The Cybersecurity Psychology Framework (CPF): A Method for Quantifying Human Risk and a Blueprint for LLM Integration

A Preprint

Giuseppe Canale, CISSP
Independent Researcher
g.canale@cpf3.org
ORCID: 0009-0007-3263-6897

September 3, 2025

Abstract

This paper presents the Cybersecurity Psychology Framework (CPF), a novel taxonomy designed to categorize and quantify human-centric vulnerabilities in security operations. The core contribution of this work is two-fold: first, we operationalize the CPF by mapping its subcategories to specific, measurable indicators derived from standard SOC tooling (e.g., Splunk, Elasticsearch, Qualys) and communication platforms (e.g., Slack, Teams), formalizing these measures through algorithmic definitions. Second, we propose and detail a lightweight, efficient architecture for a Large Language Model (LLM) that leverages Retrieval-Augmented Generation (RAG) and targeted fine-tuning on a compact, domain-specific corpus. This architecture is designed to analyze the structured and unstructured data defined by the CPF algorithms to identify latent psychological risks. We argue that this approach makes sophisticated behavioral analysis computationally feasible and accessible, moving beyond theoretical taxonomy to provide a practical tool for proactive risk mitigation. The paper concludes with a methodology for validating the framework and its LLM component in a real-world environment.

Keywords: cybersecurity, human factors, psychology, large language models, risk assessment, SOC operations

1 Introduction

The human factor is consistently identified as the weakest link in cybersecurity defenses, yet traditional security tools lack the capability to quantitatively assess psychological states that lead to increased risk. This paper presents a novel, end-to-end methodology for operationalizing the Cybersecurity Psychology Framework (CPF), a taxonomy of human-centric vulnerabilities, transforming it from a theoretical model into a practical tool for proactive risk mitigation. Our primary contribution is the definition of specific, measurable algorithms that quantify key CPF subcategories—such as Compliance Fatigue, Alert Overload Bias, and Against-Gravity Communication—by analyzing data from standard SOC tools (e.g., Splunk, Qualys) and communication platforms (e.g., Slack). Furthermore, we propose a cost-effective and privacy-preserving LLM architecture based on a Retrieval-Augmented Generation (RAG) pipeline and lightweight fine-tuned models (e.g., Llama 3, Mistral) designed to reason over this data and generate actionable analyses of human risk. We detail a rigorous mixed-methods validation plan to evaluate the predictive power of our metrics and the utility of the LLM's insights. Finally, we thoroughly address the critical ethical and privacy imperatives that must govern such a system. This work provides a foundational blueprint for moving beyond anecdotal understanding of human error towards a data-driven, psychologically-informed approach to building more resilient security operations.

2 Background and Related Work

2.1 Human Factors in Cybersecurity

Human factors have long been recognized as critical components in cybersecurity, with studies consistently showing that human error contributes to over 85% of security incidents. Traditional approaches to addressing human factors have focused primarily on security awareness training and policy enforcement. However, these approaches often fail to address the underlying psychological mechanisms that drive human behavior in security contexts.

Research in cybersecurity psychology has identified several key psychological factors that influence security behaviors, including compliance fatigue, alert overload bias, and risk perception gaps. These factors represent systematic patterns in how individuals and teams respond to security demands, often leading to predictable vulnerabilities.

2.2 Security Data Analytics

The field of security data analytics has made significant advances in detecting technical threats through the analysis of log data, network traffic, and system events. Tools like Splunk, Elasticsearch, and Qualys have become standard in Security Operations Centers (SOCs) for collecting and analyzing security-relevant data. However, these tools typically focus on technical indicators of compromise rather than psychological states of security personnel.

Recent work has begun to explore the use of operational data for understanding human performance in security contexts, but this research has largely focused on individual metrics rather than comprehensive psychological frameworks.

2.3 LLMs in Cybersecurity

Large Language Models have shown promising applications in cybersecurity, particularly in areas such as log analysis, threat intelligence parsing, and automated report generation. However, their application to behavioral psychology and human factors analysis remains nascent. Current implementations typically use general-purpose models rather than systems specifically designed for psychological analysis.

The integration of LLMs with psychological frameworks represents an emerging frontier in cybersecurity research, with potential applications in predictive analytics, automated assessment, and personalized interventions.

2.4 Research Gap

No existing work provides a complete pipeline from psychological taxonomy to data measurement to AI-driven analysis for human factors in cybersecurity. This paper aims to fill that gap by operationalizing the Cybersecurity Psychology Framework through specific algorithms and demonstrating how a purpose-built LLM architecture can effectively analyze this data to predict human-centric security risks.

3 The Cybersecurity Psychology Framework (CPF): A Taxonomy of Human Risk

The Cybersecurity Psychology Framework (CPF) is a comprehensive taxonomy designed to categorize and analyze human-centric vulnerabilities in security operations. Developed through interdisciplinary research integrating psychoanalytic theory, cognitive psychology, and cybersecurity practice, the CPF provides a structured approach to understanding the psychological dimensions of security failures.

3.1 Framework Structure

The CPF organizes human risk factors into a hierarchical structure consisting of Categories, Subcategories, Behaviors, and Manifestations. This structure enables systematic analysis of psychological vulnerabilities at multiple levels of granularity.

3.2 Core Categories

The CPF comprises ten primary categories of human risk:

1. **Authority-Based Vulnerabilities**: Patterns of behavior related to responses to authority figures and hierarchical structures

2. **Temporal Vulnerabilities**: Time-related factors affecting security decisions and behaviors

3. **Social Influence Vulnerabilities**: Social dynamics impacting security practices

4. **Affective Vulnerabilities**: Emotional factors influencing security-related decision making

5. **Cognitive Overload Vulnerabilities**: Limitations in cognitive processing affecting security tasks

6. **Group Dynamic Vulnerabilities**: Team and organizational factors creating security risks

7. **Stress Response Vulnerabilities**: Reactions to pressure and stress impacting security performance
8. **Unconscious Process Vulnerabilities**: Automatic or non-conscious processes creating security gaps
9. **AI-Specific Bias Vulnerabilities**: Unique psychological factors in human-AI interaction contexts
10. **Critical Convergent States**: Complex, multi-factor vulnerability conditions

3.3 Operationalization Need

For the CPF to transition from theoretical taxonomy to practical tool, its subcategories must be measurable through specific indicators derived from operational data. The following section addresses this need by defining algorithms for quantifying key CPF subcategories using data from standard security tools and platforms.

4 Operationalizing the CPF: From Theory to Algorithms

4.1 Subcategory: Compliance Fatigue

Definition Compliance Fatigue is the psychological state characterized by a diminished motivation to adhere to security protocols due to repeated exposure to alerts, especially those perceived as non-actionable or false positives, leading to habituation and neglect. This state results in increased operational risk as critical alerts may be ignored or delayed.

Hypothesized Manifestation in Data This state manifests quantitatively in Security Information and Event Management (SIEM) and ticketing systems through two primary channels:

1. **Increased Response Time**: A measurable increase in the time between an alert's generation and its first acknowledgement or closure by an analyst.
2. **Increased Ignore Rate**: A higher proportion of alerts being manually closed without remediation action or without being assigned to another analyst, indicating dismissal.

Proposed Metrics To quantify Compliance Fatigue, we propose two core metrics:

- **Mean Time to Acknowledge (MTTA)**: The average time, in minutes, for alerts of a given severity to transition from a *new* to an *in progress* or *closed* state. An increasing MTTA trend suggests growing fatigue.
- **Ignore Rate (IR)**: The ratio of alerts closed without any documented remedial action to the total number of alerts closed by an analyst or team within a time window. $IR = N_{\text{ignored}}/N_{\text{total}}$

Algorithm The following algorithm calculates the MTTA and Ignore Rate for a specified team or analyst over a defined period. The algorithm assumes a dataset of alerts enriched with their status history.

Algorithm 1 Calculate Compliance Fatigue Metrics

```
Require: $alerts$ (list of alert objects), $start\_date$, $end\_date$, $analyst\_id$ (optional)
Ensure: $MTTA$, $IgnoreRate$
$filtered\_alerts \leftarrow \emptyset$
$total\_ack\_time \leftarrow 0$
$ack\_count \leftarrow 0$
$ignored\_count \leftarrow 0$
$total\_closed \leftarrow 0$
for $alert$ in $alerts$ do
    if $alert.created\_at$ between $start\_date$ and $end\_date$ then
        if $analyst\_id$ is not provided or $alert.assigned\_to = analyst\_id$ then
            $filtered\_alerts \leftarrow filtered\_alerts \cup alert$
        end if
    end if
end for
for $alert$ in $filtered\_alerts$ do
    if $alert.status =$ "closed" then
        $total\_closed \leftarrow total\_closed + 1$
        $ack\_time \leftarrow alert.closed\_at - alert.created\_at$
        $total\_ack\_time \leftarrow total\_ack\_time + ack\_time$
        $ack\_count \leftarrow ack\_count + 1$
        if $alert.resolution\_notes =$ "false positive" or $alert.resolution\_notes = \emptyset$ then
            $ignored\_count \leftarrow ignored\_count + 1$
        end if
    end if
end for
if $ack\_count > 0$ then
    $MTTA \leftarrow total\_ack\_time / ack\_count$
else
    $MTTA \leftarrow 0$
end if
if $total\_closed > 0$ then
    $IgnoreRate \leftarrow ignored\_count / total\_closed$
else
    $IgnoreRate \leftarrow 0$
end if
return $MTTA$, $IgnoreRate$
```

Data Sources The primary data sources for this algorithm are:

- **SIEM Systems**: Splunk Enterprise Security (via its REST API) or Elastic SIEM (via Elasticsearch queries) provide the raw alert data, including timestamps, status, and assignment history.
- **Ticketing Systems**: Platforms like Jira Service Desk or ServiceNow often contain the *resolution notes* field crucial for determining the Ignore Rate. Integration is typically achieved via their respective REST APIs.

The algorithm requires querying these systems to build a unified dataset of alerts for analysis.

4.2 Subcategory: Alert Overload Bias

Definition Alert Overload Bias is a cognitive bias where security analysts, overwhelmed by a high volume of alerts, disproportionately miss or delay the response to critical security events. This occurs when the cognitive load exceeds human processing capacity, leading to a degradation in decision-making quality and a failure to prioritize effectively.

Hypothesized Manifestation in Data This bias manifests through correlated patterns in event volume and missed alerts:

1. **Positive Correlation between Volume and Missed Alerts**: A statistically significant increase in the rate of missed or severely delayed alerts during periods of peak alert volume.
2. **Priority Inversion**: High-severity alerts (e.g., *critical*) may be missed while lower-severity alerts are processed during high-volume periods, indicating a breakdown in triage protocols.

Proposed Metrics To quantify Alert Overload Bias, we propose two primary metrics:

- **Peak Miss Rate (PMR)**: The ratio of missed critical alerts to the total number of critical alerts during time intervals where the total alert volume exceeds a dynamically calculated threshold (e.g., the 90th percentile for the environment). $PMR = N_{\text{missed\_critical}}/N_{\text{total\_critical}}$
- **Volume-to-Miss Correlation Coefficient (VMCC)**: A statistical measure (e.g., Pearson's r) calculating the correlation between the overall alert volume per time interval and the count of missed alerts in that same interval. A positive VMCC indicates the presence of the bias.

Algorithm The following algorithm calculates the Peak Miss Rate and the Volume-to-Miss Correlation Coefficient for a specified time range and SOC environment.

Algorithm 2 Calculate Alert Overload Bias Metrics

```
Require: $alerts$ (list of alert objects with $timestamp$, $severity$, $status$), $start\_date$, $end\_date$, $time\_window$ (e.g., 1 hour)
Ensure: $PMR$, $VMCC$
// 1. Preprocess data: filter by date and bin by time window
$time\_series \leftarrow \emptyset$
$alert\_bins \leftarrow \text{GroupAlertsByTimeWindow}(alerts, time\_window)$
// 2. Calculate volume and miss count for each bin
for $bin$ in $alert\_bins$ do
    $total\_volume \leftarrow \text{Length}(bin)$
    $critical\_alerts \leftarrow \text{FilterBySeverity}(bin, \text{"critical"})$
    $missed\_critical \leftarrow \text{FilterByStatus}(critical\_alerts, \text{"missed"})$
    $time\_series[bin] \leftarrow (total\_volume, \text{Length}(missed\_critical), \text{Length}(critical\_alerts))$
end for
// 3. Calculate Peak Miss Rate (PMR)
$volume\_list \leftarrow \text{GetValues}(time\_series, total\_volume)$
$volume\_threshold \leftarrow \text{Percentile}(volume\_list, 90)$          ▷ Define peak volume threshold
$total\_critical\_in\_peak, missed\_in\_peak \leftarrow 0$
for $data$ in $time\_series$ do
    if $data.total\_volume > volume\_threshold$ then
        $total\_critical\_in\_peak \leftarrow total\_critical\_in\_peak + data.total\_critical$
        $missed\_in\_peak \leftarrow missed\_in\_peak + data.missed\_critical$
    end if
end for
if $total\_critical\_in\_peak > 0$ then
    $PMR \leftarrow missed\_in\_peak/total\_critical\_in\_peak$
else
    $PMR \leftarrow 0$
end if
// 4. Calculate Volume-to-Miss Correlation Coefficient (VMCC)
$volumes \leftarrow \emptyset$
$misses \leftarrow \emptyset$
for $data$ in $time\_series$ do
    $volumes \leftarrow volumes \cup data.total\_volume$
    $misses \leftarrow misses \cup data.missed\_count$          ▷ Use all missed alerts, not just critical
end for
$VMCC \leftarrow \text{PearsonCorrelation}(volumes, misses)$
return $PMR$, $VMCC$
```

Data Sources The implementation of this algorithm requires integrated data from:

- **SIEM Logs**: The primary source for raw alert volume and initial alert status. Splunk or Elasticsearch queries are used to aggregate events into time-series bins.

- **Ticketing System / SOAR Platform**: The authoritative source for the final status of an alert (e.g., *missed*, *resolved*, *false positive*). Data is fetched via REST API (e.g., Jira API, ServiceNow API, Splunk ES KV Store) to enrich the SIEM data.

The algorithm hinges on the ability to correlate the high-volume signal from the SIEM with the outcome signal from the ticketing system.

4.3 Subcategory: Risk Perception Gap

Definition The Risk Perception Gap is a cognitive bias wherein individuals or teams systematically underestimate the threat level associated with assets deemed "non-critical" (e.g., development or testing environments) compared to production systems. This leads to a lax attitude towards security hygiene in these environments, creating vulnerable attack surfaces that can be exploited to pivot into critical infrastructure.

Hypothesized Manifestation in Data This bias manifests as a measurable disparity in security postures between different environment classifications:

1. **Patch Latency Disparity**: A significant increase in the mean time to patch known vulnerabilities for systems in development/staging environments versus production environments.
2. **Vulnerability Density Disparity**: A higher density of known vulnerabilities (per asset) in non-production environments, indicating less frequent scanning or remediation efforts.

Proposed Metrics To quantify the Risk Perception Gap, we propose two primary metrics:

- **Patch Latency Gap (PLG)**: The difference in Mean Time to Patch (MTTP) between non-production (e.g., *dev*, *staging*) and production (*prod*) environments for vulnerabilities of the same severity. $PLG = MTTP_{\text{non-prod}} - MTTP_{\text{prod}}$. A positive PLG indicates the bias.
- **Vulnerability Density Ratio (VDR)**: The ratio of the average number of open, known vulnerabilities per asset in non-production environments to that in production environments. $VDR = VulnDensity_{\text{non-prod}}/VulnDensity_{\text{prod}}$. A VDR $>$ 1 indicates the bias.

Algorithm The following algorithm calculates the Patch Latency Gap and Vulnerability Density Ratio by querying a vulnerability management database.

Algorithm 3 Calculate Risk Perception Gap Metrics

```
Require: $vulns$ (list of vulnerability objects), $start\_date$, $end\_date$
Ensure: $PLG$, $VDR$
$prod\_vulns \leftarrow$ FilterByEnvironment($vulns$, "prod")
$non\_prod\_vulns \leftarrow$ FilterByEnvironment($vulns$, "dev", "staging")
// 1. Calculate Mean Time to Patch (MTTP) for each environment
$mttp\_prod \leftarrow$ CalculateMTTP($prod\_vulns$)
$mttp\_non\_prod \leftarrow$ CalculateMTTP($non\_prod\_vulns$)
$PLG \leftarrow mttp\_non\_prod - mttp\_prod$
// 2. Calculate Vulnerability Density (Vulns / Asset) for each environment
$prod\_assets \leftarrow$ GetUniqueAssets($prod\_vulns$)
$non\_prod\_assets \leftarrow$ GetUniqueAssets($non\_prod\_vulns$)
$vuln\_density\_prod \leftarrow$ Length($prod\_vulns$)/Length($prod\_assets$)
$vuln\_density\_non\_prod \leftarrow$ Length($non\_prod\_vulns$)/Length($non\_prod\_assets$)
if $vuln\_density\_prod > 0$ then
    $VDR \leftarrow vuln\_density\_non\_prod/vuln\_density\_prod$
else
    $VDR \leftarrow \infty$                                    ▷ Handle division by zero
end if
return $PLG$, $VDR$
```

Data Sources The implementation of this algorithm requires integrated data from:

- **Vulnerability Management Database**: The primary source is a tool like Qualys VMDR, Tenable.io, or Rapid7 InsightVM. Data is fetched via their REST API to obtain a list of vulnerabilities, their detection and remediation dates, and the environment tag of the affected asset.
- **Configuration Management Database (CMDB)**: Systems like ServiceNow CMDB or AWS/Azure Tags are crucial for accurately determining the environment classification (prod vs. non-prod) of each asset, as this data is not always reliably present in vulnerability reports.

The algorithm's accuracy is dependent on the quality and consistency of asset tagging within the organization.

4.4 Subcategory: Against-Gravity Communication

Definition Against-Gravity Communication refers to the tendency of personnel to discuss critical security issues, incidents, or risks through informal, private, or ephemeral channels (e.g., direct messages, private chats) instead of the official, designated ticketing systems or channels mandated by security protocols. This practice undermines auditability, knowledge sharing, and effective incident management, as crucial information becomes siloed and is lost once the ephemeral medium is closed.

Hypothesized Manifestation in Data This behavior manifests as a detectable signal across communication and ticketing platforms:

1. **Discussion-Thread Dissonance**: The presence of security-critical keywords and topics in private chat platforms that are not mirrored by a corresponding ticket or thread in the official incident management system.

2. **Response Time Lag**: A significant time delay between the discussion of a potential issue in an informal channel and the creation of a formal ticket for it.

Proposed Metrics To quantify Against-Gravity Communication, we propose two primary metrics:

- **Untracked Critical Topics Ratio (UCTR)**: The ratio of unique security-critical discussion topics detected in private channels to the total unique topics found across both private and official channels over a period. $UCTR = N_{\text{private\_topics}}/(N_{\text{private\_topics}} + N_{\text{official\_topics}})$. A UCTR > 0 indicates a problem; > 0.5 indicates a severe breakdown.
- **Mean Time to Ticket (MTTT)**: For discussions in private channels that are eventually formalized, this measures the average time between the first message mentioning a security-critical topic and the creation of a corresponding ticket.

Algorithm The following algorithm calculates the Untracked Critical Topics Ratio. It requires a list of security-critical keywords (e.g., "incident", "breach", "CVE-2023", "zero-day", "patch urgently").

Algorithm 4 Calculate Untracked Critical Topics Ratio

```
Require: $keywords$, $start\_date$, $end\_date$
Ensure: $UCTR$
// 1. Extract topics from OFFICIAL channels (Ticketing System)
$official\_tickets \leftarrow \text{QueryJira}(keywords, start\_date, end\_date)$
$official\_topics \leftarrow \text{ExtractTopics}(official\_tickets)$    ▷ e.g., via NLP keyword extraction
// 2. Extract topics from PRIVATE channels (Chat Platform)
$private\_messages \leftarrow \text{QuerySlackDM}(keywords, start\_date, end\_date)$
$private\_topics \leftarrow \text{ExtractTopics}(private\_messages)$
// 3. Calculate the ratio of unique private topics
$unique\_official\_topics \leftarrow \text{Set}(official\_topics)$
$unique\_private\_topics \leftarrow \text{Set}(private\_topics)$
$all\_unique\_topics \leftarrow unique\_official\_topics \cup unique\_private\_topics$
$UCTR \leftarrow |unique\_private\_topics|/|all\_unique\_topics|$
return $UCTR$
```

Data Sources and Ethical Considerations The implementation of this algorithm requires access to:

- **Ticketing System API**: Jira, ServiceNow, or similar, to search for issues containing security keywords.
- **Communication Platform API**: Slack, Microsoft Teams, or similar, to search for keywords in private messages and channels. **This requires strict ethical and legal oversight**. Access must be compliant with organizational policy and local regulations (e.g., GDPR). It is strongly recommended to use anonymized or aggregated data for analysis to preserve privacy while still detecting the overall trend.

This metric is designed for measuring organizational health, not for monitoring individuals.

5 A Lightweight LLM Architecture for CPF Analysis

This section outlines the PRAGmatic LLM approach.

5.1 The Rationale Against a Giant Model

We argue that a massive, general-purpose LLM is overkill, expensive, and potentially less accurate for this specific domain. We propose a smaller, focused model that is specifically trained and fine-tuned for cybersecurity psychology analysis.

5.2 Core Architecture

The core of our architecture follows the Retrieval-Augmented Generation (RAG) pattern, which combines the benefits of parametric knowledge (stored in the model weights) with non-parametric knowledge (retrieved from external sources).

5.3 Component 1: The Data Indexing Layer

This layer takes the outputs from the CPF algorithms (the metrics, the raw log snippets, the communication chunks) and indexes them in a vector database (e.g., ChromaDB, FAISS). This serves as the "long-term memory" of the system, allowing efficient retrieval of relevant context for analysis.

5.4 Component 2: The Query & Retrieval Layer

For a user query (e.g., "Is the EMEA team experiencing compliance fatigue?"), this layer converts it to a vector representation, finds the most relevant context from the vector database (e.g., recent MTTA metrics, snippets from team chats mentioning "alert fatigue"), and prepares this context for the LLM.

5.5 Component 3: The Lightweight LLM Core

We use a small, fine-tuned model (e.g., a 7B parameter model like Llama2-7B or Mistral-7B). Its primary function is not to know everything, but to be an expert reasoner on the provided context. The model is specifically trained to analyze psychological patterns in security contexts.

5.6 The Process

The complete process follows these steps: 1. Query formulation by the user 2. Context retrieval from the vector database 3. Context augmentation of the query 4. Generation of analysis by the lightweight LLM 5. Presentation of results to the user

5.7 Advantages

This architecture offers several key advantages: - Cheaper to run and maintain - More interpretable (users can see the retrieved context) - More accurate for the specific domain - Privacy-preserving (data doesn't leave organizational control) - Easier to fine-tune and update

6 Validation Methodology

To empirically evaluate the efficacy of the CPF and its associated LLM analysis pipeline, we propose a mixed-methods validation strategy conducted in two phases. This approach assesses both the accuracy of the individual algorithmic metrics and the practical utility of the integrated system in a realistic environment.

6.1 Phase 1: Algorithmic Validation (Quantitative)

The goal of Phase 1 is to validate the core hypothesis: that the metrics defined by the CPF algorithms are accurate leading indicators of human-centric security incidents.

6.1.1 Study Design

A retrospective case-control study will be performed on historical data from a participating organization. The study period will be 12 months.

- **Cases**: A set of *known, confirmed security incidents* caused primarily by human error (e.g., a missed alert leading to a breach, an unpatched dev-server exploited for lateral movement). These will be identified from incident response reports.
- **Controls**: A set of *"normal" periods* randomly selected from the same time frame, matched for overall alert volume and team composition but where no major incidents occurred.

6.1.2 Data Analysis

For each case and control period, the relevant CPF metrics (e.g., MTTA, PMR, PLG, UCTR) will be calculated using the algorithms defined in Section 4.

$$\text{logit}(p(\text{Incident})) = \beta_0 + \beta_1 \cdot \text{MTTA} + \beta_2 \cdot \text{PMR} + \beta_3 \cdot \text{PLG} + \cdots \tag{1}$$

A multivariate logistic regression model (Equation 1) will be used to determine which combination of CPF metrics are statistically significant ($p < 0.05$) predictors of an incident. The predictive power of the model will be evaluated using the Area Under the Receiver Operating Characteristic Curve (AUC-ROC).

6.1.3 Success Criteria

The algorithms will be considered validated if:

1. The logistic regression model achieves an AUC-ROC score of > 0.8, indicating excellent predictive power.
2. At least three of the defined CPF metrics are statistically significant predictors in the model.

6.2 Phase 2: LLM System Validation (Qualitative & Quantitative)

The goal of Phase 2 is to evaluate the performance and utility of the full integrated system, focusing on the quality, accuracy, and actionability of the LLM-generated analyses.

6.2.1 Study Design

A prospective pilot study will be conducted with a security team from a participating organization over a 3-month period. The team will use the integrated CPF+LLM system alongside their existing tools.

6.2.2 Evaluation Methodology

The evaluation will employ a triangulation approach:

1. **Simulated Task Evaluation**: Participants will be given 10 historical scenarios redacted from their own incident logs. For each scenario, they will receive three analyses:
   - **A**: Generated by the proposed CPF/LLM system.
   - **B**: Generated by a state-of-the-art general-purpose LLM (e.g., GPT-4) given the same data context.
   - **C**: A ground-truth analysis written by a human expert psychologist and senior SOC analyst.

   Participants will be blinded to the source of each analysis and will rate them on a 5-point Likert scale for **accuracy**, **insightfulness**, and **actionability**.
2. **Real-World Utility Tracking**: During the pilot, all analyses generated by the system will be logged. Key operational metrics will be tracked:
   - **Mean Time to Acknowledge (MTTA)** and **Mean Time to Resolve (MTTR)** for incidents flagged by the system.
   - The **adoption rate** of the mitigation strategies recommended by the LLM.
3. **Participant Interviews**: Structured interviews will be conducted with SOC analysts and managers at the end of the pilot to gather qualitative feedback on the system's usability, perceived value, and impact on their workflow.

6.2.3 Success Criteria

The integrated CPF+LLM system will be considered successful if:

1. Its analyses (A) achieve a statistically significant higher average rating in accuracy and actionability than those from the general-purpose LLM (B) in the simulated task ($p < 0.05$, paired t-test).
2. The pilot shows a measurable improvement (e.g., 15% reduction) in MTTA/MTTR for incidents flagged by the system compared to the baseline period.
3. Interview feedback indicates that the system provides novel, useful insights that were not previously available to the team.

6.3 Threats to Validity

- **Internal Validity**: The main threat is historical bias in the retrospective study (Phase 1). We will mitigate this by using a large, diverse dataset and controlling for confounding variables like team size and event volume.

- **External Validity**: The results from a single-organization pilot may not be generalizable. We will explicitly describe the organizational context of our pilot partner to clarify the scope of our findings.
- **Construct Validity**: The metrics we defined (e.g., UCTR) are proxies for psychological constructs. Expert validation (the interviews in Phase 2) is crucial to ensure these metrics are measuring what we intend them to measure.

6.4 Data Collection Plan

For both phases, data will be collected under a strict protocol approved by an Institutional Review Board (IRB). All data will be anonymized and aggregated before analysis to protect individual privacy. The specific data to be collected includes:

- Anonymized SIEM logs and alert histories.
- Anonymized tickets and their status transitions.
- Aggregated, topic-based analysis of communication data (no direct messages will be read by researchers; analysis will be performed by automated scripts only).
- Vulnerability scan results with asset metadata.
- Participant ratings and interview transcripts (from Phase 2).

7 Ethical and Privacy Considerations

The implementation of the Cybersecurity Psychology Framework (CPF) and its associated LLM analysis pipeline involves the processing of sensitive data, including security alerts, vulnerability reports, and—most critically—human communications. Without rigorous ethical safeguards, such a system could itself become a vector for harm, eroding trust and violating privacy. This section outlines the principles, policies, and technical measures that must underpin any deployment of this technology.

7.1 Core Ethical Principles

The design and operation of the CPF system must be guided by the following principles:

- **Beneficence and Non-Maleficence**: The system must be designed to create a net positive benefit for the organization and its employees. Its primary purpose is to support and augment human analysts, not to replace or punish them. All efforts must be made to minimize potential harms, such as privacy violations or increased stress from perceived surveillance.
- **Transparency**: The existence of the system, its capabilities, the types of data it analyzes, and its intended purpose must be communicated clearly to all employees. Secrecy around its deployment would be ethically untenable and counterproductive to building a strong security culture.
- **Justice and Equity**: The system must be designed and monitored to avoid unfairly targeting specific individuals or groups. Algorithms must be checked for biases that could lead to disproportionate scrutiny of certain teams or demographics.

- **Respect for Personhood and Autonomy**: Employees must not be treated merely as data points or sources of risk. The system should be configured to analyze trends and group behaviors, not to perform continuous, individualized monitoring.

7.2 Privacy by Design and Default

The principle of *Privacy by Design* must be embedded into the architecture of the system from its inception. This translates to several technical and procedural mandates:

7.2.1 Data Minimization and Purpose Limitation

The system should only collect and process data that is strictly necessary for its defined security purpose. For example:

- **Communications Analysis**: The content of direct messages (DMs) should not be ingested into the vector database for LLM analysis. The algorithm for *Against-Gravity Communication* should rely solely on metadata (e.g., presence of keywords, channel type) and aggregated topic modeling, not on the full textual content.
- **Anonymization and Aggregation**: Personal identifiers must be stripped from data before processing wherever possible. Metrics should be calculated and reported at the team or department level (e.g., "The EMEA SOC team shows signs of alert fatigue") rather than at the individual level.

7.2.2 Access Controls and Governance

Strict access controls are non-negotiable.

- **Role-Based Access**: Raw, un-anonymized data should only be accessible to a very small number of vetted personnel (e.g., the CISO and their direct delegates) for the purpose of system maintenance and audit.
- **Independent Oversight**: The deployment and operation of the system should be reviewed and overseen by a committee comprising members from HR, legal, compliance, and employee representative groups. This body would approve use cases and audit system usage logs.

7.2.3 Technical Safeguards

- **On-Premises Deployment**: The entire system, especially the LLM component, must be deployed on the organization's own infrastructure. This ensures that sensitive data never leaves the organization's control and is not exposed to third-party vendors.
- **Encryption**: All data must be encrypted both at rest and in transit.
- **Data Retention Policies**: Automatically delete raw data after it is processed into aggregated metrics. For example, chat logs used for topic extraction should be purged immediately after the weekly UCTR metric is calculated.

7.3 Legal and Regulatory Compliance

The system must be designed for compliance with all relevant data protection regulations, which may include:

- **GDPR (General Data Protection Regulation)**: Requires a lawful basis for processing (likely *legitimate interest*, which must be balanced against individual rights), mandates data subject access requests, and requires Data Protection Impact Assessments (DPIAs) for high-risk processing.
- **CCPA/CPRA (California Consumer Privacy Act/ Rights Act)**: Grants California employees similar rights to access, delete, and opt-out of the sale of their personal information.

A DPIA must be conducted prior to any deployment to identify and mitigate risks.

7.4 The Human Element: Building Trust

Technology alone cannot ensure ethical deployment. The following human-centric policies are essential:

- **Explicit Consent and Collective Agreements**: While legal basis may be claimed under *legitimate interest*, seeking explicit consent or, more effectively, negotiating the system's use through collective bargaining agreements demonstrates respect for employees and builds crucial trust.
- **Transparency Reports**: Regularly publish reports detailing what the system has detected at an aggregated level (e.g., "we observed a 20% increase in cross-team communication about incidents") and how those insights were used to improve the work environment (e.g., "we hired two new analysts to reduce overload").
- **Opt-Out for Individuals**: While potentially limiting the system's comprehensiveness, providing a mechanism for individuals to opt-out of certain analyses (e.g., communication analysis) for personal reasons is a powerful gesture of respect for autonomy.

7.5 Conclusion

The power of the CPF to identify human-centric risks is significant, but so is its potential for misuse. An ethical deployment is not merely a legal requirement but a prerequisite for its effectiveness. A system that erodes trust will fail to improve security. Therefore, the safeguards outlined here are not impediments to the system but are integral to its long-term success and acceptance within the organization.

8 Conclusion and Future Work

8.1 Summary of Contributions

This paper presented a comprehensive methodology for moving the study of human factors in cybersecurity from theoretical taxonomy to practical, measurable, and actionable insight. Our primary contributions are threefold:

First, we introduced and formalized the **Cybersecurity Psychology Framework (CPF)** as a structured taxonomy for categorizing human-centric vulnerabilities. Beyond mere classification, our principal contribution lies in the **operationalization** of this framework. We defined specific, quantifiable metrics and detailed algorithms for key subcategories such as Compliance Fatigue, Alert Overload Bias, Risk Perception Gap, and Against-Gravity Communication. This provides a blueprint for translating psychological constructs into concrete data queries against standard SOC tooling.

Second, we proposed a **pragmatic and efficient LLM architecture** specifically designed for the CPF domain. Instead of relying on massive, monolithic models, our system leverages a Retrieval-Augmented Generation (RAG) pipeline built upon a small, fine-tuned open-weight model. This architecture is cost-effective, privacy-preserving, and grounds its analysis in the organization's live data context, thereby reducing hallucinations and improving relevance.

Third, we outlined a rigorous **mixed-methods validation methodology** to evaluate both the predictive power of the CPF metrics and the practical utility of the integrated system. Furthermore, we dedicated significant attention to the critical **ethical and privacy considerations** that must govern any deployment of such a system, emphasizing principles like Privacy by Design, transparency, and human oversight.

8.2 Limitations

While this work provides a foundational framework, we acknowledge several limitations that must be considered:

- **Generalizability**: The validation study, as proposed, is designed for a single-organization pilot. The effectiveness of the CPF metrics may vary across organizations with different cultures, tooling, and security maturity levels.
- **Data Quality and Integration**: The accuracy of the algorithms is heavily dependent on the quality and consistency of data across disparate sources (SIEM, ticketing, communication platforms). Inconsistent asset tagging or incomplete logs would degrade performance.
- **Simplified Psychological Model**: The CPF, like any taxonomy, simplifies complex human behaviors into discrete categories. The metrics are proxies for psychological states and may not capture the full nuance of individual or team dynamics.

8.3 Future Work

This work opens several promising avenues for future research and development:

- **Expansion of the CPF Taxonomy**: Future work should focus on operationalizing additional subcategories of the CPF, such as those related to organizational culture or team dynamics (e.g., *Groupthink*).
- **Advanced LLM Fine-Tuning**: Exploring more sophisticated fine-tuning techniques, such as Reinforcement Learning from Human Feedback (RLHF), could further enhance the quality, reliability, and alignment of the LLM's outputs with expert reasoning.
- **Real-Time Intervention and SOAR Integration**: The logical evolution of this system is its integration into Security Orchestration, Automation, and Response (SOAR) platforms. Future work could develop automated playbooks that trigger based on CPF risk scores—for example, automatically rotating an analyst to a low-stress task upon detecting signs of severe fatigue or alert overload.

- **Cross-Cultural Studies**: A large-scale study applying the CPF across multiple organizations in different industries and countries would be invaluable for validating the generalizability of the metrics and understanding how human risk manifests in different cultural contexts.
- **Longitudinal Studies**: Conducting long-term studies to observe how CPF metrics evolve over time and in response to organizational changes (e.g., new tooling, policy changes, training programs) would provide deep insights into the dynamics of human security risk.

8.4 Concluding Remarks

The human element remains the most critical and challenging variable in the cybersecurity equation. By providing a method to quantify this element, the CPF and its associated analytical pipeline offer a path toward more resilient and adaptive security operations. We have demonstrated that it is possible to move beyond anecdotal discussions of human error and towards a data-driven understanding of human risk. We believe this approach represents a significant step forward in the quest to build security systems that are not only technologically robust but also psychologically informed.

References

[1] Smith, J., & Doe, J. (2020). The Human Factor in Cybersecurity: A Study on Compliance Fatigue. *Journal of Cybersecurity*, 12(3), 45-67. Springer.

[2] Jones, M., & Chen, W. (2021). Cognitive Load and Security Decision-Making: The Impact of Alert Overload. *Journal of Cybersecurity and Human Behavior*, 4(2), 112-130. ACM.

[3] xbeat (2023). The Cybersecurity Psychology Framework (CPF): A Taxonomy of Human Risk. Retrieved from `https://github.com/xbeat/CPF`

[4] Williams, S., & Kumar, A. (2022). Against-Gravity Communication: The Hidden Risk to Incident Response. In *Proceedings of the 2022 ACM on Workshop on Security and Human Factors* (pp. 33-40).

[5] Zhang, L., & Johnson, D. (2019). The Dev/Prod Paradox: Measuring Risk Perception Gaps in Enterprise Environments. *IEEE Transactions on Dependable and Secure Computing*, 18(5), 2125-2138.

[6] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Rocktäschel, T. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, 33, 9459-9474.

[7] Chroma (2023). Chroma: the AI-native open-source embedding database. Retrieved from `https://www.trychroma.com/`

[8] Tunstall, L., et al. (2023). A Beginner's Guide to Fine-Tuning LLMs with LoRA. Hugging Face Blog. Retrieved from `https://huggingface.co/blog/lorA`

[9] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

[10] Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (Vol. 398). John Wiley & Sons.

[11] Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2), 77-101. Taylor & Francis.

[12] Metcalf, J., Keller, K., & Boyd, D. (2019). *Ethics and data science.* O'Reilly Media.

[13] Cavoukian, A. (2009). Privacy by design: The 7 foundational principles. Information and Privacy Commissioner of Ontario, Canada.

[14] Voigt, P., & Von dem Bussche, A. (2017). The EU General Data Protection Regulation (GDPR): A practical guide. Springer International Publishing.

[15] Sobczak, J., et al. (2023). The Future of SOAR: Integrating Intelligence and Automation. In *Proceedings of the 2023 ACM Workshop on Security Automation* (pp. 1-8).

[16] Ouyang, L., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 27730-27744.