Script: Introduction

(Slide 1)
Good morning, respected members of the jury and fellow attendees.
My name is **[Your Name]**, and today, on behalf of my team, I am pleased to present our internship project titled **Cyber Risk Assessment & Threat Intelligence Platform**, developed using Python.

In today’s digital landscape, a cyberattack occurs approximately every 39 seconds. Traditional security solutions often stop at detection, whereas modern organizations require **proactive risk management**. With this motivation, our objective was to design a platform that not only identifies vulnerabilities but also helps organizations understand, prioritize, and mitigate cyber risks effectively.

Script: Problem statement

(slide 2)
This leads to an important question: *Why do we need another security tool?*

Currently, organizations rely on multiple scanners and intelligence sources that generate fragmented and highly technical outputs. These results are difficult to correlate and are often retrospective in nature. As a result, security teams are forced into a reactive cycle, making it challenging to anticipate and prioritize future threats.

Script: Solution statement

(slide 3)
To address this challenge, we developed a **Python-based Cyber Risk Assessment and Threat Intelligence platform**.

Our solution integrates **network scanning, threat intelligence enrichment, and AI-driven summarization** into a unified system. Unlike traditional tools that merely list vulnerabilities, our platform correlates data from multiple trusted intelligence sources and prioritizes risks based on real-world exploitability.

Additionally, it explains complex technical findings in **natural language** through an interactive dashboard, enabling both technical and non-technical stakeholders to make informed decisions.

Script: Layer 1 (Scanning engine)

(slide 4)
The foundation of our platform is the **Vulnerability Scanning Engine**, built using **Nmap**. This layer serves as the primary entry point of the system. It scans IP addresses, domains, or CIDR blocks to identify exposed assets within a target environment.

(slide 5)
The scanning engine detects live hosts, open ports, and, most importantly, the specific versions of running services. Identifying service versions is critical for mapping known vulnerabilities. All scan results are structured into **JSON format**, ensuring seamless integration with subsequent layers of the platform.

Script: Layer 2 (Threat Intel)

(slide 6)
However, raw scan data alone is not sufficient. Security findings require **context**, which is provided by our Threat Intelligence Engine.

(slide 7)
We designed a **three-tier intelligence model** that delivers comprehensive visibility:

- Exposure Intelligence
- Malware Intelligence
- Vulnerability Intelligence

(slide 8)
Exposure Intelligence leverages **Shodan** to identify assets exposed to the public internet.
Malware Intelligence uses **VirusTotal** to analyze the reputation and historical behavior of IPs and domains.
Vulnerability Intelligence enriches CVE data using **NVD, Vulners, and CISA KEV**, allowing us to prioritize vulnerabilities that are actively exploited in the real world.

These insights are then forwarded to our **Risk Scoring and AI summarization engine**.

Script: Layer 3 (Risk Scoring & AI summarization)

(slide 9)
This layer can be considered the brain of the platform, as it determines the overall risk posture and helps anticipate potential future threats.

(slide 10)
Rather than relying solely on traditional CVSS scores, we enhance risk evaluation by incorporating **EPSS (Exploit Prediction Scoring System)**. This enables us to rank vulnerabilities based on their likelihood of exploitation in real-world scenarios, rather than just theoretical severity.

(slide 11)
The platform converts raw technical data into **actionable insights** for decision-makers.
Through a **Natural Language Query Interface**, users can clarify doubts and interactively explore risks.
The summarized risk reports are transmitted to the dashboard via **REST APIs**, ensuring smooth data communication between layers.

Script: Layer 4 (Dashboard engine)

(slide 12)
Next, we move to the **Dashboard Layer**, which acts as the user’s primary interaction and remediation interface. It serves as a centralized view that visualizes outputs from all backend layers in a unified manner.

(slide 13)

A key design principle of our architecture is **separation of concerns**.
The frontend is dedicated entirely to user experience and visualization, while the backend handles resource-intensive scanning and analysis operations independently.
Technologically, the platform uses **Python** as the core language, **Streamlit** for the user interface, **Pandas** for data transformation, **Plotly** for interactive visualizations, and **RESTful APIs implemented using FastAPI** for backend communication.

(slide 14)
From a workflow perspective, once a user submits a scan target—such as an IP address, domain, or file—the frontend sends a request to the backend API. The scan is executed by the scanning engine, enriched by subsequent intelligence layers, and finally returned to the dashboard where the consolidated results are displayed to the user.

Script: Screenshots of the platform

(slide 15)

Here, we present a few screenshots of the platform.
The first image shows the detailed output generated after a scan, while the second illustrates the global exposure and distribution of detected assets.

Script: Future plans on CRATIP

(slide 16)
Looking ahead, the future scope of this project includes:

- Personalized risk profiling for users
- Real-time monitoring and alert mechanisms
- Integration of automated threat response capabilities

These enhancements aim to further strengthen the platform's proactive defense capabilities.

Script: Thank you

(slide 17)
Thank you for your time, patience, and attention.
We also extend our gratitude to our teammates for their collaboration and dedication.

We are happy to take any questions.