Cybersecurity Psychology Framework (CPF) - Integrazione SOC

Contents

Questo repository contiene l'implementazione ufficiale del **Cybersecurity Psychology Framework (CPF)**, un metodo innovativo per quantificare il rischio incentrato sull'uomo nei Security Operations Center (SOC). Fornisce una suite di algoritmi per misurare le vulnerabilità psicologiche definite dalla tassonomia CPF utilizzando i dati di strumenti SOC standard.

> **Riferimento Accademico**: Questo lavoro è l'implementazione pratica della metodologia descritta nel preprint: **"The Cybersecurity Psychology Framework (CPF): A Method for Quantifying Human Risk and a Blueprint for LLM Integration"** *di Giuseppe Canale, CISSP* Link al Documento

Panoramica

Il fattore umano è la vulnerabilità più critica nella sicurezza informatica. Gli strumenti tradizionali si concentrano su indicatori tecnici, lasciando gli stati psicologici non misurati. Questo progetto

operazionalizza il CPF, traducendo le sue categorie teoriche in algoritmi specifici e misurabili che analizzano i dati da strumenti come Splunk, Elasticsearch, Qualys, Jira e Slack.

Il framework è organizzato nelle dieci categorie primarie del CPF. Ogni directory di categoria contiene implementazioni per le sue sottocategorie specifiche (ad es. `1.1-unquestioning-compliance.py`).

Struttura del Repository

```
cpf-soc-integration/

  implementation/
      1.x-authority/              # Vulnerabilità basate su Autorità
          1.1-unquestioning-compliance.py
          1.2-diffusion-responsibility.py
          ... (altre sottocategorie)

      2.x-temporal/               # Vulnerabilità Temporali
      3.x-social/                 # Vulnerabilità per Influenza Sociale
      4.x-affective/              # Vulnerabilità Affettive
      5.x-cognitive/              # Vulnerabilità da Sovraccarico Cognitivo
      6.x-group/                  # Vulnerabilità da Dinamica di Gruppo
      7.x-stress/                 # Vulnerabilità da Risposta allo Stress
      8.x-unconscious/            # Vulnerabilità da Processi Inconsci
      9.x-ai/                     # Vulnerabilità di Bias Specifico per IA
      10.x-convergent/            # Stati Convergenti Critici

  docs/
      CPF-Taxonomy-Complete.pdf   # La tassonomia CPF completa
      operational-sheets/         # Schede dettagliate per ogni sottocategoria

  config/
      example.config.yaml         # Configurazione di esempio per chiavi API e endpoint

  requirements.txt
  README.md
```

Guida Introduttiva

Prerequisiti

- **Python 3.8+**
- Accesso alle fonti di dati SOC (ad es. Splunk, Elasticsearch, Jira, Slack, API Qualys)
- Chiavi API/token per i servizi sopra menzionati

Installazione

1. **Clonare il repository:**

   ```
   git clone https://github.com/your-username/cpf-soc-integration.git
   ```

```
cd cpf-soc-integration
```

2. **Creare un ambiente virtuale (consigliato):**

```
python -m venv venv
source venv/bin/activate  # Su Windows: .\venv\Scripts\activate
```

3. **Installare le dipendenze richieste:**

```
pip install -r requirements.txt
```

Configurazione

1. Copiare il file di configurazione di esempio e adattarlo al vostro ambiente:

```
cp config/example.config.yaml config/config.yaml
```

2. Modificare `config/config.yaml` con i vostri dettagli specifici:

```
splunk:
  host: "your-splunk-host.com"
  port: 8089
  username: "your_username"
  password: "your_password"

jira:
  server: "https://your-company.atlassian.net"
  email: "your.email@company.com"
  api_token: "your_jira_api_token"

# ... configurare altre fonti di dati
```

Utilizzo

L'uso principale di questo repository è il calcolo di metriche per sottocategorie CPF specifiche. Ogni script è progettato per essere eseguito in modo indipendente o integrato in una pipeline di analitiche più ampia.

Esecuzione di una Singola Metrica

Per calcolare una metrica specifica, eseguire lo script Python corrispondente. La maggior parte degli script accetterà parametri o trarrà dalla configurazione centrale.

Esempio: Calcolo della Fatica di Conformità (5.1 Alert Fatigue Desensitization)

```
# Navigare alla directory di sovraccarico cognitivo
cd implementation/5.x-cognitive

# Eseguire l'algoritmo per un analista o team specifico
python 5.1-alert-fatigue-desensitization.py --analyst-id "analyst_john.doe" --start-date "2023
```

Output Previsto:

```
Calculating Compliance Fatigue for analyst_john.doe (2023-11-01 to 2023-11-30)...
MTTA: 18.7 hours
Ignore Rate: 22.5%
[STATUS] YELLOW: Moderate fatigue detected. Recommend task rotation.
```

Integrazione con un LLM (Avanzato)

Come illustrato nel documento che accompagna questo progetto, questi algoritmi sono progettati per alimentare una pipeline Retrieval-Augmented Generation (RAG) per un LLM leggero. L'output di questi script (le metriche e i frammenti di dati rilevanti) può essere indicizzato in un database vettoriale (ad es. ChromaDB, FAISS) per fornire contesto a un LLM per analisi sofisticate del rischio psicologico.

Contributi

I contributi sono benvenuti! Questo è un progetto di ricerca-in-pratica su larga scala.

1. **Fare un fork del repository.**
2. **Creare un branch per la feature:** `git checkout -b feature/amazing-algorithm`
3. **Implementare il vostro algoritmo** per una sottocategoria CPF nella directory appropriata.
4. **Committare i vostri cambiamenti:** `git commit -m 'Add amazing algorithm for subcategory X.Y'`
5. **Eseguire il push al branch:** `git push origin feature/amazing-algorithm`
6. **Aprire una Pull Request.**

Assicurarsi che il vostro codice sia ben commentato e includa una docstring che spieghi la metrica, la formula e le fonti di dati.

Licenza

Questo progetto è concesso in licenza secondo la Licenza MIT - vedere il file LICENSE.md per i dettagli. Questa licenza consente l'uso accademico e commerciale con attribuzione.

Disclaimer ed Uso Etico

Questo strumento è progettato per misurare la salute organizzativa e il rischio psicologico a livello di team, NON per il monitoraggio di individui.

- **Privacy:** Anonimizzare sempre i dati dove possibile. Seguire il principio della minimizzazione dei dati.
- **Etica:** L'implementazione di questo framework deve essere trasparente ai dipendenti e conformarsi a tutti i regolamenti locali sulla protezione dei dati (ad es. GDPR, CCPA). Deve essere utilizzato per supportare e aumentare i team di sicurezza, non per punirli.
- **Accuratezza:** Queste metriche sono proxy per costrutti psicologici. Devono essere utilizzate come indicatori anticipati e integrate con audit e interviste condotte da esseri umani.

Cita Questo Lavoro

Se utilizzi questo framework nella tua ricerca o lavoro, please cita il documento che lo accompagna:

```
@article{canale2025cpf,
  title={The Cybersecurity Psychology Framework (CPF): A Method for Quantifying Human Risk and
  author={Canale, Giuseppe},
  journal={Preprint},
  year={2025},
  url={https://github.com/xbeat/CPF}
}
```

Contatti

Giuseppe Canale, CISSP - g.canale@cpf3.org | ORCID

Link del Progetto: https://github.com/xbeat/CPF