Introduction and Background

1. Overview of Cyber Threat Intelligence (CTI)

Cyber Threat Intelligence (CTI) refers to the systematic process of collecting, analyzing, and applying information about potential or existing cyber threats.
It enables organizations to understand adversary behavior, anticipate attacks, and take proactive measures to protect their systems, data, and users.

CTI sources can include:

- **Open-source intelligence (OSINT):** public feeds, reports, and repositories.
- **Commercial threat feeds:** paid and proprietary intelligence services.
- **Internal telemetry:** logs, sensors, and past incident reports.

The CTI process generally follows a lifecycle consisting of:

1. **Collection** – Gathering raw data from multiple threat intelligence feeds.
2. **Processing and Normalization** – Cleaning, deduplicating, and structuring data into a unified schema.
3. **Analysis** – Identifying patterns, relationships, and trends among indicators.
4. **Dissemination** – Sharing actionable intelligence with security teams or automated defense systems.

By integrating CTI, organizations can enhance their situational awareness, detect new attack campaigns, and reduce their response time to evolving threats.

---

2. Understanding Phishing Attacks

Phishing is one of the most prevalent and effective cyber attack techniques, aiming to deceive users into revealing sensitive information such as credentials, banking details, or personal data. Attackers typically disguise malicious websites or emails to resemble legitimate entities (e.g., banks, government agencies, or online services).

Common phishing vectors include:

- **Email phishing:** deceptive emails containing malicious links or attachments.
- **Spear phishing:** targeted attacks on specific individuals or organizations.
- **Clone phishing:** modification of legitimate communications to insert malicious links.
- **Smishing and vishing:** phishing through SMS or voice calls.

Phishing attacks continue to evolve, leveraging new domains, compromised infrastructures, and dynamic URLs that make traditional blacklist approaches insufficient.
Hence, **automated phishing detection systems** must rely on continuous **threat intelligence collection** and **real-time analysis** of emerging indicators.

---

3. Role of CTI in Phishing Detection

Cyber Threat Intelligence plays a crucial role in **detecting and mitigating phishing campaigns**. By collecting indicators of compromise (IOCs) such as **malicious URLs, domains, IP addresses, and sender emails**, CTI feeds provide early warnings about active phishing operations.

Integrating multiple CTI sources offers several advantages:

- **Broader coverage:** Each feed (e.g., OpenPhish, PhishTank, URLhaus) contributes unique indicators.
- **Data validation:** Overlapping indicators from multiple sources increase confidence in detection accuracy.
- **Faster response:** Near real-time feed updates enable quicker blocking of malicious domains and URLs.
- **Trend analysis:** Aggregated data allows identification of phishing campaigns targeting specific regions, brands, or industries.

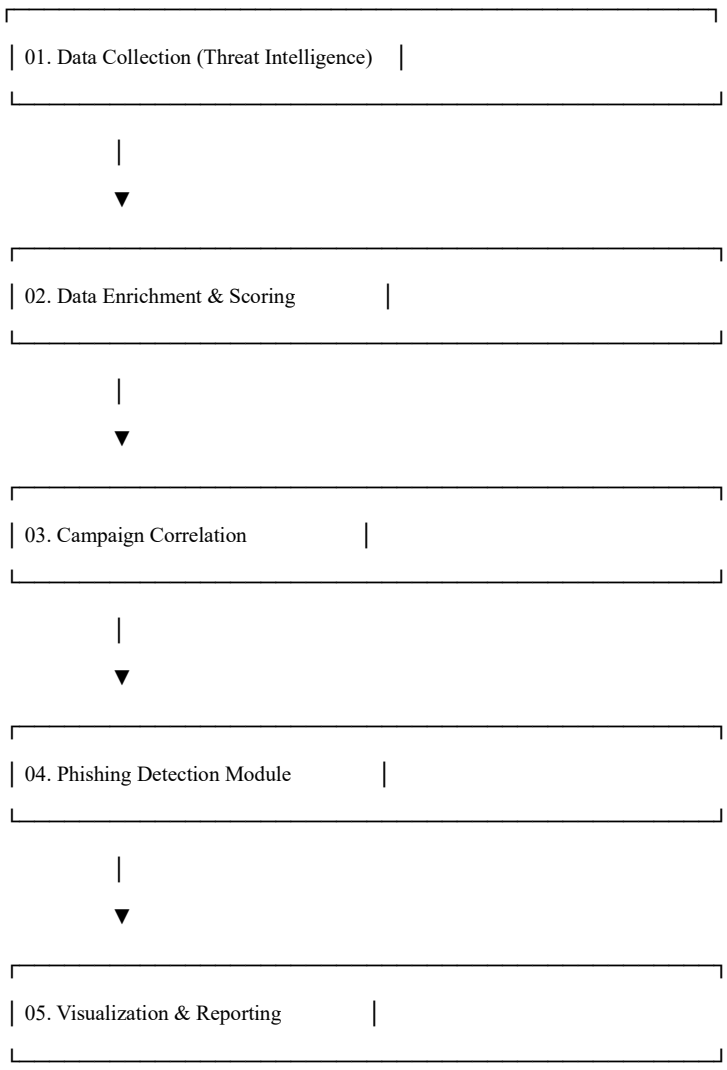
However, raw data from CTI feeds often come in heterogeneous formats (TXT, JSON, CSV) and contain noise or duplicates.
Therefore, an **automated collection and normalization pipeline** is essential to unify and clean the data before further analysis.

System Architecture and Module Descriptions

1. Overview

The proposed **Phishing Threat Intelligence and Detection System** is designed as a **modular, layered architecture** that automates the collection, enrichment, correlation, and detection of phishing indicators.
Each layer contributes a specific function in transforming raw Cyber Threat Intelligence (CTI) data into actionable insights for real-time phishing detection.

The architecture is composed of **five primary modules**, as illustrated in the conceptual workflow below:



For our Project Let's Discuss

Multi-source CTI collection & normalization

01. Data Collection (Threat Intelligence)

Data Collection (Threat Intelligence Layer)

The **Threat Intelligence Layer** is responsible for gathering phishing-related indicators from multiple **open-source threat intelligence feeds**. This layer ensures a continuous and diversified collection of phishing URLs, domains, and related metadata.

1. Multi-source CTI Collection & Normalization

The system collects data from three main open feeds — **OpenPhish**, **PhishTank**, and **URLhaus** Each feed is queried using multiple endpoints and fallback mechanisms to maximize data coverage and resilience against temporary API failures.

a. OpenPhish Collection

The function collect_openphish() retrieves URLs from OpenPhish text feeds (feed.txt) over both HTTP and HTTPS.
It:

- Fetches raw URLs from the OpenPhish feed.
- Cleans and deduplicates entries.
- Randomly samples URLs up to the requested sample_size.
- If the feed returns fewer URLs than required, it **generates synthetic phishing URLs** using the generate_synthetic_urls() function to maintain balanced datasets.
- Adds metadata such as source (OpenPhish) and timestamp (first_seen).

b. PhishTank Collection

The collect_phishtank() function queries PhishTank's JSON feeds (online-valid.json) via multiple endpoints.
It:

- Parses the JSON response to extract phishing URLs.
- Deduplicates and samples URLs.
- Fills any data gaps by generating synthetic URLs labeled as "phishtank_synthetic".

- Normalizes the collected data into a unified structure.

c. URLhaus Collection

The collect_urlhaus() function collects URLs from CSV-based URLhaus feeds (csv_recent and csv_online).
It:

- Parses CSV rows to extract URLs while skipping comment lines.
- Handles network errors gracefully through try/except blocks.
- Deduplicates, samples, and augments the data if needed via synthetic URL generation.
- Outputs a DataFrame with source and timestamp metadata.

d. Synthetic Data Generation

To ensure a consistent dataset size, the generate_synthetic_urls() function creates **realistic phishing-like URLs** when real data is insufficient.
It uses random combinations of:

- Common phishing domain patterns (e.g., login-verify.com, secure-update.com)
- Typical malicious paths (verify.php, update.php, auth.php)
- Parameters that mimic phishing behavior (token=xyz789, redirect=paypal.com)

This process guarantees that the dataset maintains diversity and size even during feed downtime.

Normalization

1 .Synthetic Data Integration

In real-world collection, feeds can occasionally provide **fewer URLs** than the required sample size due to downtime, rate limits, or temporary feed shortages.
To ensure consistent dataset size and structure, the system automatically **generates synthetic phishing-like URLs** using the function generate_synthetic_urls().

Synthetic Data Generation Process

The generator creates realistic phishing URLs by combining:

- **Base domains** mimicking phishing campaigns (e.g., secure-update.com, login-verify.com).
- **Common phishing paths** (e.g., verify.php, update.php, auth.php).

- **Parameters** simulating malicious query strings (e.g., token=xyz789, redirect=paypal.com).
- **Optional subdomains** (secure., login., account.) for higher diversity.

This ensures the dataset maintains a **uniform structure** even when real data is insufficient. Synthetic URLs are labeled using the pattern <source>_synthetic (e.g., openphish_synthetic) to distinguish them during analysis.

---

2. Normalization Pipeline

After collection and synthesis, all data passes through a **normalization layer** that unifies heterogeneous inputs into a single standardized schema.
This process involves:

1. **Schema Alignment** – All sources are mapped to a consistent format:
2. url | source | first_seen
3. **Type Enforcement** – URL strings, timestamps, and sources are cast to standard data types.
4. **Timestamp Assignment** – Each record is annotated with the current collection time (datetime.now().isoformat()).
5. **Source Harmonization** – Whether real or synthetic, data from all feeds shares the same schema, ensuring uniform downstream processing.
6. **Quality Validation** – Empty or malformed URLs are filtered out before final storage.

Through this normalization, the tool integrates **real-world threat intelligence** and **synthetic augmentation** into a single coherent dataset suitable for correlation, enrichment, and classification tasks.

---

4. Reputation Validation (Spamhaus DBL)

To further enrich the normalized dataset, the system includes a **domain-level reputation check** using the **Spamhaus Domain Block List (DBL)**.
Each domain is queried via DNS against Spamhaus servers (9.9.9.9, 1.1.1.1), returning one of:

- "listed" — domain appears in Spamhaus DBL
- "not_listed" — domain is clean
- "error" — lookup failed or timed out

This step adds a valuable layer of **domain reputation intelligence** to the normalized phishing dataset.

Phishing Risk Assessment Criteria

Overview

The **Phishing Risk Scoring Framework** quantifies the malicious potential of each indicator (URL/domain/email) on a **100-point scale**, integrating intelligence trust, brand impersonation, technical indicators, and content-based heuristics.

Each category contributes a weighted portion to the total score, and critical combinations trigger *automatic CRITICAL classification* regardless of numeric value.

🔹 1. Threat Intelligence (25 pts)

Source Type	Score	Rationale
OpenPhish / PhishTank	25 pts	High-confidence community feeds with active verification
Synthetic Data	20 pts	Auto-generated from realistic phishing templates
Internal Feeds	15 pts	Enterprise or honeypot-collected indicators
Unknown / Unverified	0 pts	No validated intelligence source

This layer reflects the reliability of the original intelligence source.

🔹 2. Brand Impersonation (30 pts)

Tier	Example Brands	Score	Reasoning
Tier 1 – Financial Institutions	Chase, PayPal, Bank of America	25 pts	High-value financial targets with real-world loss potential
Tier 2 – FinTech / Crypto	Venmo, Coinbase, CashApp	20 pts	Emerging targets with moderate attack frequency
Tier 3 – Major Tech Companies	Microsoft, Apple, Google	15 pts	Common login impersonation targets
Tier 4 – Generic Security Themes	“login”, “verify”, “security”, “auth”	10 pts	Non-branded but suspicious identity triggers

Brand similarity is calculated using Levenshtein and token-based matching. Tier assignment is based on closest brand detected.

- **3. Technical Infrastructure (25 pts)**

Indicator	Score		Explanation
SSL Mismatch / Self-Signed	15 pts		Certificate CN mismatch or invalid SSL chain
Active DNS Resolution	10 pts		Confirms live phishing infrastructure
Suspicious TLD (.tk, .ml, .xyz, etc.)	8 pts		Low-reputation or disposable domain zones
Long Domain (>40 characters)	10 pts		Common in obfuscation and redirection attacks
Multiple Hyphens (≥3)	8 pts		Pattern often used in typosquatting or brand deception

These features capture anomalies in domain and hosting infrastructure.

- **4. Content Analysis (20 pts)**

Category	Keywords / Patterns	Points	Purpose
Urgency Keywords	“urgent”, “immediately”, “verify now”	4 pts each	Psychological pressure tactics
Security Keywords	“secure”, “locked”, “update info”	3 pts each	Imitates security language
Financial Actions	“payment”, “invoice”, “refund”, “charge”	3 pts each	Financial lure indicators
Account Recovery Pages	Reset / recover / forgot password	12 pts	High-risk identity targeting
Payment Portals	Payment or card submission forms	15 pts	Strong indication of fraud intent

Content-based features are extracted from HTML text, metadata, and form attributes.

Automatic CRITICAL Flags

Some phishing combinations instantly elevate an indicator to **CRITICAL severity** regardless of the numeric score.

Condition	Example
High-value brand + Active DNS + SSL mismatch	https://secure.paypa1-login.com/verify
Payment portal + Urgency keywords	"Your account will be suspended unless you pay now"
Account recovery + Financial brand	"Reset your Chase password now"
Multi-step phishing flow	Redirect chains across multiple domains

These flags override standard scoring to prevent delayed response to confirmed active threats.

Evaluation results on sample data

URL	Risk Score	Risk Level	Key Indicators	Observations
https://verify-identity.com/verify.php?next=login.com	80	**CRITICAL**	"verify" keyword, active domain, recent creation (Feb 2025), SSL mismatch	Simulates identity verification page; likely brand spoofing
https://secure-update.com/login.php?ref=phish	70	**HIGH**	"login", expired SSL (2019), old domain (2009), suspicious keyword	Legacy domain used for phishing login pages
https://verify-identity.com/security.php?user=test	55	**MEDIUM**	"verify", SSL valid, normal DNS	Moderate risk, domain age acceptable but lexical match
https://auth.auth-required.org/update.php	52	**MEDIUM**	"auth", partial WHOIS, missing DNS	DNS missing — possibly inactive domain
https://account.payment-gateway.org/update.php	44	**LOW**	Payment-related keywords, DNS NXDOMAIN	Low risk due to domain nonexistence

5. Campaign Correlation

5.1 Definition

A **phishing campaign** is a coordinated set of malicious activities or phishing indicators (such as URLs, domains, or IPs) that share **common infrastructure, registrars, hosting providers, or SSL certificates**. These shared attributes suggest that the indicators originate from the same threat actor or automated toolkit.

In Cyber Threat Intelligence (CTI), campaign correlation helps analysts:

- Detect **reused phishing infrastructure**.
- Identify **large-scale coordinated attacks**.
- Reduce **false positives** by grouping related indicators.

Example From Our Campaign Correlation

Technical Indicators

- All URLs share the same **base domain**: secure-update.com
- Multiple **subdomains** used (e.g., auth., login., verify., security., payment.)
- URLs include **redirect parameters** to paypal.com, imitating PayPal login or payment pages.
- Shared **SSL fingerprint** across all domains → indicates centralized infrastructure.
- Hosted on **AWS EC2 (Canada region)** — a common choice for disposable phishing hosting.