## 🔍 **A Day in the CyberSecure System**

**Meet Joel.**  
Joel is a cybersecurity officer working with a national cybercrime task force. He logs into the **CyberSecure Law Enforcement Portal** using his credentials. The system checks his **username** and **hashed password** in the Users table and recognizes him as a user with the **Officer** role.

### 📝 **Reporting an Incident**

Joel receives a tip about a phishing scam targeting local businesses.  
He goes to the portal and **creates a new incident report**. This gets stored in the Incidents table:

* A new incident\_id is generated.
* The user\_id of Joel links him to this incident as the reporter.
* He enters the **title** as "Business Email Compromise" and writes a full **description**.
* He marks the **status** as "Open" and sets the **severity** to "High".
* The date\_reported gets auto-filled with the current timestamp.

### 📂 **Adding Digital Evidence**

Joel collects some logs and email headers from a victim and uploads them.  
Each file becomes a new entry in the Evidence table:

* A unique evidence\_id is assigned.
* It's linked to the incident\_id Joel just created.
* The system stores the **file name**, **file path**, and a **hash value** to ensure file integrity.
* Joel’s user\_id is recorded as the **uploader**.
* The **timestamp** records when the file was added.

The system ensures these files are **encrypted**, and logs every action — creating an **audit trail** for accountability.

### 👤 **Identifying a Suspect**

Through investigation, Joel and his team find an IP address and email that point to a possible suspect named **Ravi Verma**.

* They create a **Suspect Profile**, linked to the same incident\_id.
* The suspect\_id is unique, and attributes like **name**, **email**, and **IP address** are stored.
* The portal uses AI to assign Ravi a **risk score** based on his digital activity — say **0.87**, indicating high risk.
* Joel adds some **notes** about Ravi's known aliases and previous offenses.

This is stored in the Suspect Profiles table.

### 🤖 **Threat Intelligence Comes In**

Meanwhile, the portal’s backend AI analyzes the uploaded evidence.  
It flags the threat as **Phishing**, assigns a **confidence score** of 0.92, and logs this in the Threat Intelligence table:

* It’s linked back to the same incident\_id.
* The **threat type**, **threat level**, and **timestamp** of detection are recorded.

### ✉️ **Collaborating via Secure Messaging**

Joel needs input from an analyst named **Anita**.  
He sends her a secure, **end-to-end encrypted** message through the portal.

* The message is logged in the Messages table:
  + Sender: Joel’s user\_id
  + Receiver: Anita’s user\_id
  + Content is encrypted
  + Timestamp is saved

This allows seamless inter-agency communication — all searchable by incident or user later.

### 📄 **Generating a Report**

As the investigation progresses, Joel generates a **summary report**.  
He selects **“Threat Brief”** as the report type, and the portal auto-generates a PDF.

* A new entry is created in the Reports table:
  + Linked to the incident\_id
  + Includes Joel’s user\_id as the report creator
  + Saves the **file path** to the document
  + Logs the **timestamp**

### ✅ **Wrapping Up the Case**

After weeks of work, Ravi is apprehended, and the phishing operation is shut down.  
Joel marks the incident **status as "Resolved"**.

The portal now has:

* A complete **case history**,
* Linked **evidence**,
* Associated **threat intelligence**,
* A detailed **suspect profile**,
* All **messages** exchanged during the case,
* And the final **report**.

### 💡 The Outcome

Thanks to the schema:

* Every action is traceable and role-based.
* Files and messages are secure.
* AI aids human decision-making.
* Collaboration and documentation are seamless.

And Joel? He’s ready for the next case.

## ✅ What is confidence\_score?

In cybersecurity, a confidence score reflects how sure the system is that a reported item (file, IP, URL, etc.) is a threat. Typically ranges from 0.0 (least confident) to 1.0 (very confident).

## 🎯 Key Approaches to Compute Confidence Score

### 🔹 Option 1: Rule-Based Confidence (No AI Needed)

You can define rules based on properties of the incident or evidence.

#### Example Heuristics:

| **Feature** | **Weight** | **Description** |
| --- | --- | --- |
| Severity is "Critical" | +0.3 | High-severity incidents likely correlate with real threats |
| Multiple evidences uploaded | +0.2 | More proof increases confidence |
| Keywords in description (like "ransomware", "malware") | +0.2 | Match against known threat terms |
| File extension is suspicious (e.g., .exe, .js) | +0.15 | Dangerous file types |
| Reported by Admin | +0.15 | Admins may report verified cases |

#### Sample Python Code:

python

CopyEdit

def compute\_confidence\_score(incident):

score = 0.0

if incident.severity == 'Critical':

score += 0.3

elif incident.severity == 'High':

score += 0.2

elif incident.severity == 'Medium':

score += 0.1

if incident.evidences.count() >= 3:

score += 0.2

elif incident.evidences.count() >= 1:

score += 0.1

keywords = ['ransomware', 'malware', 'trojan', 'phishing']

if any(kw in incident.description.lower() for kw in keywords):

score += 0.2

for evidence in incident.evidences.all():

if evidence.file.name.endswith(('.exe', '.js', '.bat')):

score += 0.1

break

if incident.user.role == 'Admin':

score += 0.1

return min(score, 1.0)

## CyberSecure Law Enforcement Portal:

**📊 Database Schema**

**1. Users Table**

| **Field Name** | **Data Type** | **Description** |
| --- | --- | --- |
| user\_id | INTEGER (PK) | Unique ID for each user |
| username | VARCHAR | Username for login |
| password\_hash | VARCHAR | Hashed password |
| email | VARCHAR | User’s email |
| role | VARCHAR | Role: Officer, Analyst, Admin |
| created\_at | DATETIME | Timestamp of registration |
| last\_login | DATETIME | Last login timestamp |

**🛠 Tasks:**

* Implement user registration and login system with hashed password storage.
* Define role-based access (RBAC) for Officers, Analysts, Admins.
* Log user activity (last login, login attempts, etc.).

**2. Incidents Table**

| **Field Name** | **Data Type** | **Description** |
| --- | --- | --- |
| incident\_id | INTEGER (PK) | Unique incident/case ID |
| user\_id | INTEGER (FK) | Officer who reported the incident |
| title | VARCHAR | Title/summary of incident |
| description | TEXT | Full description of the cybercrime |
| status | VARCHAR | Status: Open, In Progress, Resolved |
| severity | VARCHAR | Severity: Low, Medium, High, Critical |
| date\_reported | DATETIME | When the incident was reported |

**🛠 Tasks:**

* Allow users to file new incident reports with file uploads.
* Enable tracking and updating of case status.
* Use severity to prioritize AI threat analysis.

**3. Evidence Table**

| **Field Name** | **Data Type** | **Description** |
| --- | --- | --- |
| evidence\_id | INTEGER (PK) | Unique evidence ID |
| incident\_id | INTEGER (FK) | Related incident |
| file\_name | VARCHAR | Name of uploaded file |
| file\_path | VARCHAR | Path to stored evidence |
| hash\_value | VARCHAR | Hash for integrity verification |
| uploaded\_by | INTEGER (FK) | User who uploaded evidence |
| timestamp | DATETIME | Upload time |

**🛠 Tasks:**

* Securely store digital files with encryption and hashing.
* Provide forensic tools to view, analyze, and verify files.
* Maintain audit trail for evidence access and modifications.

**4. Threat Intelligence Table**

| **Field Name** | **Data Type** | **Description** |
| --- | --- | --- |
| threat\_id | INTEGER (PK) | Unique threat ID |
| incident\_id | INTEGER (FK) | Source incident |
| threat\_type | VARCHAR | Type: Malware, Phishing, Ransomware, etc. |
| confidence\_score | FLOAT | AI confidence in detection (0–1) |
| threat\_level | VARCHAR | Level: Low, Medium, High |
| detected\_at | DATETIME | Timestamp of detection |

**🛠 Tasks:**

* Integrate AI models for detecting and classifying threats.
* Auto-populate threat data from incident analysis.
* Visualize in the threat intelligence dashboard.

**5. Suspect Profiles Table**

| **Field Name** | **Data Type** | **Description** |
| --- | --- | --- |
| suspect\_id | INTEGER (PK) | Unique ID |
| incident\_id | INTEGER (FK) | Linked incident |
| name | VARCHAR | Known name or alias |
| email | VARCHAR | Suspect email (if identified) |
| ip\_address | VARCHAR | IP traces |
| risk\_score | FLOAT | AI-assigned score based on activities |
| notes | TEXT | Analyst notes |

**🛠 Tasks:**

* Extract suspect behavior from patterns in cases.
* Use ML to assign a risk score.
* Store indicators like IP address, email, etc.

**6. Messages Table (For Inter-Agency Communication)**

| **Field Name** | **Data Type** | **Description** |
| --- | --- | --- |
| message\_id | INTEGER (PK) | Unique ID |
| sender\_id | INTEGER (FK) | Sender user |
| receiver\_id | INTEGER (FK) | Receiver user |
| content | TEXT | Encrypted message content |
| timestamp | DATETIME | Sent time |

**🛠 Tasks:**

* Enable secure, end-to-end encrypted messaging.
* Store communication logs between agencies.
* Use filters for searching messages by user or incident.

**7. Reports Table**

| **Field Name** | **Data Type** | **Description** |
| --- | --- | --- |
| report\_id | INTEGER (PK) | Unique report ID |
| incident\_id | INTEGER (FK) | Related case |
| generated\_by | INTEGER (FK) | User who created the report |
| report\_type | VARCHAR | Summary, Final, Threat Brief |
| file\_path | VARCHAR | Report storage path |
| created\_at | DATETIME | Report generation time |

**🛠 Tasks:**

* Enable auto-generated case summaries and threat reports.
* Export reports in PDF/Excel formats.
* Link reports with incidents and responsible users.