# LLMエージェントを用いたフィッシングキャンペーン特定

千葉 大紀[1,a)]　中野 弘樹[1]　小出 駿[1]

**概要**：フィッシング攻撃は，クローキングに代表される回避技術を駆使し，フィッシングキャンペーンとして大規模かつ巧妙に展開されており，従来のコンテンツベース検知手法では対応が困難となっている．本稿ではこの問題に対し，適応型マルチエージェント手法 PhishLumos を提案する．本手法の独自性は，検知回避の発生を失敗ではなく，調査を開始する重要シグナルと捉え，攻撃インフラの深層分析へと調査をピボットさせる点にある．大規模言語モデル（LLM）を搭載したエージェント群が自律的に連携し，共有ホスティング，TLS 証明書，ドメイン名登録パターンといったインフラ上の繋がりを解明することで，単一の URL からキャンペーンの全体像を特定する．実データ評価において，本手法はキャンペーンの特徴を捉えた検知ルールを自動生成し，セキュリティ専門家が個別の URL を悪性と判定する時点よりも中央値で約 8 日早く，キャンペーンの全体像を特定可能であることを実証した．このルールは，同一キャンペーンに属する未知のフィッシングサイトを網羅的に発見する実用的な脅威インテリジェンスとして機能し，攻撃の未然防止や被害拡大の抑制に貢献できる．

# Phishing Campaign Identification Using LLM Agents

Daiki Chiba[1,a)]　Hiroki Nakano[1]　Takashi Koide[1]

***Abstract:*** Phishing campaigns increasingly use advanced evasion techniques like cloaking, challenging conventional content-based detection. We introduce PhishLumos, an adaptive multi-agent system that treats evasion not as a failure, but as a signal to pivot its investigation to the underlying attack infrastructure. LLM-powered agents collaborate to uncover infrastructure connections, such as shared hosting and TLS certificates, revealing an entire campaign from a single seed URL. Evaluated on real-world data, our method generates campaign-wide detection rules identifying campaigns over a week before expert confirmation. These rules enable proactive detection of unknown sites, helping mitigate widespread harm.

## 1. はじめに

フィッシング攻撃は，機密情報窃取を目的とするソーシャルエンジニアリング攻撃であり，依然として深刻な脅威となっている．攻撃者の手口は巧妙化・大規模化し，個別のサイトを散発的に設置する手口に留まらず，多数の類似サイトを同時に展開する「フィッシングキャンペーン」としての活動が主流となっている．その際，攻撃者の多くはセキュリティスキャナによる自動検知から逃れるため，クローキングに代表される高度な回避技術を用いる [1], [2]．クローキングは，アクセス元が人間かボットかを判別し，前者には悪性コンテンツを，後者には良性コンテンツを返す，といった表示内容を動的に変化させる技術である．このため，コンテンツ分析を基盤とする従来の検知手法や，それによって生成されるブロックリストの多くは，その有効性を大きく損なわれている．このような攻撃者の TTPs（Tactics, Techniques, Procedures）の進化は，防御側との間に深刻な非対称性を生み出しており，攻撃の変化に追従できる，より能動的な対策技術が急務となっている．

Web ページから得られる情報が乏しい状況において，コンテンツベースの検知手法は機能不全に陥るという本質的な限界を抱えている．攻撃者は，CAPTCHA 認証の背後にページを隠したり，セキュリティスキャナに対して意図的に 404 エラーを返したりすることで，この弱点を突く [3], [4]．コンテンツへの依存というこの問題は，大規模

[1] NTT セキュリティホールディングス & NTT
NTT Security Holdings Corporation & NTT, Inc.
[a)] daiki.chiba@ieee.org

言語モデル（LLM）を用いた最新の検知手法 [5], [6] においても本質的に変わらない．LLM は豊富なコンテンツから悪意を推論する能力に長けているが，分析対象のコンテンツ自体が存在しない，あるいは意図的に無害化されている状況では，その性能を発揮できない．この問題を解決するには，従来とは異なる新たなアプローチが不可欠である．

本稿では，攻撃者がフィッシングサイトを容易に増殖させる一方で，防御側は依然としてセキュリティ専門家による手動分析に依存するという，サイバーセキュリティにおける非対称性の問題に立ち向かうべく，適応型マルチエージェント手法「PhishLumos」を提案する．PhishLumos は，単一の URL を起点とし，専門家が行うような調査を自動化することで，フィッシングキャンペーン全体を解明する．**図 1** に示すように，本手法の独自性は，クローキングのような検知回避の挙動を，検知の失敗ではなく，むしろ攻撃インフラの深層分析へと踏み出すための重要な「シグナル」として捉え直す点にある．この逆転の発想に基づき，PhishLumos は最小限の手がかりから調査方針を動的にピボットさせ，LLM を搭載したエージェント群が自律的に連携することで，共有ホスティング，TLS 証明書，ドメイン名登録パターンといった攻撃インフラ上の繋がりを解き明かす．最終的に，本手法は検証済みの検知ルールを自動生成し，実用的な脅威インテリジェンスとして提供する．

本研究の貢献は，フィッシングキャンペーンの早期緩和に繋がる，実用的な脅威インテリジェンスを提供できる点にある．我々は，実際のフィッシング攻撃データを用いた評価を通じて，本手法が公的機関の専門家による確認よりも平均して 1 週間以上早くキャンペーンを特定可能であることを実証した．本稿が示すアプローチは，事後的な URL 単位の対処から，キャンペーン全体を標的としたプロアクティブな緩和へと防御のパラダイムを転換するものであり，攻撃による大規模な被害の未然防止に貢献する．

## 2. 関連研究

フィッシングキャンペーンの検知は，特に攻撃者がクローキングのような高度な回避技術を用いて分析可能な情報を制限する場合，依然として大きな課題となっている．従来の研究では，コンテンツベースの分析，回避技術への対策，そして LLM の活用といったアプローチが探求されてきた．しかし，これらの手法は，分析に足る Web ページのコンテンツが利用可能であることを前提としており，コンテンツへの依存という本質的な限界を共有している．また，その多くは個別の悪性 URL の特定に焦点を当てており，最小限の初期情報から攻撃キャンペーンの全体像を解明し，プロアクティブな防御に繋がる脅威インテリジェンスを生成することに特化してはいない．本節では，先行研究を概観し，PhishLumos の位置付けを明確にする．

**コンテンツベースのフィッシング検知．** コンテンツベース

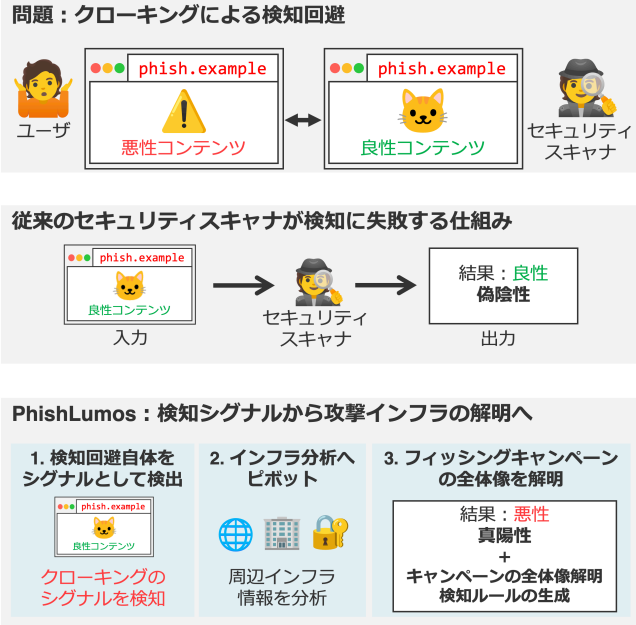


**図 1** PhishLumos の動作概念
**Fig. 1** The core concept of PhishLumos

の検知はフィッシング対策の基盤技術であるが，コンテンツが隠蔽されている，あるいは極端に乏しい状況下では，その有効性が低下する．ロゴ検知 [7] や視覚的類似性 [8]，あるいは Web ページの意図識別 [9] に基づく手法は強力であるものの，その成功は分析可能なデータが豊富に存在することに依存する．クローキングや最小限のコンテンツしか提示しない状況では，これらの手法は性能の維持に苦慮する．PhishLumos は，直接的なコンテンツ検査に依存するのではなく，攻撃インフラの分析へとピボットすることでキャンペーン全体を解明し，この問題に対処する．

**回避技術と対策．** クローキングに代表される回避技術と，その対策についても多くの研究が行われてきた．初期の研究には Web クローキングの体系的な分析 [10] や，クライアントサイドの回避技術の詳細な調査 [2] がある．また，他の研究でもクローキングが依然として有効な戦術であることが示されている [3], [11]．これらに対する対策技術として，CAPTCHA を突破する PhishDecloaker [1] や，スキャナの検知を逆手に取る手法を分析する PhishPrint [4] が提案されている．これらのアプローチが回避技術の解明や迂回を目的とするのに対し，PhishLumos は，コンテンツにアクセスできないという事実そのものを，検知の失敗ではなく攻撃インフラ調査を開始するための重要な「シグナル」として捉え直す点で独自性を持つ．このシグナルが，キャンペーンレベルの脅威インテリジェンス生成の起点となる．

**LLM の活用．** LLM の登場により，フィッシング検知のための強力な手法が次々と提案されている．しかし，これらの手法の多くもまた，アクセス可能な Web ページの情報に依存するという本質的な限界を抱えている．例えば，参

表 1　PhishLumos を構成するエージェントの役割
**Table 1**　Roles and Responsibilities of Agents in PhishLumos

| エージェント種別 | 主要な目的 | 主な入出力 |
|---|---|---|
| 監督エージェント | 分析フロー全体の指揮 | 入力：URL，出力：フィッシングキャンペーン検知ルール |
| 専門エージェント | | |
| ・URL 分析エージェント | URL の過去のスキャン履歴を取得 | 入力：URL，出力：スキャン履歴，スキャン時ステータス，IP アドレス |
| ・ドメイン名分析エージェント | 過去の DNS 履歴を分析 | 入力：ドメイン名，出力：IP アドレス履歴，利用パターン |
| ・IP 分析エージェント | IP アドレス上のドメイン名を分析 | 入力：IP アドレス，出力：関連ドメイン名，過去の悪性活動 |
| ・証明書分析エージェント | TLS 証明書の履歴を分析 | 入力：ドメイン名，出力：発行者パターン，共有証明書 |
| 統合エージェント | | |
| ・キャンペーン分析エージェント | フィッシングキャンペーンを特定 | 入力：全収集データ，出力：キャンペーン種別，主要指標 |
| ・ルール生成エージェント | 検知ルールを生成・検証 | 入力：主要指標，出力：検証済みのフィッシングキャンペーン検知ルール |

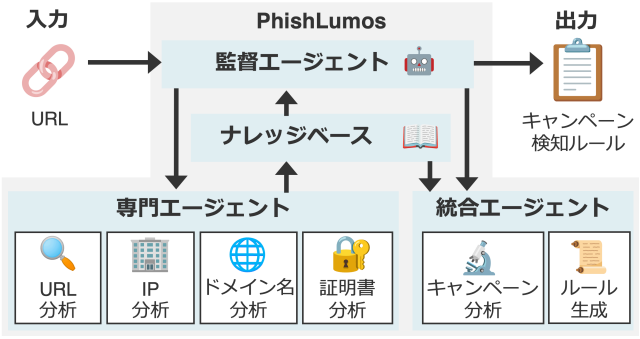


**図 2**　PhishLumos のアーキテクチャ
**Fig. 2**　PhishLumos Architecture

照ページに基づいて検知精度を向上させる研究 [12] や，クロールした Web データ（HTML，テキスト，スクリーンショット）から悪意を分析するために LLM の推論能力を活用する研究 [5], [6], [13], [14] がある．また，LLM を用いて特定のフィッシング手口を検知する手法 [15], [16], [17] も存在する．これらのアプローチが豊富なコンテンツを前提として高い性能を発揮するのに対し，PhishLumos は分析可能なコンテンツが乏しい状況に最適化されている点が根本的に異なる．PhishLumos における LLM エージェントの役割は，コンテンツの分析に留まらない．むしろ，最小限のデータから調査方針を動的に決定し，外部データソースを横断するピボット分析を実行し，断片的な手がかりを統合してキャンペーンの全体像を構築した上で，実用的な検知ルール，すなわち脅威インテリジェンスを生成することが，その中心的な機能となる．

## 3.　提案手法：PhishLumos

### 3.1　手法概要

本稿では，適応型マルチエージェント手法「PhishLumos」を提案する．本手法は，特にコンテンツが乏しい状況下で，クローキングのような高度な回避技術を用いるフィッシングキャンペーンを特定するために設計されている．従来の検知手法が情報の欠如に直面して機能不全に陥るのとは対照的に，PhishLumos は LLM の高度な推論能力を駆使し，過去の DNS レコード，IP アドレス情報，TLS 証明書といった多様な外部データソースを横断する「ピボット分析」を自律的に実行する．

**図 2** に本手法のアーキテクチャを示す．PhishLumos は，分析全体を指揮する「監督エージェント」，特定のデータ収集を担う複数の「専門エージェント」，そして収集された情報を統合して結論を導き出す「統合エージェント」から構成される．分析は，単一の URL 入力から始まる．監督エージェントは初期情報を評価し，攻撃インフラの繋がりを発見するために最も有望な調査の軸足（ピボットポイント）を特定する．例えば，初期スキャンで過去に悪性ドメイン名と関連付けられた IP アドレスが判明した場合，監督エージェントは IP 分析エージェントのタスクを優先する．各専門エージェントは，それぞれが持つツールを用いて情報を収集し，構造化された分析結果を中央リポジトリである「ナレッジベース」に集約する．このプロセスを通じて，監督エージェントは攻撃の全体像を逐次的に把握し，ナレッジベースの進化に応じて後続のタスクを動的に順序付け，キャンペーンの痕跡を体系的に解明していく．

### 3.2　脅威モデル

本研究が対象とする脅威モデルは，認証情報の窃取を目的とした大規模なフィッシングキャンペーンであり，その技術的な特徴は以下の通りである．攻撃者は，広く利用可能なフィッシングキットを駆使し，クローキングロジックを実装した Web サイトを容易に展開する．このクローキングは，セキュリティスキャナには良性コンテンツを提示し，標的のユーザにのみ悪性コンテンツを配信するよう設計されている．PhishLumos の分析対象は，単発の攻撃ではなく，このような類似した TTPs を持つサイトを多数同時に展開するフィッシングキャンペーンである．

### 3.3　基本構成とワークフロー

提案手法のアーキテクチャは，LLM による推論能力を各エージェントに内包させると同時に，エージェント間の連携を構造化されたフレームワークで制御するという，2 つの基本原則に基づいている．この設計により，PhishLumos は新たに見つかった証拠に応じて調査戦略を適応的に変化させ，静的な分析フローでは見逃してしまうような攻撃者インフラの繋がりを発見することが可能となる．

分析のワークフローは，まず監督エージェントが URL

分析エージェントに初期調査を指示することで開始される．監督エージェントは，この初期調査の結果を評価し，次に取るべき最も効果的なアクションを決定する．そのために，データ収集を担う専門エージェント群と，より高次の分析を行う統合エージェント群から成るチームに対し，タスクを動的に割り当てる．**表 1** に各エージェントの役割を示す．各エージェントは，脅威インテリジェンスソースを照会するための専用ツールを使用し，その結果で中央のナレッジベースを更新する．このプロセスを通じて，当初は断片的に見えた情報が関連付けられ，攻撃の全体像がナレッジベース内に構築されていく．

### 3.4 適応的なピボット分析

本手法の最大の独自性は，適応的なピボット分析にある．これは，静的な分析パイプラインとは根本的に異なり，監督エージェントが，進化し続けるナレッジベースに基づいて専門エージェントに動的にタスクを割り当てることで，調査全体を指揮するアプローチである．各ステップで，監督エージェントの LLM は Algorithm 1 で示される `DecideNextAction(KB, History)` 関数を実行し，収集された証拠全体を評価して最も有望なピボット先を決定する．このマルチエージェント設計は，モジュール性，スケーラビリティ，そして関心の分離という特徴を持つ．これにより，監督エージェントは戦略的意思決定に，専門エージェントは戦術的なデータ収集に，それぞれ専念できる．

分析ループの中核では，LLM が一連のトリガーに基づいてナレッジベース全体を評価し，最も有望な調査の方向性を見つけ出す．トリガーとしては，即時トリガー（例：新規登録ドメイン名が無関係な大手サイトにリダイレクトされるといった，クローキングを示唆する重大な挙動）や，インフラトリガー（例：蓄積された証拠から，非 CDN の IP アドレス上に無関係な複数のドメイン名が同居しているといった，不審なホスティングパターンが明らかになること）が挙げられる．このような状況認識に基づいたトリガーベースのオーケストレーションにより，PhishLumos は分析リソースを効率的に割り当て，各キャンペーンが用いる特有の TTPs に調査の焦点を適応させることができる．

### 3.5 データ収集とコンテキスト管理

PhishLumos は，適応的な調査を支えるため，データ収集とコンテキスト管理を体系的に行う．外部インテリジェンスは，urlscan.io [18] や VirusTotal [19] といった任意の脅威インテリジェンス API から取得可能である．これにより，過去の Web サイトスキャン，DNS レコード，IP インテリジェンス，TLS 証明書属性といった，攻撃インフラの解明に不可欠なデータセットにアクセスできる．

LLM のコンテキスト長の制約内で効果的に動作するため，PhishLumos は体系的なコンテキスト管理戦略を実装

**Algorithm 1** オーケストレーションロジック

1: **Input:** Initial URL $u_0$
2: **Initialize:** Knowledge Base $KB \leftarrow \emptyset$, History $H \leftarrow \emptyset$
3: { 初期データの取得 }
4: $task \leftarrow (\text{URLAnalysisAgent}, u_0)$
5: $findings \leftarrow \text{ExecuteTask}(task)$
6: $KB \leftarrow KB \cup findings$
7: $H \leftarrow H \cup \{(task, findings)\}$
8: { 知識駆動型の反復的な分析ループ }
9: **while** $\text{InvestigationIsActive}(KB, H)$ **do**
10: { 監督エージェントが全体状況を評価 }
11: $action, params \leftarrow \text{LLM.DecideNextAction}(KB, H)$
12: { 例：クローキングや不審なインフラパターンを評価 }
13: **if** $action = \text{PIVOT\_TO\_INFRASTRUCTURE}$ **then**
14: $agent, \text{pivot\_data} \leftarrow params$
15: $task \leftarrow (agent, \text{pivot\_data})$
16: $findings \leftarrow \text{ExecuteTask}(task)$
17: $KB \leftarrow KB \cup findings$
18: $H \leftarrow H \cup \{(task, findings)\}$
19: **else if** $action = \text{CAMPAIGN\_ANALYSIS}$ **then**
20: **break** { 統合分析のためループを抜ける }
21: **else**
22: **break** { 決定的でなければ終了 }
23: **end if**
24: **end while**
25: { 最終的な統合とルール生成 }
26: $\text{campaign\_profile} \leftarrow \text{CampaignAnalysisAgent}(KB)$
27: **if** $\text{IsCampaignIdentified}(\text{campaign\_profile})$ **then**
28: $rules \leftarrow \text{RuleGenerationAgent}(\text{campaign\_profile})$
29: **return** $\text{campaign\_profile}, rules$
30: **else**
31: **return** $\text{campaign\_profile}, \text{NoRulesGenerated}$
32: **end if**

している．この戦略には，API の生データから必須フィールドのみを抽出する「選択的データ抽出」，分析結果を標準化された要約形式でやりとりする「構造化されたエージェント間通信」，各専門エージェントがツール出力から主要なパターンを特定する「階層的な要約」，そして監督エージェントが調査の全体像を維持しつつ新たな知見を統合する「動的なコンテキスト維持」といった機能が含まれる．この統合的な戦略は，LLM が広範な外部データソースから抽出された適切な情報に基づいて効果的に推論し，深く多面的な調査を実現する上で不可欠である．

### 3.6 キャンペーン分析とルール生成

PhishLumos の分析フローにおける最終段階では，「統合エージェント」が中心的な役割を担う．この段階は，収集された情報からキャンペーンの全体像を明らかにする「キャンペーン分析」と，それに基づき検知ルールを生成する「ルール生成」という，2 つの処理で構成される．この一連の処理は，Algorithm 1 の 25–32 行目に示すように，それまでの反復的な調査ループが完了した後に実行される．

**キャンペーン分析．**キャンペーン分析エージェントは，最終的なナレッジベースに集約された情報を横断的に分析

**表 2** キャンペーン分析エージェントが用いる分類基準
**Table 2** Classification Criteria for the Campaign Analysis

| キャンペーン種別 | 主要な判断指標 |
|---|---|
| 悪性コンテンツ確認 | 共有インフラ上の同一ブランドのなりすましや，同一コンテンツハッシュの存在. |
| クローキング | 入力 URL と最終到達 URL の不一致（特に正規サイトへのリダイレクト）. |
| インフラ再利用 | 過去に悪性判断されたドメイン名と IP アドレスや証明書の共有（コンテンツは異なる）. |
| コンテンツ取得不能 | コンテンツ取得不可（例: 404 エラー）だが，攻撃インフラが既知のキャンペーンと関連. |
| 分類不能 | 確実な分類を行うには相関的な証拠が不十分. |

し，キャンペーンの全体像を構築する．**表 2** に示す判断基準に基づき，LLM が技術的な痕跡を関連付け，キャンペーンのプロファイルを決定する．これには，共有 IP インフラ，共通のドメイン名登録パターン，TLS 証明書を介した繋がり，そして活動の時系列的なクラスタリングの特定が含まれる．エージェントはキャンペーンを分類し，その決定的な特徴を構造化された `KeyIndicators` として抽出し，後続のルール生成フェーズに渡す．

**ルール生成．**抽出された `KeyIndicators` は，ルール生成エージェントに渡される．このエージェントは，Algorithm 2 に示す複数ステップの手順を実行する．第一に，LLM が多様な候補ルール（$R_{cand}$）を生成する．第二に，誤検知によって正規のサービスに影響を与えるリスクを避けるため，各候補ルールは，実際のデータを用いてその有効性が検証される．このステップは，誤検知率が高くなる可能性のある過度に広範なルールを防ぎ，生成される脅威インテリジェンスの実用性を担保するために不可欠である．エージェントは脅威インテリジェンス API に小規模なデータサンプルを照会し，LLM にそのルールの精度と文脈的関連性を評価させる．例えば，新たに見つかったドメイン名が，初期シードと命名パターンやホスティング特性を共有しているかを確認する．最終的な出力は，検証済みのルールのセット（$R_{final}$）であり，脅威インテリジェンス（例：構造化された検索クエリ）として提供される．

### 3.7 分析例

本節では，PhishLumos の適応的なワークフローを実証するため，実際のフィッシングキャンペーンを基にした分析例を解説する．なお，各指標はセキュリティ上の理由から匿名化している．分析の起点となる URL は `https://portal-service.example.test/`である．この時点では，この URL が悪性である直接の兆候はない．

**初期の URL 分析．**URL 分析エージェントが `portal-service.example.test` のスキャン結果を照会したところ，ページのアクセスは成功（ステータス 200 OK）であったが，コンテンツからはブランドのなりすましは検出されなかった．当該ドメイン名の名前解決結果は IP アドレス `203.0.113.55` であり，一般的な発行者

**Algorithm 2** キャンペーンルールの生成と検証

```
Input: Campaign Profile P (with KeyIndicators I_k)
Initialize: Candidate Rules R_cand ← ∅, Final Rules R_final ← ∅
{ 多様なルール候補を提案 }
prompts ← LLM.CreateGenerationPrompts(I_k)
R_cand ← LLM.GenerateRulesFromPrompts(prompts)
{ 各ルール候補を実データで評価 }
for all r ∈ R_cand do
    query ← FormatAsApiQuery(r)
    { 少量のデータサンプルで検証 }
    sample_results ← ThreatIntelAPI.search(query)
    if sample_results is not empty then
        { サンプルから適合率と再現率を評価 }
        precision, coverage ← LLM.AssessSample(sample_results, P)
        { サンプルのキャンペーンプロファイルへの関連性を評価 }
        r.score ← CalculateScore(precision, coverage)
    else
        r.score ← 0
    end if
end for
{ 最適なルールを選択・推奨 }
R_final ← SelectBestRulesByScore(R_cand)
return R_final
```

（例: `Generic-Cert-Issuer`）による TLS 証明書を使用していた．直接的な悪意の証拠がないため，監督エージェントはインフラ分析へと調査の軸足を移す．

**IP 分析へのピボット．**監督エージェントは IP 分析エージェントに `203.0.113.55` の調査を指示する．IP 分析エージェントは，この IP アドレスが過去に他のドメイン名をホストしていたことを発見する．重要なことに，それらのドメイン名の一つ `secure-auth.example.org` が，数週間前に特定の金融サービス `Service-A` のブランドなりすましとして検出されていた．この発見が最初の大きな突破口となり，一見無害な入力 URL が，共有インフラを介して悪性活動と結びついた．

**証明書分析へのピボット．**並行して，証明書分析エージェントはホスティングインフラに関連する証明書を調査する．その結果，一般的な発行者であること，そして証明書の有効期間が短いことを確認する．これはフィッシングキャンペーンでよく見られる戦術である．

**キャンペーン分析による統合．**キャンペーン分析エージェントは，収集されたすべての証拠を統合する．最も重要な証拠は，IP 分析エージェントによって確立された繋がり，すなわち入力 URL の IP アドレス（`203.0.113.55`）が，最近 `Service-A` を標的とした確定フィッシングサイト（`secure-auth.example.org`）のホストに使用されていたという事実である．現在の URL 自体はブランドを騙っていないが，共有インフラという強力な繋がりが存在する．エージェントはこれを「インフラ再利用」キャンペーンと高い信頼度で分類する．

**ルールの生成と評価．**ルール生成エージェントは，主要な指標（共有 IP アドレス `203.0.113.55` と過去に関連付けられたブランド `Service-A`）を受け取る．そして，このキャンペーンを網羅的に検知するための論理的なルールを策定する．例えば，「最終到達ページの IP アドレスが `203.0.113.55` であり，かつ以下のいずれかが真である URL を検知する: (a) `Service-A` のブランドなりすましが検出される，(b) 入力ドメイン名がある特定の期間内に観測されている」といったルールが生成される．このルールは検証済みクエリとして具体化され，キャンペーンの活動範囲を特定することに成功した．

この例は，PhishLumos がピボット分析を用いて，初期 URL 単体からは見えない繋がりをいかにして明らかにするかを示すものである．IP アドレスを分析の転換点として扱うことで，一見無害に見える証拠からフィッシングキャンペーンの特定に至った．

## 4. 評価

本節では，提案手法 PhishLumos の有効性を実証するための評価実験について述べる．評価の目的は，(1) フィッシングキャンペーンを早期に特定し，実用的な脅威インテリジェンスを生成する有効性，(2) 従来手法では検知が困難なコンテンツ取得不能シナリオにおける優位性，(3) 生成される検知ルールの品質と新規性，そして (4) 本手法が採用するマルチエージェントアーキテクチャの合理性，の 4 点を明らかにすることである．

### 4.1 実験設定

**データセット．**PhishLumos の評価には，フィッシング URL のデータセットと，誤検知分析のための正規 URL のデータセットという，2 種類のデータセットを構築した．フィッシング URL のデータセットは，JPCERT/CC が公開しているフィッシングサイト報告データ [20] を基に，2023 年 1 月から 2025 年 3 月までの期間で構築した．JPCERT/CC の公開データは，同組織のセキュリティ専門家が悪性コンテンツの存在を実際に確認し，ブランドを特定した時点の情報である．したがって，この確認時点よりも前にキャンペーンを特定できれば，その手法の「先行性」を実証できる．このデータから，我々は 103 件のキャンペーンを特定・検証し，これに属する 6,020 件のユニークな URL を抽出した．正規 URL のデータセットとしては，ChatPhishDetector の研究 [6] で用いられた，企業サイトや人気 Web サイトを含む 1,000 件を利用した．

**コンテンツ取得の困難性．**本稿が取り組む課題の重要性を裏付けるため，我々のデータセット内の URL に対し，一般的なセキュリティスキャナがコンテンツ取得を試みた際の成功率を分析した．その結果，全 URL の 77.0%（キャンペーン数では 82.5%）において，人間がブラウザでアクセスすれば表示されるコンテンツが，スキャナからはクローキングやアクセスエラーによって直接取得できなかった．この内訳は，38.8%がクローキング，37.6%がアクセス不能（例：403/404 エラー），残りの 23.5%が悪意の判断が困難な曖昧なコンテンツの表示であった．この問題は年々深刻化しており，コンテンツ取得不能シナリオの割合は 2023 年の 48.7%から 2024 年には 95.4%に急増し，2025 年初頭に観測されたキャンペーンでは 100%に達した．この現実は，コンテンツベースの検知手法が持つ本質的な限界と，PhishLumos のように攻撃インフラの分析に主眼を置くアプローチの必要性を明確に示している．

**表 3** キャンペーン検知における総合性能評価（全 103 キャンペーン）
**Table 3** Overall Performance on Campaign Mitigation

| 評価種別 | 評価指標 | 平均 | 中央値 |
|---|---|---|---|
| 有効性 | キャンペーン網羅率（%） | 93.0 | 100.0 |
| | 新規発見 URL 数 | 751.4 | 297.0 |
| 先行性 | 検知先行時間（時間） | 529.1 | 192.8 |
| 効率性 | 処理時間（秒/URL） | 63.9 | 60.4 |
| | 処理コスト（米ドル/URL） | 0.25 | 0.23 |

**表 4** コンテンツ取得の可否による性能比較
**Table 4** Performance Comparison by Content Accessibility

| シナリオ | 手法 | 適合率 | 再現率 | F1 値 | 偽陽性率 |
|---|---|---|---|---|---|
| コンテンツ取得可能（18 件） | PhishLumos（提案） | 0.947 | 1.000 | 0.973 | 0.001 |
| | ChatPhishDetector | 0.621 | 1.000 | 0.766 | 0.011 |
| | Phishpedia | 0.882 | 0.833 | 0.857 | 0.002 |
| | VisualPhishNet | 0.033 | 0.667 | 0.064 | 0.347 |
| コンテンツ取得不能（85 件） | PhishLumos（提案） | 0.988 | 1.000 | 0.994 | 0.001 |
| | ChatPhishDetector | 0.725 | 0.341 | 0.464 | 0.011 |
| | Phishpedia | 0.333 | 0.012 | 0.023 | 0.002 |
| | VisualPhishNet | 0.000 | 0.000 | 0.000 | 0.347 |

**比較手法．**PhishLumos の性能を，コンテンツベースの検知手法である ChatPhishDetector [6]，Phishpedia [7]，VisualPhishNet [8] と比較した．これらの手法は，主としてアクセス可能な Web ページのコンテンツ（テキスト，画像，ロゴ）に依存して個別の URL を分類する．PhishLumos については，あるキャンペーンに対する有効な検知ルールが生成された場合，その調査の起点となった URL を正しく検知したものとして扱う．

**実装．**PhishLumos は Python 3.11.12 環境で実装し，マルチエージェントのワークフロー管理には LangGraph ライブラリ（v0.4.5）を用いた．全エージェントの LLM には Azure OpenAI の `gpt-4.1-2025-04-14` モデルを使用した．実験結果の再現性を確保するため，ハイパーパラメータは固定値（`temperature`=0, `seed`=42）に設定した．外部データ収集には，urlscan.io [18] の API を利用した．

### 4.2 キャンペーン検知の有効性と先行性

最初に，提案手法の主目的である，単一 URL から攻撃キャンペーン全体を早期に検知する性能を評価した．**表 3** に主要な結果を示す．

PhishLumos は優れた性能を示し，キャンペーン網羅率

**表 5** PhishLumos の分析能力を示す代表的なケーススタディ
**Table 5** Representative Case Studies of PhishLumos's Analysis Capabilities

| ケース | 攻撃者の TTPs | 生成されたルール | 分析内容：ルールの新規性と実用性 |
|---|---|---|---|
| 金融 A | セキュリティスキャナで<br>コンテンツ取得不能 | Initial Domain: *.co.jp.*.test AND<br>Date:[2023-09-02 TO 2024-02-27] | コンテンツ取得不能な URL からドメイン名へとピボットし，<br>不審なドメイン名パターンを発見．405 件の未知 URL を発見. |
| 金融 B | 巧妙なクローキング実施<br>（正規サイトへリダイレクト） | Initial URL: */[unique_string]* AND<br>Date:[2024-11-26 TO 2024-12-24] | 偽装リダイレクトを無視し，URL パス内の固有文字列を<br>キャンペーンの痕跡として特定．540 件の未知 URL を発見. |
| 運輸 C | 巧妙なクローキング実施<br>（公式サイトへリダイレクト） | Initial Domain: *.suspicious.test AND<br>Final Domain: official.example AND<br>Date:[2023-04-24 TO 2023-05-15] | 不審な初期ドメインと正規の最終ドメインの不一致を検知<br>することでクローキングを看破．281 件の未知 URL を発見. |
| 通販 D | セキュリティスキャナにのみ<br>403 エラーを応答し検知回避 | Final IP: 203.0.113.43 AND<br>Final Status: 403 AND<br>Date:[2024-03-30 TO 2024-07-26] | 「403 Forbidden」という回避戦術自体を検知シグナル<br>として活用するルールを生成．65 件の未知 URL を発見. |
| 通販 E | IP アドレスを頻繁に変更し<br>スキャナから検知回避 | Final ASN: AS64500 AND<br>Final Domain: *.test AND<br>Date:[2023-06-14 TO 2024-01-01] | 短期的に変化する IP アドレスではなく，上位の AS 番号<br>（ASN）を対象とすることで回避耐性の高いルールを生成.<br>4,118 件の未知 URL を発見. |

**表 6** 各構成要素の重要性評価（指標：キャンペーン網羅率）
**Table 6** Component Importance via Ablation Study

| システム構成 | 平均網羅率 | 中央値網羅率 |
|---|---|---|
| PhishLumos（全機能） | 93.0% | 100.0% |
| w/o ピボット分析 | 56.1% | 84.8% |
| w/o ドメイン名分析 | 62.4% | 90.0% |
| w/o IP 分析 | 56.6% | 84.8% |
| w/o 証明書分析 | 60.2% | 87.7% |
| w/o キャンペーン分析 | 49.0% | 48.1% |
| w/o ルール生成・検証 | 26.5% | 2.2% |

の中央値は 100%に達した．これは，生成された検知ルールが，特定のキャンペーンに属する既知の URL のほぼすべてを識別できたことを意味する．さらに，本手法が生成する脅威インテリジェンスが，未知の脅威をプロアクティブに発見する高い能力を持つことも確認された．元の正解データには含まれていなかった 77,391 件の URL を新たに発見し，これらを VirusTotal [19] で評価したところ，そのうち 62,321 件（80.5%）が後に少なくとも 1 つ以上のセキュリティエンジンによって悪性と判定された．

本手法がもたらす最大の利点は，検知の「先行性」にある．これは，PhishLumos がルールを生成した時点から，JPCERT/CC がそのサイトの悪性コンテンツを確認するまでの時間差を計測したものである．結果として，本手法は JPCERT/CC による悪性確認よりも中央値で 192.8 時間（約 8 日）早くキャンペーンを特定した．この時間的な優位性は，ブラウザベンダやセキュリティ企業が広範な被害発生前にブロッキングルールを展開するといった，実用的な事前防御策を講じる上で重要である．分析は効率的で，URL 1 件あたりの処理時間の中央値は 60.4 秒であった．

### 4.3 コンテンツ取得不能シナリオにおける性能評価

本手法の優位性は，従来のコンテンツベースの検知手法が機能しないコンテンツ取得不能シナリオにおいて特に顕著であった．セキュリティスキャナが最終的なフィッシングコンテンツにアクセスできたか否かに基づいてデータセットを「取得可能」と「取得不能」に分割し，比較手法との性能を評価した．評価には適合率（検知の正確さ），再現率（正解の網羅率），F1 値（適合率と再現率の調和平均），そして偽陽性率（正規 URL の誤検知率）を用いた．

**表 4** に示すように，すべてのコンテンツベース手法の性能は，コンテンツ取得不能シナリオ下で著しく低下した．Phishpedia や VisualPhishNet のような画像ベースの検知手法は，ロゴやサイトレイアウトといった視覚的な手がかりがないため，F1 スコアがそれぞれ 0.023，0.000 となり，ほぼ無力であった．LLM ベース手法である ChatPhishDetector でさえ，分析すべきテキストコンテンツが不足しているため，再現率が 0.341 まで低下した．

対照的に, PhishLumos は性能を維持し, F1 スコア 0.994, 再現率 1.000 を達成した．この堅牢性は，変化しやすいコンテンツではなく安定した攻撃インフラのパターンを分析するという，本手法の基本設計そのものに起因する．これにより，フィッシングページがクローキングされている，あるいはアクセス不能な場合でも，脅威を特定できる．

### 4.4 ケーススタディによる定性分析

**表 5** に，提案手法の分析能力を示す 5 つの代表的なケーススタディを提示する．なお，セキュリティの観点から，表中に記載のドメイン名，IP アドレス，AS 番号は，実際のキャンペーンの特徴を維持しつつ匿名化したものである．これらの事例は，本手法がゼロに近い情報からピボットする能力（金融 A），一見しただけでは分からない URL パス内の痕跡を発見する能力（金融 B），高度なクローキングを特定する能力（運輸 C），サーバの応答から攻撃パターンを一般化する能力（通販 D），そして自律システム番号（ASN）レベルでインフラを捉えることで堅牢なルールを生成する能力（通販 E）を持つことを示している．特に，ルール生成エージェントが，過度に広範であったり一時的であったりするルール候補を自動的に破棄し，検証済みのルールのみを出力する機能は，最終的な脅威インテリジェンスの精度と持続性を保証する上で不可欠であった．

### 4.5 構成要素の重要性分析

最後に，提案手法の各構成要素が性能に与える影響を評価するため，要素を個別に取り除いた場合の性能変化を測

定した．**表 6** の結果は，すべての構成要素が本手法の成功にとって不可欠であることを裏付けている．

性能低下が最も顕著だったのは，「ルール生成エージェント」の機能を単純化した構成である．これは，多様なルール候補を複数生成・検証するプロセスを経ずに，LLM が最良と考えるルール候補のみを生成する方式を試したもので，キャンペーン網羅率の中央値がわずか 2.2%にまで落ち込んだ．この結果は，実用的な脅威インテリジェンスを作成する上で，多様な仮説（ルール候補）を生成し，それらを検証するというプロセスが最も重要であることを示している．また，本手法の中核である「ピボット分析」の機能を無効化した場合も，平均網羅率が 93.0%から 56.1%へと大幅に低下した．このことは，コンテンツからインフラへと調査の軸足を移すことの重要性を裏付けている．他のいずれのエージェントを除去した場合も性能は著しく低下し，本手法が採用する協調的なマルチエージェントアーキテクチャの合理性が示された．

## 5. 議論

**研究倫理．**本研究における開発および評価は，すべて公開情報に基づいて構築したデータセットを用いており，ユーザの個人情報やその他の機密情報は一切含まれていない．外部 API を用いたデータ収集にあたっては，各サービスの利用規約を遵守し，サーバへの過度な負荷を避けるよう配慮した．これにより，研究の倫理性を確保すると同時に，第三者による再現可能性を担保している．

**制約事項．**本手法の性能は，分析の中核を担う LLM の能力に大きく依存する．LLM のモデル更新により，分析結果が変動する可能性がある．また，本手法は外部の LLM API を利用するため，実行にはコストを要する．しかし，同等性能を持つ LLM の推論コストは低下しており [21], [22]，コスト面の制約は将来的に緩和されることが期待される．

## 6. おわりに

巧妙化・大規模化するフィッシングキャンペーンの脅威に対し，本稿は適応型マルチエージェント手法「PhishLumos」を提案した．本手法の独自性は，検知回避を調査開始のシグナルと捉え直し，LLM エージェント群が自律的なピボット分析によって攻撃インフラの繋がりを解明する点にある．これにより，コンテンツに依存する従来手法の本質的な限界を克服し，キャンペーンの全体像を特定する．実データ評価において，本手法は専門家による悪性確認よりも中央値で約 8 日早くキャンペーンを特定し，その活動を網羅する検知ルール，すなわち実用的な脅威インテリジェンスを自動生成可能であることを実証した．本研究が提示する，URL 単位の事後対処からキャンペーン単位のプロアクティブな緩和へのパラダイム転換は，攻撃による大規模被害の未然防止に貢献するものである．

## 参考文献


[1] Teoh, X. et al.: PhishDecloaker: Detecting CAPTCHA-cloaked Phishing Websites via Hybrid Vision-based Interactive Models, *Proc. USENIX Security* (2024).

[2] Zhang, P. et al.: CrawlPhish: Large-scale Analysis of Client-side Cloaking Techniques in Phishing, *Proc. IEEE S&P* (2021).

[3] Oest, A. et al.: PhishFarm: A Scalable Framework for Measuring the Effectiveness of Evasion Techniques against Browser Phishing Blacklists, *Proc. IEEE S&P* (2019).

[4] Acharya, B. et al.: PhishPrint: Evading Phishing Detection Crawlers by Prior Profiling, *Proc. USENIX Security* (2021).

[5] Liu, R. et al.: Less Defined Knowledge and More True Alarms: Reference-based Phishing Detection without a Pre-defined Reference List, *Proc. USENIX Security* (2024).

[6] Koide, T. et al.: ChatPhishDetector: Detecting Phishing Sites Using Large Language Models, *IEEE Access* (2024).

[7] Lin, Y. et al.: Phishpedia: A Hybrid Deep Learning Based Approach to Visually Identify Phishing Webpages, *Proc. USENIX Security* (2021).

[8] Abdelnabi, S. et al.: VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity, *Proc. ACM CCS* (2020).

[9] Liu, R. et al.: Inferring Phishing Intention via Webpage Appearance and Dynamics: A Deep Vision Based Approach, *Proc. USENIX Security* (2022).

[10] Invernizzi, L. et al.: Cloak of Visibility: Detecting When Machines Browse a Different Web, *Proc. IEEE S&P* (2016).

[11] Oest, A. et al.: PhishTime: Continuous Longitudinal Measurement of the Effectiveness of Anti-phishing Blacklists, *Proc. USENIX Security* (2020).

[12] Li, Y. et al.: KnowPhish: Large Language Models Meet Multimodal Knowledge Graphs for Enhancing Reference-Based Phishing Detection, *Proc. USENIX Security* (2024).

[13] Cao, T. et al.: PhishAgent: A Robust Multimodal Agent for Phishing Webpage Detection, *Proc. AAAI* (2025).

[14] Wang, Y. et al.: Can You Walk Me Through It? Explainable SMS Phishing Detection using LLM-based Agents, *Proc. SOUPS* (2025).

[15] Koide, T. et al.: ChatSpamDetector: Leveraging Large Language Models for Effective Phishing Email Detection, *Proc. SecureComm* (2024).

[16] Bitaab, M. et al.: ScamNet: Toward Explainable Large Language Model-Based Fraudulent Shopping Website Detection, *Proc. AAAI* (2025).

[17] Nakano, H. et al.: ScamFerret: Detecting Scam Websites Autonomously with Large Language Models, *Proc. DIMVA* (2025).

[18] urlscan.io: `https://urlscan.io/` (2025).

[19] VirusTotal: `https://www.virustotal.com/` (2025).

[20] JPCERT/CC: phishurl-list, `https://github.com/JPCERTCC/phishurl-list/` (2025).

[21] Chiba, D. et al.: DomainLynx: Advancing LLM Techniques for Robust Domain Squatting Detection, *IEEE Access* (2025).

[22] Appenzeller, G.: Welcome to LLMflation – LLM inference cost is going down fast, `https://a16z.com/llmflation-llm-inference-cost/` (2024).