations to defend against certain threats.

   Hint: This is not a subject about basic authentication methods, or specific kinds of encryption protocols, or other specific defence mechanisms relevant to this system. However, we do expect you to be able to do some research on different kinds of security mitigation methods available for different kinds of threats, to choose the best one under the threats you have identified and the assumptions you have made in your analysis.

   As a graduate student, we expect to see real creativity and a desire to push the boundaries of your knowledge. This assignment is designed to be an ideal opportunity to educate yourself on security threats and defences, as relevant to this kind of system.

6 Marking Criteria

There is not a set of right or wrong answers for this assignment. Instead, it is testing your ability to understand and apply the concepts presented in lectures about security and safety engineering. If you think that some of the requirements are ambiguous, then you should decide on an appropriate interpretation and, very importantly, you should document what your interpretation was. That way, you cannot be penalised for making an assumption that is different to what I or the markers had in mind.

You are also free to discuss the requirements on the Ed forum, especially where you think they are ambiguous, to help clarify them. If you are not sure about whether a particular threat is realistic, or how an attacker might exploit a particular threat, or you are not sure what kind of security mitigation to require to defend against it, you are free to email Hira or your tutor to discuss or ask during a consultation session. Please try to avoid giving away information about your potential threats and security requirements while discussing with other students, especially on the Ed forum. Ask the subject staff in the first instance.

7 Submission

One member of your pair should create a PDF file named assignment_pair_N.pdf, where N is your group number, containing your joint answers to the questions. Submit this file via your Assignment Pair group on Canvas.

8 Communication Rules

You may discuss the questions freely within your pair, and write up your joint answers together. You may also consult any other materials you find on the Internet (or in the library), as long as you give proper references in your report. You may *not* discuss this with anyone other than your project partner, except to clarify the requirements of the assignment. In particular, cross-pair collaboration is not allowed. However, you may ask or answer any question you like on the LMS discussion board—this is up to you. You may share answers or raise interesting questions if you like, for the benefit of all. This allows ideas to be shared but mitigates the (unfair) advantage of having clever friends.

9 Late submissions

Please submit on time. It's much better to submit a not-quite-finished version on time than a perfect version late. 1 mark will be deducted each day (or part thereof) after the submission deadline.

If you require an extension, please refer to the document FEIT Extensions and Special Consideration listed in the **Welcome** module on Subject Canvas. Ensure you follow the official procedure and submit your request through the appropriate channels.

10 Academic Misconduct

The University Academic Integrity Policy MPF1310 (`https://policy.unimelb.edu.au/MPF1310`) applies to this assignment. What you submit must be your own work, and where you use the work of others it must be appropriately acknowledged (e.g. by citation).

The subject staff take plagiarism very seriously. In the past, we have successfully prosecuted several students that have breached the university policy. Often this results in receiving 0 marks for the assessment, and in some cases, has resulted in failure of the subject.

Quoting from the University's Statement on the use of AI in assignments (https://academicintegrity.unimelb.edu.au/plagiarism-and-collusion/artificial-intelligence-tools-and-technologies):

> If a student uses artificial intelligence software such as ChatGPT or QuillBot to generate material for assessment that they represent as their own ideas, research and/or analysis, they are NOT submitting their own work. Knowingly having a third party, including artificial intelligence technologies, write or produce any work (paid or unpaid) that a student submits as their own work for assessment is deliberate cheating and is academic misconduct.
>
> If a student uses AI generated material in the preparation of their assessment submission, this must be appropriately acknowledged and cited in accordance with the Assessment and Results Policy (MPF1326) https://policy.unimelb.edu.au/MPF1326.