TRACK 6 — CYBER SECURITY
Problem Statements

HTH-CS-01: Explainable Phishing & Malicious URL Triage System

Category	Details
Industry	Enterprise Security Operations
Scenario	Security teams receive far more reported phishing emails than they can manually verify, and users distrust binary "safe/unsafe" verdicts without reasoning.
Problem Statement	Analyze emails/URLs and flag phishing attempts using domain reputation, spoofing indicators, mismatched links, and urgency-language patterns, with a clear explanation per flag, then prioritize a SOC review queue under limited analyst capacity by estimated blast-radius (e.g., how many employees received similar emails).
What Teams Should Build	Multi-signal phishing classifier, explanation generator, capacity-constrained review-queue ranking weighted by estimated exposure/blast-radius.
24-Hour MVP	Public phishing-email/URL datasets, classify + explain, simulate a multi-recipient scenario to demonstrate blast-radius-weighted prioritization under a fixed analyst-review capacity.
Data/Input	Public phishing datasets (e.g., PhishTank, public phishing email corpora).
Key Constraints	Every verdict must include explanation, not just a label; review capacity fixed and small relative to alert volume.
Expected Output	Prioritized SOC review queue with per-case explanation and estimated exposure.
Evaluation Criteria	Detection accuracy, explanation quality, exposure-aware prioritization.
Bonus Features	Auto-generated employee warning broadcast for high-blast-radius campaigns.

HTH-CS-02: Risk-Tiered Secure Authentication Platform with Session Governance

Category	Details
Industry	Enterprise IT Security
Scenario	Organizations need MFA and secure login, but also need visibility/control over active sessions to respond to compromised-credential incidents quickly.
Problem Statement	Build a login system with proper password hashing, rate-limiting, and TOTP-based MFA, plus a risk-tiering layer that requires step-up authentication for unusual login context (new device/location) and a session-governance dashboard for revoking access during an incident.
What Teams Should Build	Secure auth backend (bcrypt/argon2 hashing, rate limiting, TOTP MFA), contextual risk-scoring for step-up auth triggers, session management/revocation dashboard.
24-Hour MVP	Working login+MFA flow, simulated “new device/location” login triggering step-up challenge, admin dashboard listing/revoking active sessions.
Data/Input	Self-generated test accounts/sessions; simulated login-context metadata (IP/device fingerprint).
Key Constraints	Passwords must never be stored in plaintext or with weak hashing; rate-limiting must be demonstrably effective against brute-force in the demo.
Expected Output	Working secure login system + session dashboard with revocation capability and audit log.
Evaluation Criteria	Security correctness (hashing, rate-limiting, MFA implementation), risk-based step-up logic, session-governance usability.
Bonus Features	Device/session anomaly scoring feeding into the risk-tiering engine.

HTH-CS-03: Business-Risk-Ranked Vulnerability Scanner & Remediation Planner

Category	Details
Industry	Enterprise Security / IT Operations
Scenario	Vulnerability scanners produce long flat lists that overwhelm small security teams who can only remediate a handful of issues per sprint.
Problem Statement	Scan a given IP/domain (lab environment) for open ports, outdated services, and misconfigurations, then rank findings by actual business risk (exploitability × asset criticality × exposure), not just CVSS score alone, and generate a remediation plan sized to a limited weekly remediation capacity.
What Teams Should Build	Scanning pipeline (using standard open-source scanning libraries against a lab/sandboxed target), risk-scoring model incorporating asset-criticality metadata, capacity-constrained remediation-plan generator.
24-Hour MVP	Scan a controlled lab environment (self-hosted vulnerable VM/sandbox), generate a risk-ranked report and a remediation plan under a fixed weekly capacity (e.g., 5 fixes/week).
Data/Input	Self-hosted intentionally-vulnerable lab targets only (e.g., DVWA, Metasploitable) — never scan systems without authorization.
Key Constraints	Must only scan authorized/lab targets; ranking must use business risk, not raw CVSS/port-count alone; remediation plan must respect fixed capacity.
Expected Output	Risk-ranked vulnerability report + capacity-constrained remediation roadmap.
Evaluation Criteria	Scan correctness, quality of business-risk ranking logic, realistic remediation planning.
Bonus Features	Trend tracking of risk posture over multiple simulated scan cycles.

HTH-CS-04: Tamper-Evident File Integrity Monitor with Incident Timeline

Category	Details
Industry	IT Security / Compliance
Scenario	Regulated organizations must prove exactly what changed, when, and by inference how severe it was, not just that "something changed."
Problem Statement	Monitor a folder/system files for unauthorized modification/deletion/addition using cryptographic hash comparison, and build an incident timeline that correlates multiple file changes into a single "event" with a severity estimate (e.g., many config files changed together = higher severity than one log file).
What Teams Should Build	Hash-based integrity monitoring (SHA-256), change-event correlation engine (grouping related changes in time/location), severity-scoring logic, timeline visualization.
24-Hour MVP	Monitor a sample directory structure, simulate a multi-file tamper event, generate a correlated incident timeline with severity estimate.
Data/Input	Self-created sample file/folder structures with simulated tamper scripts.
Key Constraints	Must correlate related changes into events, not just list every diff independently; must be tamper-evident (log itself should be protected/append-only).
Expected Output	Incident timeline dashboard with grouped events, severity, and full change audit log.
Evaluation Criteria	Detection accuracy, correlation/severity logic quality, log integrity design.
Bonus Features	Rollback-suggestion feature showing last-known-good state per file.

HTH-CS-05: Metadata-Minimized End-to-End Encrypted Team Communication Tool

Category	Details
Industry	Enterprise Communications / Privacy Tech
Scenario	Organizations handling sensitive information need messaging where even the server operator cannot read content, with defensible key-exchange design.
Problem Statement	Build a chat application with client-side encryption using a standard cryptographic library, secure key exchange without hardcoded keys, and a clear demonstration/explanation of why the server cannot decrypt messages — including how key exchange resists a basic man-in-the-middle scenario.
What Teams Should Build	Client-side E2E encryption implementation (using established libraries, not custom crypto), secure key-exchange protocol (e.g., Diffie-Hellman-based), demo proving server-side content is unreadable.
24-Hour MVP	Working chat between 2+ simulated clients with E2E encryption, server-log demonstration showing only ciphertext stored, basic key-exchange security explanation/demo.
Data/Input	Self-generated test conversations; no external data required.
Key Constraints	Must use established cryptographic libraries, never custom/home-rolled crypto primitives; keys must not be hardcoded or transmitted in plaintext.
Expected Output	Working E2E encrypted chat + technical writeup proving server cannot read content.
Evaluation Criteria	Correct use of established crypto, key-exchange soundness, clarity of the “server can’t read this” demonstration.
Bonus Features	Forward secrecy (session-key rotation) implementation.

HTH-CS-06: Privacy-Preserving Breach & Credential Exposure Checker

Category	Details
Industry	Consumer Security / Identity Protection
Scenario	Users want to know if their credentials are compromised without ever transmitting the actual password to a third party.
Problem Statement	Build a tool where users check if their email/password has appeared in known breaches using a privacy-preserving method (e.g., k-anonymity hash-prefix lookup, as used by real breach-check APIs) so raw passwords are never transmitted, and generate a personalized remediation checklist based on exposure severity.
What Teams Should Build	k-anonymity-based breach lookup integration (using a public breach API), exposure-severity classification, personalized remediation-action generator (which accounts to prioritize changing first).
24-Hour MVP	Working lookup tool using k-anonymity method against a public breach-check API, severity classification of results, generated remediation checklist.
Data/Input	Public breach-check API (e.g., Have I Been Pwned's k-anonymity endpoint) — no raw password transmission.
Key Constraints	Raw password must never leave the client in plaintext or full-hash form — must use partial-hash/k-anonymity approach; must explain the privacy mechanism to the user.
Expected Output	Exposure report with severity tiers and a prioritized remediation checklist.
Evaluation Criteria	Correct privacy-preserving implementation, clarity of severity/remediation logic, user-facing explanation quality.
Bonus Features	Password-reuse pattern detection across multiple checked accounts (without storing raw passwords).

HTH-CS-07: Explainable Insider-Threat Behavioral Anomaly Detector

Category	Details
Industry	Enterprise Security / Insider Risk
Scenario	Security teams need to catch compromised or malicious insider activity, but black-box anomaly scores get ignored or distrusted by analysts.
Problem Statement	Analyze simulated login/activity logs (times, locations, access patterns) to flag unusual user behavior, with every flag showing the specific rule/pattern violated (e.g., "access from new country + off-hours + sensitive file access") rather than an opaque anomaly score, then prioritize flags by combined signal strength under limited investigator capacity.
What Teams Should Build	Multi-signal behavioral baseline □odelling per user, explainable rule/pattern-based flagging (not black-box score alone), capacity-constrained investigation-queue ranking.
24-Hour MVP	Simulated multi-user activity-log dataset with injected insider-threat scenarios, explainable flagging demo, prioritized queue under a fixed investigator capacity.
Data/Input	Simulated/synthetic user-activity log dataset (documented generation) or public insider-threat datasets (e.g., CERT Insider Threat dataset).
Key Constraints	Every flag must show explainable contributing factors, not a single opaque score; investigator capacity fixed and small.
Expected Output	Explainable, prioritized investigation queue with per-flag contributing-factor breakdown.
Evaluation Criteria	Detection quality on injected scenarios, explainability, realistic capacity-aware prioritization.
Bonus Features	Per-user behavioral baseline drift visualization over time.

HTH-CS-08: Measurable Security Awareness & Phishing Simulation Platform

Category	Details
Industry	Enterprise Security Training / Risk Management
Scenario	Organizations run phishing simulations but often can't translate click-rate data into a defensible, quantified organizational risk metric or targeted training plan.
Problem Statement	Build a platform that sends simulated phishing emails (to consenting test users), tracks click/response/report behavior, follows up with a targeted awareness quiz calibrated to the user's specific mistake, and produces a quantified departmental risk score with a training-priority plan under limited training-budget/time.
What Teams Should Build	Simulated phishing campaign engine, behavior tracking, mistake-targeted quiz generation, department-level risk scoring, capacity-constrained training-priority planner.
24-Hour MVP	Simulated campaign to test users (self-contained test group), tracked behavior dashboard, targeted quiz demo, department risk score + training plan under a fixed training-hours budget.
Data/Input	Self-generated test-user group and synthetic click/response data; template phishing content used only for internal, consented testing.
Key Constraints	Must be used only on consenting test users in a controlled demo, never real unconsented targets; training-plan must be prioritized under a fixed budget, not "train everyone equally."
Expected Output	Risk dashboard by department + prioritized, budget-constrained training plan.
Evaluation Criteria	Realistic simulation design, quality of risk quantification, sensible budget-constrained prioritization, ethical framing.
Bonus Features	Longitudinal risk-trend tracking across repeated simulation rounds.

HTH-CS-09: Utility-Preserving PII Anonymization & Re-Identification Risk Auditor

Category	Details
Industry	Data Privacy / Compliance
Scenario	Organizations need to share/analyze datasets containing PII, but naive masking either destroys analytical utility or still leaves re-identification risk (e.g., via quasi-identifier combinations).
Problem Statement	Build a tool that anonymizes/masks PII (names, emails, phone numbers) while preserving analytical usability, and additionally audits residual re-identification risk from quasi-identifier combinations (e.g., zip+age+gender), reporting a risk score rather than assuming anonymization = safe.
What Teams Should Build	PII detection/masking pipeline, utility-preservation techniques (e.g., generalization, tokenization), k-anonymity-style re-identification risk scoring on the resulting dataset.
24-Hour MVP	Public/synthetic dataset with PII fields, anonymization pipeline + before/after utility comparison + re-identification risk report highlighting risky quasi-identifier combinations.
Data/Input	Public/synthetic datasets containing PII-like fields (self-generated to avoid real personal data exposure).
Key Constraints	Must quantify and report residual re-identification risk, not just perform masking and declare success; must preserve some measurable analytical utility (not just delete all fields).
Expected Output	Anonymized dataset + utility-comparison report + re-identification risk audit.
Evaluation Criteria	Anonymization technique soundness, quality of risk audit, utility-preservation demonstration.
Bonus Features	Reversible pseudonymization mode with strict authorized-access controls and access-audit logging.

HTH-CS-10: Signal-Fused Intrusion Detection & Response Prioritization Dashboard

Category	Details
Industry	Security Operations Center (SOC)
Scenario	SOC analysts are flooded with raw log alerts (failed logins, port scans, traffic spikes) from disconnected tools and struggle to see which combination of signals represents a real, urgent incident.
Problem Statement	Monitor simulated network/system logs in real time, fuse multiple weak signals (repeated failed logins + port scan + traffic spike from the same source) into higher-confidence composite incidents, assign severity, and prioritize analyst response under a fixed shift capacity.
What Teams Should Build	Simulated multi-source log stream, signal-fusion/correlation engine, severity scoring for composite incidents, capacity-constrained analyst-response queue.
24-Hour MVP	Simulated log stream with injected multi-stage attack scenarios (recon → brute force → exfil attempt), live dashboard fusing signals into composite incidents with severity and ranked response queue under a fixed analyst-capacity limit.
Data/Input	Simulated/synthetic network and system log data (documented generation) or public IDS datasets (e.g., CICIDS2017, NSL-KDD) as raw signal source.
Key Constraints	Must fuse multiple weak signals into composite incidents rather than alerting on each raw log line independently; analyst capacity fixed and small relative to raw alert volume.
Expected Output	Live SOC dashboard with composite incidents, severity, and a capacity-constrained response queue.
Evaluation Criteria	Signal-fusion quality, severity-scoring realism, capacity-aware prioritization, live-demo feasibility.
Bonus Features	Automated containment-action suggestion (e.g., "isolate host X") with human-approval gate.