

- Problem Statement ID : PS 02
- Team Name : GTS
- TEAM ID : HK-184
- TEAM MEMBERS : Ishaan Parashar, Aryan Sharma,
Aman Verma, Mahin



Problem

Cyber-attacks against Indian critical infrastructure are increasing rapidly.

Hackers often:

- Plan attacks on forums, dark webs, Telegram groups, and other underground platforms.
- Leak stolen credentials (usernames, passwords, addresses, configs)
- Sell database dumps and internal access
- Coordinate ransomware and DDoS attacks

The major issue is:

*There is **no automated system** continuously monitoring these platforms to detect early warning signs.*

As a result:

- Threats are discovered only after damage is done
- Organizations react late
- National infrastructure remains vulnerable



SOLUTION

We propose “Threat Forewarn” that acts as an early-warning system for cyber threats.

Our model:

- Continuously scans selected suspicious forums at regular time intervals.
- Analyze posts using NLP to detect threat-related keywords.
- Monitor websites and assign it a “Threat Score” based on severity, credibility, etc.
- Sends an alert if the Threat Score exceeds the threshold (e.g., 40)
- Enables early warning and timely action by concerned authorities.
- Detects leaked credentials using pattern matching.



Flow of Solution

1. Keep an eye on public forums.

An automated system monitors cybersecurity forums continuously.

2. Gather and Clean Data

Posts that are relevant are retrieved and prepare.

3. Threat Analysis of NLP

AI models determine the threat type, target, and intention.

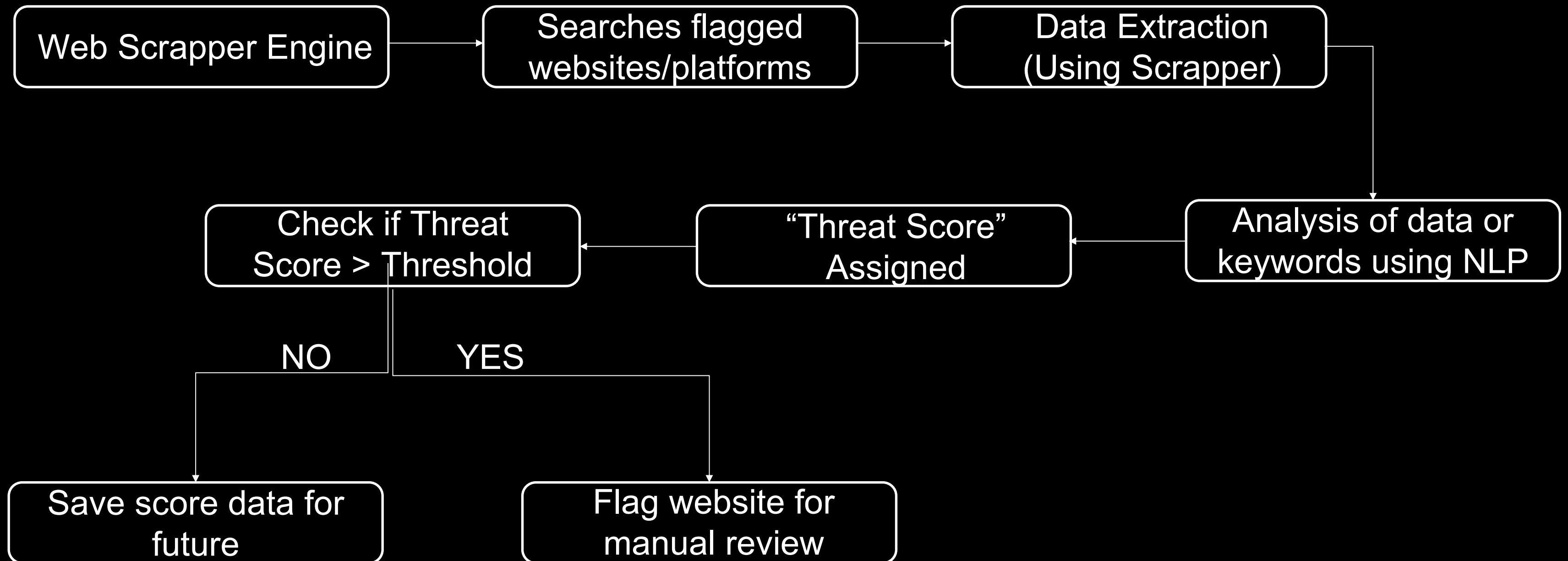
4. Threat Scoring

Every threat is ranked according to its level of risk and severity.

5. Dashboard & Alerts

Real-time insights are shown for those websites that has high Threat Score.

Flow of Solution



TECH STACK & APPROACH

Frontend: React.js + Tailwind CSS – For responsive, threat dashboard.

Backend: Node.js + Express – Scalable API architecture for handling alerts.

Logic : Web scrapper (Scrappy)+ (spaCy + Regex) – NLP engine for entity extraction and pattern matching. (Using Python)

Database: MongoDB – storage for threat detection and intelligence data

UNIQUENESS & INNOVATION FACTOR

Structured Threat Scoring Model:

Instead of flagging every suspicious post, our system intelligently prioritizes them. Each alert is scored based on severity, source credibility, target importance, recency, and evidence of leaked data.

Human-in-the-Loop Validation:

Since automation is not perfect, high-risk alerts are reviewed by analysts. They confirm whether the threat is real, false, or needs monitoring. This reduces false positives and continuously improves the system over time.

FeASiBility & ChAllenGeS

Feasibility:

- Built using proven open-source technologies for reliable scrapping, analysis and data management.
- Core stack includes Scrapy, spaCy, MongoDB, and Node.js for scalable and efficient system performance.

Challenges:

- Threat Credibility - Use a credibility scoring system based on user history, platform reputation, proof of leak, and cross-platform verification to filter out false threats.
- Missing High-Risk Alerts - Implement priority-based filtering to ensure critical threats are never hidden among low-risk alerts.

ReSeARCh & ReFeRenCe

- spaCy Documentation – Named Entity Recognition & Rule-based Matching.
- Scrapy Documentation.
- Verizon – Data Breach Investigations Report. (DBIR)
- IBM – X-Force Threat Intelligence Index.