Problem Statement 1:

"VishGuard – Voice Pattern & Call Behavior Analyzer for Social Engineering Attack Prevention"

Challenge:

Design a system that analyzes **voice characteristics, speech patterns, and call behavior** to **detect potential social engineering attacks**, such as impersonation of executives, bank officials, or tech support scammers.

Objectives:

- Analyze **voice tone, emotion, speech cadence**, and anomalies to flag suspicious calls.
- Identify **behavioral red flags** like urgency, authority abuse, or information probing.
- Use **AI/ML models** to detect repeated scam patterns or deepfake voice signatures.

Key Features to Implement:

- Voice fingerprinting to match against known scam profiles or deepfake models.
- Real-time transcription + NLP analysis to flag risky intent (e.g., asking for OTPs, credentials).
- Integration with VoIP/Telephony platforms for corporate environments.
- Alert mechanism for suspicious calls and potential vishing attacks.

Bonus Points:

- Develop a mobile app or enterprise dashboard for scam call reporting.
- Build a training module that **educates users on live threats based on recent calls**.
- Create an API for law enforcement or SOC teams to upload flagged audio samples.

Problem Statement 2:

"Detecting Fake Crypto Investment Apps"

Challenge:

Build a solution that can identify **fake cryptocurrency apps** pretending to be real platforms. These apps trick users into buying or investing in Bitcoin or other cryptocurrencies, but later **block access or prevent withdrawal**, leading to financial loss.

Goal:

- Spot fake or cloned apps using app data, suspicious behavior, or user reviews.
- Alert users before they download or invest through such apps.
- Help users verify whether a crypto investment app is safe or not.

Bonus:

- Create a public tool where users can check or report suspicious crypto apps.
- Compare app UI and permissions with real apps to catch fakes.
- Work with real crypto platforms to maintain a list of trusted apps.

Problem Statement 3:

"Catch Me If You Can – Real-Time Phishing Attack Detection"

Challenge:

Design a browser extension or email scanning tool that detects and blocks phishing attempts in real time using AI, threat intelligence, and known malicious behavior patterns.

Objectives:

- Analyze email headers, URLs, and content for phishing indicators.
- Detect domain spoofing, lookalike links, and suspicious attachments.
- Prevent credential harvesting before the user interacts with the phishing page.

Key Features to Implement:

- Integration with threat intel feeds (e.g., VirusTotal, AbuseIPDB).
- ML model to classify phishing emails and websites.

- Real-time browser alert with visual warning and threat score.

Bonus Points:

- Identify credential-harvesting websites by analyzing HTML DOM and login form behavior.
- Maintain a personal threat history for the user.
- Add phishing simulation capability for user training.

Problem Statement 4:

"Dark Web Threat Intel Engine"

Challenge:
Build an automated system to scan dark web forums, marketplaces, and paste sites to extract threat intelligence such as leaked credentials, malware samples, or emerging tactics.

Objectives:

- Gather data from hidden services (.onion) and monitor threat actors.
- Extract Indicators of Compromise (IoCs), CVEs, or exploit mentions.
- Tag and correlate data with known cybercriminal activities.

Key Features to Implement:

- Tor-based crawler with search and filtering capability.
- NLP for multilingual and slang-heavy content understanding.
- Threat heatmap and alert system based on topic spikes.

Bonus Points:

- Integrate with SIEM or TIP platforms via API.
- Map content to MITRE ATT&CK tactics or OWASP vulnerabilities.
- Alert on targeted industry mentions (e.g., BFSI, healthcare).

Problem Statement 5:

"Secure the Smart City – IoT Honeypot Challenge"

Challenge:
Develop an IoT honeypot framework that mimics real smart city devices to attract attackers and analyze their behavior in a controlled environment.

Objectives:

- Emulate vulnerable IoT devices (e.g., CCTVs, traffic lights, sensors).
- Collect logs of attack vectors, exploits, and attacker movements.
- Generate threat intelligence from real-world attacks.

Key Features to Implement:

- Low-interaction and high-interaction honeypots with logging.
- Dashboard to visualize attacker sessions and TTPs.
- Alerting system for newly seen attack patterns.

Bonus Points:

- Generate automatic IoCs from logs.
- Export alerts in STIX/TAXII format for SOC use.
- Integrate with SIEM solutions like Splunk or Azure Sentinel.

Problem Statement 6:

"The Fraud Buster – Credit Card Scam Pattern Identifier"

Challenge:
Create a machine learning-based system that detects financial fraud by analyzing transactional patterns in e-commerce or banking systems.

Objectives:

- Identify anomalies that deviate from normal user spending.
- Flag fraudulent behaviors such as card testing, location mismatches, or bot-driven purchases.

- Correlate user behavior over time to improve accuracy.

Key Features to Implement:

- Supervised or unsupervised ML models for anomaly detection.
- Risk scoring engine with real-time alert generation.
- Visualization dashboard for fraud analysts.

Bonus Points:

- Heatmaps for fraud locations or merchants.
- Integration with payment gateways or fintech APIs.
- Simulated fraud test suite with synthetic datasets.

Problem Statement 7:

"Generative AI Cybercrime – Detect Deepfake & AI Abuse"

Challenge:
Develop tools that detect malicious use of AI, such as deepfake audio/video or generative AI-based scams used in social engineering and impersonation attacks.

Objectives:

- Identify manipulated media used for scams or impersonation.
- Detect deepfake voices used to imitate executives or public figures.
- Provide visual/auditory cues to warn users.

Key Features to Implement:

- Deepfake detection models using facial landmarks or audio spectrograms.
- Browser or video-conferencing integration to flag real-time deepfake risk.
- Explainable AI output for user trust.

Bonus Points:

- Build a real-time Zoom/Meet plugin to detect deepfakes in virtual meetings.
- Include phishing or scam phrase detection in transcripts.
- Add watermarking or integrity verification for trusted content.

Problem Statement 8:

"Cybercrime Evidence Vault – Chain of Custody Automation"

Challenge:
Build a blockchain or distributed ledger-based system to securely log, store, and trace the lifecycle of digital evidence from collection to court presentation.

Objectives:

- Ensure tamper-proof timestamping and evidence verification.
- Maintain full audit logs of access and movement.
- Generate legally admissible reports for investigators.

Key Features to Implement:

- Evidence ingestion with hash validation (SHA256/SHA512).
- Smart contract-based access control and logs.
- Visual timeline of evidence handling and user actions.

Bonus Points:

- Integration with forensics tools like Autopsy or FTK Imager.
- Exportable chain-of-custody reports in PDF and JSON.
- Real-time alert if evidence is accessed or tampered with.

Problem Statement 9:

"Hack the Misinformation – Social Media Threat Analyzer"

Challenge:
Design a system to detect and analyze coordinated disinformation or manipulation campaigns on platforms like Twitter, Facebook, or Telegram.

Objectives:

- Identify fake accounts, bot clusters, or coordinated posts.
- Detect spreading of harmful narratives, propaganda, or election interference.
- Classify misinformation types (health, politics, scams, etc.).

Key Features to Implement:

- NLP and sentiment analysis of social media content.
- Bot detection algorithms using graph analytics.
- Visualization of message propagation and influence paths.

Bonus Points:

- Link fake accounts across multiple platforms.
- Alert system for trending misinformation spikes.
- Create educational tools to help users spot fake news.

Problem Statement 10:

"BitScan – Bitcoin Scam Pattern Analyzer for Investment Fraud Detection"

Challenge:
Build a solution that detects **fraudulent Bitcoin investment schemes** conducted via fake apps, rogue wallets, or suspicious transactions. Victims often invest in crypto through cloned apps or manipulated wallets, only to find their funds irretrievable.

Objectives:

- Analyze transaction flows across public blockchain networks to flag high-risk patterns.
- Identify fake or cloned apps/wallets associated with investment fraud.
- Detect behaviors such as mixing services, rapid fund splitting, and scam wallet reuse.

Key Features to Implement:

- Blockchain graph analytics to visualize and trace suspicious wallet activity.
- Risk scoring engine for Bitcoin addresses or smart contracts.
- Data enrichment from scam reports and block explorers.

Bonus Points:

- Public portal or browser extension for users to check wallet reputation.
- Integration with crypto exchanges or DeFi platforms for real-time flagging.
- ML model to identify new scam typologies based on wallet clusters.

Problem Statement 11:

"ForensicVault – Digital Forensics Toolkit for Cybercrime Investigation"

Challenge:
Develop a modular digital forensics toolkit to help investigators collect, preserve, and analyze evidence from compromised systems, mobile devices, or cloud accounts during cybercrime investigations.

Objectives:

- Enable memory capture, disk imaging, file system parsing, and log analysis.
- Ensure evidence integrity via hashing and timestamping.
- Provide readable forensic reports and event reconstruction.

Key Features to Implement:

- Forensic modules for Windows, Linux, Android, and cloud logs (Azure, AWS).
- File carving, deleted file recovery, and keyword search engine.
- GUI dashboard with timeline view and user interaction replay.

Bonus Points:

- Chain-of-custody logging integrated with blockchain or secure audit trail.
- Generate legally admissible PDF reports for law enforcement.
- Preloaded investigation templates (e.g., ransomware, insider threat, phishing).

Problem Statement-14.

Child Exploitation & Grooming Pattern Detector for Social Media

Design a system to detect grooming patterns and child exploitation attempts on social media platforms using language modeling, chat monitoring, and behavioral indicators.

Objectives:

- Monitor and analyze real-time chat patterns or comments for signs of grooming.
- Detect inappropriate conversations, predator-like behavior, or solicitation attempts.
- Flag high-risk accounts or conversations for review by law enforcement.

Key Features to Implement:

- NLP models trained on grooming-related conversations.
- Age-gap detection and behavioral scoring engine.
- Dashboard for investigators to review, tag, and escalate cases.

Bonus Points:

- Integration with social media platforms or parental control apps.
- Generate redacted but court-admissible evidence bundles.

Problem Statement 15:

Voiceprint Authentication & Spoofing Detector

Create a system that verifies voice identity (voice biometrics) and detects voice spoofing or deepfake audio attacks.

Objectives:

- Identify and authenticate legitimate users through voiceprints.
- Detect altered, replayed, or synthetically generated voices.

Key Features to Implement:

- Voice biometric engine using MFCC and speaker embeddings.
- Spoofing classifier for deepfake or replay detection.

- Real-time alerting and logging.

Bonus Points:

- Integration with mobile apps or IVR systems.
- Use in secure military or field communications.

Problem Statement 16:

Counter-Terrorism OSINT Analysis & Threat Actor Profiling

Develop a system that scans OSINT sources (social media, forums, public data) to detect early signs of terrorism-related activities.

Objectives:

- Extract and analyze posts, images, or metadata to identify radicalization indicators.
- Profile threat actors, affiliations, and regional influence.

Key Features to Implement:

- NLP engine for intent classification and hate/radical keyword matching.
- Entity and relationship mapping to visualize networks.
- Alert system for newly detected high-risk users or groups.

Bonus Points:

- Geo-fencing and multi-language support.
- Integration with law enforcement dashboards.

Problem Statement 17:

Drone Traffic Anomaly Detection & Geo-Fencing Breach Monitor

Build a system that monitors and detects unauthorized or suspicious drone activities near restricted areas.

Objectives:

- Detect geo-fence breaches and illegal drone entries.
- Identify unusual flight behavior (hovering, circling, evasive movements).

Key Features to Implement:

- Real-time drone telemetry feed parsing and behavior analysis.
- Geo-fence management and alerting system.
- Visualization dashboard for drone flight paths.

Bonus Points:

- Simulate jamming or interception scenarios.
- Integrate with radar or RF-based detection hardware.

Problem Statement 18:

AI-Powered Border Surveillance Intrusion Predictor

Create an AI-driven surveillance solution to detect and predict illegal intrusions along international or state borders.

Objectives:

- Monitor real-time video feeds or motion sensors.
- Predict and alert on high-risk intrusion patterns.

Key Features to Implement:

- Computer vision for human and vehicle movement tracking.
- ML model to identify patterns of smuggling or unauthorized entry.
- Time-series heatmap and alert system.

Bonus Points:

- Night-vision and low-light model support.
- Integration with satellite data or UAV cameras.

Problem Statement 19:

Dark Web Arms Trade & Radical Content Detector

Design a solution that scans dark web marketplaces, forums, and encrypted chat channels for arms trade, explosives, or extremist propaganda.

Objectives:

- Identify listings or messages related to weapons, training material, or recruitment.

- Track seller aliases, cryptocurrency wallets, and transaction chains.

Key Features to Implement:

- Onion crawler with keyword and metadata filtering.
- Dark web language model for decoding obfuscated content.
- Risk categorization and trend analysis panel.

Bonus Points:

- Integration with national threat databases.
- Create an alert mechanism for law enforcement when new listings appear.

Problem Statement 20:

Secure Digital Evidence Collection Kit for Field Investigators

Build a portable toolkit (software prototype) for law enforcement or military officers to collect, preserve, and validate digital evidence from suspect devices in the field.

Objectives:

- Enable quick capture of files, logs, and metadata.
- Ensure hash-based validation and tamper-proof storage.

Key Features to Implement:

- USB-bootable forensic OS or mobile app interface.
- File hashing, metadata tagging, and case linking.
- Export options in PDF or JSON formats.

Bonus Points:

- Chain-of-custody ledger or blockchain support.
- Offline operation mode for remote areas.

Problem Statement 21:

Targeted Disinformation Detector for Defense Personnel Channels

Develop a tool that detects attempts to spread misinformation specifically targeting military or defense personnel through emails, forums, or internal channels.

Objectives:

- Detect content crafted to lower morale, influence decision-making, or leak sensitive info.
- Analyze campaign coordination and source credibility.

Key Features to Implement:

- NLP engine for detecting manipulated narratives and psychological ops.
- Source reputation scoring and fake persona detection.
- Visual storyline builder showing disinformation propagation.

Bonus Points:

- Integration with defense communication tools.
- Archive flagged messages and misinformation samples for training.