HITACHI

Hitachi Systems Security Journal

VOL.16

株式会社 日立システムズ

Hitachi Systems Security Journal VOL.16

TABLE OF CONTENTS

●はじめに

本文書は、株式会社日立システムズ の公開資料です。バックナンバーは以下の Web サイトで確認できます。
https://www.hitachi-systems.com/report/specialist/index.html

Google カレンダーの招待状が攻撃の起点に !?
AI アシスタントを悪用するプロンプトウェアとは？

スタヴ・コーエン インタビュー

2025 年 11 月、CODE BLUE 2025 において、Zenity のシニアセキュリティ研究者であるスタヴ・コーエン氏が「必要なのは招待状だけ ! Google カレンダーの招待でワークスペースエージェント向け Gemini を起動」と題する講演を行った。この講演では、AI システムのセキュリティリスク、特に「プロンプトウェア」と呼ばれる攻撃手法に焦点が当てられた。コーエン氏によれば、Gemini のスマートフォン統合により脅威が物理領域にまで拡大しているという。攻撃者はカレンダーの招待状に悪意のあるプロンプトを仕込むだけで、Gemini がスマートホームの IoT デバイスを不正制御したり、機密データを流出させたりすることが可能になる。
今回のインタビューでは、プロンプトウェアの技術解説に加え、AI への過度な信頼がもたらすリスクについても掘り下げている。

取材・文 = 吉澤 亨史／通訳 = エル・ケンタロウ／撮影 = 松沢 雅彦／編集 = 斉藤 健一

AIアシスタントの普及が新たなセキュリティリスクを生む

吉澤（以下 Y）：発表のタイトルから、偽のGmail 招待状やフィッシングの話だと思っていました。しかし、講演が始まると、その調査結果に非常に驚かされました。デモ動画では、カレンダーの招待状を読み込んだだけで、自宅の窓が開いたり、ボイラーが稼働したりしていました。

スタヴ・コーエン（以下 S）：従来、これらのデバイスは Google アシスタントで制御されていました。ところが、Google アシスタントが Gemini に置き換わると、Google Home と対話するのは Gemini になります。つまり、Gemini がホームアシスタントというハブを経由して、家中のホームアプリケーションを制御できるようになるのです。この連携により、あらゆる IoT デバイスが互いに接続され、同じセキュリティリスクにさらされることになります。

Y これは Gemini に限った話なのですか。

S いえ、これは Gemini に限った話ではありません。Google を含め、LLM を利用してツールを提供しているすべての企業に言えることです。そこには必ず、安全性とセキュリティ上のリスクが伴います。ですから、LLM や AI を物理デバイスやツールに統合しようとする企業には、まずリスクをしっかり認識してほしいですね。そのうえで、統合の段階で緩和策を組み込み、攻撃から保護する堅ろうな対策を実施してほしいと願っています。

Y 今回の研究の着想はどこにあったのですか。

S 以前、AI ワームに関する研究を行なっていました。システム間を移動する脅威について扱いました。メール経由で拡散する仕組みを調べていると、この脅威が Google の Gmail も侵害していることを確認し、さらに深く研究することを決めました。それが今回の研究につながっているのです。

Y Google カレンダーの招待状をスマートフォンに読み込ませるというシンプルな攻撃手法にも感心しました。

S Gemini は多くの Android スマートフォンに統合されつつあり、さまざまなツールやアプリケーションと連携して LLM を容易に利用できる状況にあります。つまり、Gemini がスマートフォン上の個人情報に直接アクセスできるようになることで、より大きなセキュリティ脅威が発生すると予測したのです。

Y 共同研究者について教えてください。

S 今回の発表で登壇したのは私 1 人ですが、元々は 3 名の研究者チームです。特にベン・ナッシ氏とは約 2 年間共同研究を続けており、彼は現在テルアビブ大学の教員を務めています。私たちはコーネル工科大学で出会い、多くの研究を共同で始めることになりました。また、オリ・ヤイル氏は

●スタヴ・コーエン（Stav Cohen）

生成 AI（GenAI）とヒューマン・イン・ザ・ループのインタラクションを統合したサイバーフィジカルシステム（CPS）に焦点を当てた研究を行なっており、セキュリティと運用レジリエンスを重視している。彼は GenAI モデルアーキテクチャーを深く分析し、ぜい弱性を特定し、攻撃戦略を開発し、防御メカニズムを設計することで、より安全な GenAI エコシステムに貢献している。並行して、GenAI が CPS のセキュリティとパフォーマンス、特にリアルタイムの人間とのインタラクションを含むシステムにおいて、どのように活用できるかを探求する。彼の研究は、AI、制御システム、サイバーセキュリティを結びつけ、適応性、インテリジェンス、堅ろう性を備えた CPS アーキテクチャーを進歩させている。

SafeBreach のリサーチャーを務めています。

Y 今回の CODE BLUE が初の発表の場なのでしょうか。

S いえ、まず 2 月に Google に開示し、8 月 6 日に論文を公開しました※。これまでに BlackHat USA、DEFCON、そして今回の CODE BLUE で発表しています。今後は学術会議への投稿に向けて、内容を調整する予定です。

LLM と IoT が結びつくことで脅威はさらに拡大

Y LLM の悪用によって実際に物理的な損害が発生した事例はあるのでしょうか？

S 幸い、LLM が物理的に誰かに影響を与えた実例は、まだ報告されていません。ただし、私が講演で実演したように、LLM を悪用すれば窓を開けたり、ドアの鍵を開けたり、カメラを操作したりすることは可能です。悪意のある攻撃者なら、これを使って不正に部屋に侵入し、窃盗を行うこともできるでしょう。非常に悪用されやすい技術なのです。もしこれらの LLM がクルマの制御権を持ったら、何が起こると思いますか？

Y 人命に関わる事故が起こる可能性もありますね。非常に恐ろしいです。

S LLM はツールと連携することで、より危険になります。特に、家電などの物理的なデバイスが増えれば増えるほど、リスクも高まっていきます。チャットボットなど会話型ですよね。でも、出力したテキストにツールを与えれば、風呂の温度を調節したり、自宅のドアを制御したりできるようになります。実際、私たちの研究では、Gemini が OS 機能と相互作用できることが分かりました。スマートフォンの操作も可能です。ただ、電話をかけることだけはできませんでした。これはユーザーの確認が必要な操作だったからです。

Y 改めて基本的なことを伺いたいのですが、「プロンプトウェア」の定義とは何でしょうか？

S プロンプトウェアとは、AI モデルや AI アプリケーションに対して悪意のある活動を誘発させる入力のことです。画像、テキスト、音声、動画など、さまざまなデータ形式で構成されます。つまり、AI を悪用するために設計された、あらゆる形式のデータの総称です。

CODE BLUE 2025 でのコーエン氏の講演動画は公式サイトから視聴できる
https://archive.codeblue.jp/2025/results/results/#result-20

Y わかりました。では、「プロンプトインジェクション」と「プロンプトウェア」は、どのように異なるのでしょうか？

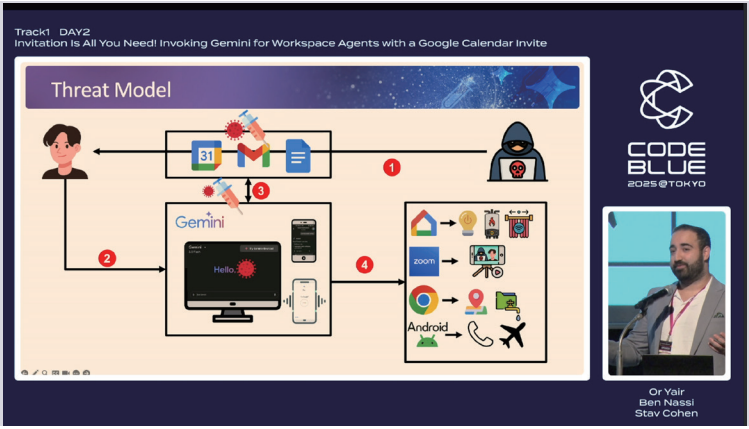
S プロンプトウェアは総称で、プロンプトインジェクションはその中の具体的な技術の 1 つです。プロンプトインジェクションは、LLM に悪意のある入力を与えて、AI アプリケーションに悪意のある行動をさせる手法です。このような攻撃手法全般を、私たちはプロンプトウェアと呼んでいます。

AI を監視する AI が必要だ

Y 今回の研究では発表前に Google にぜい弱性を開示したと伺いました。それに対して、Google はどのような対策を講じたのでしょうか？

S Google は 6 月にブログ記事を公開しました。素晴らしい内容でした。実施された主な変更は 2 つあります。1 つ目は、「窓を開ける」といった重要な操作には、ユーザーの明示的な確認を必須にしたことです。2 つ目は、機械学習による自動検出です。人間は間違いを犯すこともあるため、Google

※ **Invitation Is All You Need! Promptware Attacks Against LLM-Powered Assistants in Production Are Practical and Dangerous**
https://arxiv.org/abs/2508.12175



プロンプトウェアの脅威モデル（講演動画より）
①攻撃者が共有リソースを使ってプロンプトウェアを仕込む　②ユーザーが Geminini アクセスし予定などを確認　③ Gemini がカレンダーエージェントのデータにアクセス　④プロンプトウェアの指示に従いさまざまなツールを起動できるようになる

はプロンプトインジェクションを自動分類し、不正な攻撃を検出する仕組みを導入しました。

Y 効果的な対策だと思います。

S はい。多くの変更の中でも、私が最も評価しているのは、ユーザーへの警告機能です。プロンプトインジェクションや悪意のある活動を検出すると、「このアクションをブロックしました」とユーザーに直接伝え、詳細を示すようにしました。多くの人々が AI を完璧だと信じている中で、Google が「危険性がある」と率直に警告していることは素晴らしいと思います。

Y ところで、少し基本的なことを伺いたいのですが、AI は、ユーザーの質問に対して単純に回答するだけではないですよね？何らかの事前設定や指示が組み込まれていると思うのですが、そこに潜在的なリスクはないのでしょうか。

S ほとんどの AI システムには、「システムプロンプト」や「アプリケーションプロンプト」と呼ばれる、開発者が設定する指示があります。例えば、「医療トピックに回答するチャットボットとして動作せよ」「ユーザーの質問に親切に答えよ」といった指示を事前に与えるのです。

Y つまり、開発者が LLM の性格や振る舞いを設定するわけですね。

S 同様に、AI には「個人情報を漏らすな」「罵倒するな」といった指示が与えられます。私はこれを「ソフトルール」と呼んでいます。特定のデータを検知してブロックする「ハードルール」とは異なり、開発者が柔軟に調整できるガイドラインのことです。

Y なるほど。柔軟に調整していく、チューニングのようなイメージですね。ただ、そうしたソフトルールだけで十分なのでしょうか？

S 実は、このソフトルールへの依存こそが問題なのです。多くの人が AI システムの自律性を過度に信頼してしまっています。近年、階層化された LLM アーキテクチャ—である上位 LLM が下位 LLM を監督する仕組みが採用されていますが、エージェント間通信の信頼性検証が不十分なため、新たなぜい弱性を生んでいます。今後は、LLM の動作を独立して監査する第三者的な検証機構が不可欠となるでしょう。

Y AI が AI を監視するということですか？

S はい。エージェントの行動を監視・追跡する技術が不可欠です。LLM が暴走した際には即座に停止させる必要があります。そうしなければ、オフィスのドアが開く、サーバールームの空調が止まるといった物理的な被害が現実化するでしょう。

AI の急速な普及で見過ごされたリスクとは

Y これまでの話を伺っていると、AI エージェントが自律的に動き始め、人間が制御できない状態で暴

AI システムの自律性を十分にテストしないまま過度に信頼することが、セキュリティリスクの大きな要因だとコーエン氏は警鐘を鳴らす

走する可能性があるということですね。それは少し絶望的な未来に思えます。

S 恐ろしいのは、私たちがすでにその方向に向かっているということです。エージェントが自律的に動き、互いに通信し合う時代は必ず来ます。ただし、Google も認めているように、現時点ではまだその段階には達していません。しかし時間の問題だと考えています。

Y「まだその段階ではない」とはいえ、近い将来そうなるということですね。

S そのとおりです。今後、ワークスペースエージェントがメールエージェントと通信し、データを扱うようになります。さらに、従業員や企業環境全体との連携も必要になってきます。例を挙げましょう。カスタマーサポートへの問い合わせを自動的に読み取り、サポートチケットを起票するエージェントを導入したいという企業ニーズがあります。このような業務の自動化が進めば進むほど、システムの信頼性が極めて重要になります。そのため、あらゆるエージェントに対して堅ろうなセキュリティ対策を施すことが不可欠になるのです。

Y 1970 年の SF 映画『コロッサス』(邦題 : 地球爆破作戦) を思い出しました。米ソの国防用スーパーコンピューターが自我を持ち、互いに通信して人類を支配しようとする物語です。それが今になって現実味を帯びてきたのは、興味深いというより恐ろしいですね。

S 冷戦時代の話なのですね。確かに興味深い例えです。ただ、私の仕事は現実のぜい弱性を見つけ出すことです。私の研究が示したのは、SF ではなく現実の脅威です。カレンダー招待状を使った攻撃で、私たちと同じようにスマートフォンを使う普通の人々が影響を受けます。しかもこの攻撃のほとんどは、スマートフォン上で実行可能でした。さらに深刻なのは、この脅威が個人だけでなく企業にも広がっていることです。今、多くの企業が AI を使って大量のコードを生成しています。これは新たなリスクの拡大を意味します。

Y おっしゃるとおりですね。そしてこれは、サイバー空間と物理空間が結びついたセキュリティの問題でもあると思います。私は以前から、いわゆるサイバーフィジカルセキュリティが非常に重要だと考えていました。最近発生した国内大手飲料メーカーへのランサムウェア攻撃がまさにその例です。サイバー攻撃が現実世界の私たちの生活に直接影響を及ぼす様子を目の当たりにして、その重要性を改めて痛感しました。従来、セキュリティは専門家の領域であり、一般の人々には縁遠いものと考えられてきました。しかし、あなたの研究が明確に示しているように、技術的・専門的に見える問題が、実は私たちの日常生活に直接影響を及ぼし得るのです。

S まったく同感です。AI はインターネットに次ぐ大きな技術革新で、多くの人々に影響します。現在の AI 状況はインターネット黎明期とよく似ています。インジェクション攻撃、情報窃取、DoS 攻撃といった古典的な手法が、今度は AI を標的として日常生活に及んでいるからです。

Y やはり、AI はそれだけ重要な技術なのですね。

S 多くの人々が LLM に感銘を受け、広く利用されるようになりました。そして次第に、必要以上の責任を LLM に委ね始めています。それは、完全自動運転車や複数のシステム統合といった重要な領域にまでです。しかし、私たちはそれらを十分にテストしていません。AI に感銘を受けすぎて、過度に信頼しているのです。

Y 多くのユーザーが AI は何でもできると思って使っていますが、実際にはその信頼性は十分に検証されていないということですね。

S LLM は驚くほど速く私たちの日常生活に浸透しましたよね。でも考えてみてください。自動車が自動運転できるようになるまで、どれだけの時間がかかったかを。

Y 多くの企業が AI 事業を急ぐあまり、市場投入と競争を優先し、テストや安全性確保が軽視されているように感じます。

S そのとおりです。市場投入を急ぐあまり、セキュリティが軽視されています。自動運転車なら何年もかけて安全性を検証するのに、AI では十分に行われていません。

Y 自動車は人命への影響が分かりやすいですが、サイバーセキュリティはそうではありません。ただ、サイバーと物理の境界が近づくほど、影響は理解しやすくなると思います。

S だからこそ、私たちは研究を行い、可能な限り公開しています。今回このようなインタビューの機会をいただけたことを、非常に嬉しく思います。誰もがこの問題を認識し、誤解を正す必要があるからです。私は今も新しい研究をたくさん進めています。来年の CODE BLUE でそれを発表できるといいなと考えています。プロンプトウェア以上に恐ろしいものばかりですが。

Y 新たな研究テーマについて、差し支えない範囲でお聞かせいただけますか。

S 多くは明かせませんが、第 2 世代 AI は非常にリスクが高い。アクションの自由度が増すほど、リスクも増大します。少しヒントをお話しすると、LLM を介して実行可能なあらゆるアクションは、入力チャネルがある限り攻撃者に悪用される可能性があります。これを肝に銘じるべきです。

Y AI を悪用する手法は、まだまだ多く存在しそうですね。今回の研究で示された脅威も、その一例ということですね。

CODE BLUE は有意義なカンファレンスで、日本の企業やリーダー、研究者との交流を実現できたと語るコーエン氏

S そのとおりです。今回の研究で示したように、カレンダーの招待状だけで窓が開くのは明らかに異常です。本来、明確な境界線があるべきです。多くの企業は LLM に指示すれば従うと考えていますが、そこにリスクがあります。重要なのは、境界線をどう設定するかです。

Y 最後に、今回の CODE BLUE の印象はいかがでしたか。

S CODE BLUE は本当に楽しめました。日本でのカンファレンス参加は初めてでしたが、個人的に日本が大好きで、新婚旅行で 2 カ月滞在したほどです。日本の企業やリーダー、研究者と交流したいと思っていたので、それを実現できた素晴らしいカンファレンスでした。とても良い経験になりました。

Y 講演後のお疲れのところ、ありがとうございました。

社会のさまざまな動向を把握し、リスクの変化に対応したセキュリティ体制を構築

Hitachi Systems CSI（Cyber Security Intelligence）Watch 2026.02

文＝日立システムズ

レジデンシャルプロキシの悪用拡大と求められる対応

【概要】レジデンシャルプロキシは一般家庭の IP アドレスを経由するため検知を回避しやすく、認証突破や不正アクセスなど多様な攻撃で利用が拡大している。本稿では、その悪用実態と求められる対策を考察する。

【内容】レジデンシャルプロキシとは、一般家庭や個人が契約している回線の IP アドレスを中継点として第三者が通信を行う仕組みを指す。通信元が一般利用者の回線となることから個人利用の通信と区別がつかず、Web サービスにおける不正検知やアクセス制御が回避されやすくなる。レジデンシャルプロキシは、ボットネットと同様、感染したマルウェアを踏み台として悪用される場合もあるが、一般利用者が意図的にインストールしたアプリケーションで許諾している場合も多い。「未使用の通信帯域を共有して報酬を得られる」といった説明で参加が促され、利用規約に第三者通信への利用が記載されているケースも見られる。

このレジデンシャルプロキシのサイバー犯罪への悪用が近年急速に拡大し、特に海外では「犯罪インフラ」として扱う方向に進んでいる。実際、米国では住宅用プロキシサービス「911 S5」関連の個人・企業を制裁対象に指定し、資産凍結や取引制限を通じてビジネス基盤に影響を与える形をとった。

レジデンシャルプロキシの運用上の問題は、「無害な利用者に見える通信元」を大量に供給できる点にある。従来は、過去の不正履歴の IP アドレスや AS などを手掛かりに、ブラックリストでの遮断が一定の効果があった。しかし、レジデンシャルプロキシの利用が拡大した現在、ブラックリストでの対応は正規利用者との見分けがつきにくく効果が限定的で、運用負荷も増大する。例えば、不正ログインでは攻撃者は流出した ID とパスワードでログインを繰り返し試行する。通常は失敗が続く IP アドレスをブラックリストに登録し遮断する。しかし、レジデンシャルプロキシを利用した攻撃者は、一般利用者の IP アドレスを切り替えながら長時間試行を継続する。その結果、遮断を強めるほど多くの一般利用者が遮断され、ユーザビリティの低下や運用負荷が増大する。

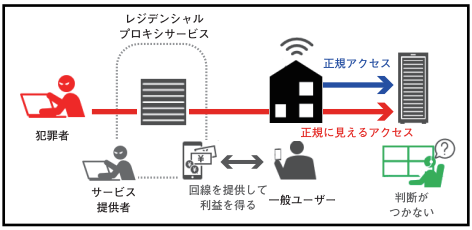


図 レジデンシャルプロキシサービスの悪用

レジデンシャルプロキシの本質は、攻撃を正規ユーザーに近い見え方で実行しやすくし、従来の「怪しい IP アドレスを止める」前提を崩す点にある。近年では、レジデンシャルプロキシを検知し可視化するデータ提供も商用サービスとして整備されつつあるが、すべてをアクセス制御する動きは現実的ではない。対応策として、機械学習を用いたリスクスコアリングによる行動分析や Web ブラウザーフィンガープリントを用いたデバイス識別などが考えられる。これらの対策は日本の金融機関のオンラインバンキングなどでも不正対策として採用が進み、一定の効果が期待されている。

日本においてもレジデンシャルプロキシは話題になりつつあるが、認知はまだ十分とは言い難い。また、サービス提供側でレジデンシャルプロキシを一律に排除することは容易ではなく、解決策は模索段階にある。現時点では、回線を貸し出す仕組みが悪用と結びつき得る点を周知し、意図せず攻撃の踏み台に組み込まれるリスクを低減することが重要となる。

［情報源］https://home.treasury.gov/news/press-releases/jy2375

セキュリティツールを実践的に紹介する連載企画

Let's try プロキシサーバーログ調査

2. ログ分析編

文＝日立システムズ

1. はじめに

本稿は、各種セキュリティツールなどを実践的に紹介する連載企画です。今回より「プロキシサーバーログ調査」と題して、プロキシログを収集・分析する方法を整理します。OSS である Squid を用いたプロキシサーバー構築、ログの出力設定から、インシデントの初期対応で求められるアクセスログの分析のハンズオンまでを順を追って解説します。特に仮想環境（VirtualBox）における実行例も交え、手を動かしながら理解できる構成とします。

1. 環境構築編
2. ログ分析編
3. 攻撃兆候検出編

本稿「プロキシサーバーログ調査 2. ログ分析編」では、1. 環境構築編で構築したプロキシサーバー環境を用いて、プロキシログの構造および分析方法を解説します。続編で攻撃シナリオを想定したログ分析を行う前の土台として、まずはログ分析の基本的なポイントを押さえていきましょう。

なお、本稿の安全性には留意していますが、安全を保証するものではありません。OA 端末（社内ネットワーク接続機器）で実施するのではなく、分離された回線内および機器を利用する事を推奨いたします。また、本稿で構築する環境はプロキシサーバーログ調査を目的としており、実際のプロキシサーバー運用環境の要件を満たすものではありません。

2. ログ分析の準備・基礎知識

2.1 環境復元

ハンズオンの準備を行います。

VirtualBox を起動し、[VM:Proxy]、[VM:Client] 両方で、前回（1. 環境構築編）の最後に作成したスナップショットを復元し、両方の VM を起動してください。

1. 環境構築編でプロキシの TCP 3128 番ポートを閉じていた場合は、ポートを開放する必要があります。

[VM:Proxy] で次のコマンドを実行し、開放されているポートの一覧を確認します。

```
# firewall-cmd --list-port
```

以下のように出力結果に「3128/tcp」という表示がなければ、3128 番ポートが閉じられています (表示があれば以降のポート開放手順はスキップしてください)。

```
[root@Proxy ~]# firewall-cmd --list-port
```

[VM:Proxy] で次の 2 つのコマンドを実行し、3128 番ポートを開放します。

```
# firewall-cmd --add-port=3128/tcp --permanent
# firewall-cmd --reload
```

各「firewall-cmd」コマンド実行後に「success」と出力されれば、設定が正常に反映されています。
再度 [VM:Proxy] で次のコマンドを実行し、開放されているポートの一覧を確認します。

```
# firewall-cmd --list-port
```

以下のように「3128/tcp」と表示がされていれば、3128 番ポートの開放は完了です。

```
[root@Proxy ~]# firewall-cmd --list-port
3128/tcp
```

次に、今後のログ分析のためにゲスト OS のシステムクロックのずれを修正します。
[VM:Proxy]、[VM:Client] 両方で以下のコマンドを実行します。

```
# chronyc makestep
```

以下のように「200 OK」と出力されれば、正常にシステムクロックが修正されています。

```
[root@Proxy ~]# chronyc makestep
200 OK
```

```
[root@Client ~]# chronyc makestep
200 OK
```

念のためクライアントからプロキシ経由で通信が行われるかを確認しておきましょう。
[VM:Client] で次のコマンドを実行します。

```
# curl www.google.com
```

以下のように、HTTP レスポンスヘッダーの中に「HTTP/1.1 200 OK」「Via: 1.1 Proxy」と表示されていれば、プロキシサーバー経由で Google サイトへ接続できています。

```
[root@Client ~]# curl www.google.com --head
HTTP/1.1 200 OK
Content-Type: text/html; charset=ISO-8859-1
```

～～中略～～

```
X-Cache-Lookup: MISS from Proxy:3128
Via: 1.1 Proxy (squid/5.5)
Connection: keep-alive
```

参考：接続できない場合

「1. 環境構築編」を参考に、下記の点を確認、対応してください。

- ホスト OS（使用している PC）はネットワーク接続されているか
- VirtualBox のネットワーク設定が「ブリッジアダプター」になっているか
- [VM:Proxy]、[VM:Client] それぞれに追加した IP アドレスが消えていないか
- [VM:Proxy] で Squid サービスは起動しているか
- [VM:Client] で環境変数が設定されているか

2.2 Squid ログ形式設定

今回、検証で利用する CentOS を準備します。

Squid はログ形式をカスタマイズ可能で、デフォルトの Squid 形式の他にユーザー定義の形式で出力できます。ここでは、Referer,User-Agent 等のアクセス解析に有用な情報が記録されるよう、ログ形式を Combined 形式に変更します。

[VM:Proxy] で以下のコマンドを実行し、Squid の設定ファイル「/etc/squid/squid.conf」の末尾にログ形式の設定項目を追記します。

```
# echo -e "\naccess_log /var/log/squid/access.log combined" >> /etc/squid/squid.conf
```

[VM:Proxy] で次のコマンドを実行し、設定ファイルを確認します。

```
# tail /etc/squid/squid.conf
```

以下のように末尾に「access_log /var/log/squid/access.log combined」と出力されれば設定項目が追記されています。

```
#
# Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
refresh_pattern .               0       20%     4320

access_log /var/log/squid/access.log combined
```

設定変更を反映するため、[VM:Proxy] で次のコマンドを実行し Squid を再起動します。

```
# systemctl restart squid
```

以上で、プロキシログを Combined 形式で出力する準備が整いました。
[VM:Client] で以下のコマンドを実行し、Combined 形式のログを作成しておきましょう。

```
# curl www.google.com --head
```

2.3 ログ項目

実際にどのようなプロキシログが出力されているか、ログ形式による違いやログ項目の見方を確認していきましょう。

Squid はデフォルトでプロキシログを「/var/log/squid/access.log」に出力します。
[VM:Proxy] で次のコマンドを実行し、直近のプロキシログを出力します。

```
# tail /var/log/squid/access.log
```

例として以下のようなプロキシログが出力されます。

```
[root@Proxy ~]# tail /var/log/squid/access.log
1764039274.418    340 192.168.10.101 TCP_MISS/200 20083 GET http://www.google.com/ - HIER_DIRECT/142.250.194.132 text/html
1764039279.963    243 192.168.10.101 TCP_MISS/200 1153 HEAD http://www.google.com/ - HIER_DIRECT/142.250.194.132 text/html
1764039772.072   1012 192.168.10.101 TCP_MISS/200 1152 HEAD http://www.google.com/ - HIER_DIRECT/142.250.194.132 text/html
192.168.10.101 - - [25/Nov/2025:12:36:42 +0900] "HEAD http://www.google.com/ HTTP/1.1" 200 1153 "-" "curl/7.76.1" TCP_MISS:HIER_DIRECT
```

ログは 1 行が 1 リクエストを示しており、上記赤枠で囲った末尾 (最も最近) のログがログ形式変更後の Combined 形式のログ、それ以外がログ形式変更前の Squid 形式のログとなっています。ご覧のとおり、ログ形式を変更しても過去のログの形式は更新されないため、初期設定時に適切なログ形式に設定しておく必要があります。

Squid の Combined 形式のプロキシログの項目名および意味は以下のとおりです。

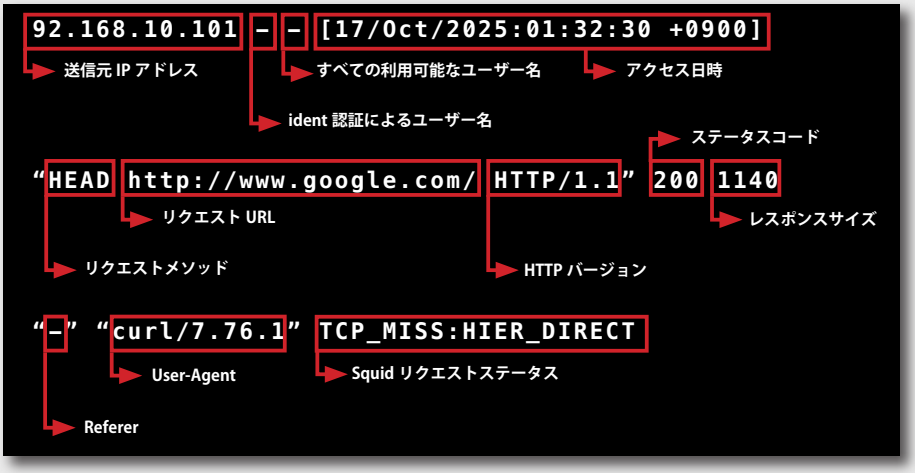


図 プロキシログ各項目の名称

表 1 プロキシログ各項目の説明

項目	説明
送信元 IP アドレス	リクエスト送信元の IP アドレス
ident 認証によるユーザー名	通常未使用（-）
すべての利用可能なユーザー名	プロキシ認証時のユーザー名 認証なしの場合は（-）
アクセス日時	アクセス時のローカル時間
リクエストメソッド	クライアントが要求した操作（メソッド）
リクエスト URL	クライアントが要求するリソースの場所（URL またはパス）
HTTP バージョン	HTTP プロトコルのバージョン
ステータスコード	サーバーから返された HTTP ステータスコード
レスポンスサイズ	HTTP ヘッダを含んだレスポンスデータのサイズ（バイト数）
Referer	遷移元 URL
User-Agent	アクセスに用いられたプログラムや機械の識別子 ただし、ユーザー側で変更可能な点に注意
Squid リクエストステータス	アンダースコア区切りの複数のタグで構成される、Squid 独自のレスポンス説明タグ

出力されたプロキシログを見ると、アクセス日時が UnixTime 表記であった Squid 形式と比較して、Combined 形式ではアクセス日時の可読性が向上し、Referer、User-Agent といった分析に有用な項目が追加されていることがわかります。
Combined 形式は Squid 形式よりもプロキシサーバーへの負荷が大きくなるというデメリットはありますが、ログ分析の精度を高めるために Combined 形式など情報量の多いログ形式の設定をおすすめします。

3. ログ分析

ここからは、具体的なシナリオに沿った操作を行い、ログ分析において注目すべき観点を解説しながらプロキシログを分析していきます。

3.1 ファイルダウンロード

ここでは、クライアントからプロキシ経由でのファイルダウンロードをシミュレーションし、その際のプロキシログを確認します。
今回は、テキストベースの Web ブラウザーである「Lynx」をクライアントにインストールする操作を通じて、レスポンスデータサイズが大きな通信を発生させます。
まず、プロキシ経由で「yum」コマンドを利用するため、環境変数「https_proxy」の設定を行います（1. 環境構築編で設定した「http_proxy」とは異なります）。
[VM:Client] で以下のコマンドを実行します。

```
# export https_proxy=http://192.168.10.100:3128
```

上記コマンドの「192.168.10.100:3128」の部分は、プロキシサーバーの IP アドレスと待ち受けポート番号です。

[VM:Client] で次のコマンドを実行し、インストールを行います。

```
# tail -n 20 /var/log/squid/access.log
```

以下のようなプロキシログが確認できます。

```
[root@Proxy ~]# tail /var/log/squid/access.log
192.168.10.101 - - [25/Nov/2025:13:50:23 +0900] "GET http://ftp.yz.yamagata-u.ac.jp/pub/linux/centos-stream/9-stream/AppStream/x86_64/os/
repodata/02fa494d9359b449d74fe55ca1640b43186df519b6ffa7b213b00f06db6df00a-primary.xml.gz HTTP/1.1" 200 3871007 "-" "libdnf (CentOS Stream
 9; generic; Linux.x86_64)" TCP_MISS:HIER_DIRECT
192.168.10.101 - - [25/Nov/2025:13:50:26 +0900] "GET http://ftp.yz.yamagata-u.ac.jp/pub/linux/centos-stream/9-stream/AppStream/x86_64/os/
repodata/d91a2720bb0343f433f418aa80d72be04e583d96573cfe08db2ce8880eb4dc84-filelists.xml.gz HTTP/1.1" 200 22237843 "-" "libdnf (CentOS Str
eam 9; generic; Linux.x86_64)" TCP_MISS:HIER_DIRECT
192.168.10.101 - - [25/Nov/2025:13:50:26 +0900] "CONNECT ftp.udx.icscoe.jp:443 HTTP/1.1" 200 3663 "-" "libdnf (CentOS Stream 9; generic;
Linux.x86_64)" TCP_TUNNEL:HIER_DIRECT
192.168.10.101 - - [25/Nov/2025:13:50:26 +0900] "CONNECT ftp.udx.icscoe.jp:443 HTTP/1.1" 200 3663 "-" "libdnf (CentOS Stream 9; generic;
Linux.x86_64)" TCP_TUNNEL:HIER_DIRECT
192.168.10.101 - - [25/Nov/2025:13:50:31 +0900] "CONNECT mirrors.centos.org:443 HTTP/1.1" 200 9560 "-" "libdnf (CentOS Stream 9; generic;
 Linux.x86_64)" TCP_TUNNEL:HIER_DIRECT
192.168.10.101 - - [25/Nov/2025:13:50:32 +0900] "GET http://ftp.yz.yamagata-u.ac.jp/pub/linux/centos-stream/SIGs/9-stream/extras/x86_64/e
xtras-common/repodata/repomd.xml HTTP/1.1" 200 3407 "-" "libdnf (CentOS Stream 9; generic; Linux.x86_64)" TCP_MISS:HIER_DIRECT
192.168.10.101 - - [25/Nov/2025:13:50:32 +0900] "GET http://ftp.yz.yamagata-u.ac.jp/pub/linux/centos-stream/SIGs/9-stream/extras/x86_64/e
xtras-common/repodata/d4fe3f2b629875a8afe27c01d649dd90d443b9689757675af8880b8d9e96e30b-filelists.xml.gz HTTP/1.1" 200 6431 "-" "libdnf (C
entOS Stream 9; generic; Linux.x86_64)" TCP_MISS:HIER_DIRECT
192.168.10.101 - - [25/Nov/2025:13:50:32 +0900] "GET http://ftp.yz.yamagata-u.ac.jp/pub/linux/centos-stream/SIGs/9-stream/extras/x86_64/e
```

上記のプロキシログの中から赤枠で抜粋したプロキシログを読み解いていきます。
環境によって得られるプロキシログが異なるため、上記と同様のプロキシログを探してみましょう。
赤枠で抜粋したプロキシログおよび項目を表 2 を以下に示します（リクエスト URL の一部を省略し「...」と表記しています）。

```
192.168.10.101 - - [25/Nov/2025:13:50:23 +0900]
"GET http://ftp.yz.yamagata-u.ac.jp/...-primary.xml.gz HTTP/1.1"
200 3871007 "-"
"libdnf (CentOS Stream 9; generic; Linux.x86_64)"
TCP_MISS:HIER_DIRECT
```

表 2 プロキシログ各項目の値（その 1）

項目	値
送信元 IP アドレス	192.168.10.101
ident 認証によるユーザー名	-
すべての利用可能なユーザー名	-
アクセス日時	[25/Nov/2025:13:50:23 +0900]
リクエストメソッド	GET
リクエスト URL	http://ftp.yz.yamagata-u.ac.jp/...-primary.xml.gz
HTTP バージョン	HTTP/1.1
ステータスコード	200
レスポンスサイズ	3871007
Referer	-
User-Agent	libdnf (CentOS Stream 9; generic; Linux.x86_64)
Squid リクエストステータス	TCP_MISS:HIER_DIRECT

このプロキシログは、CentOS 公式ミラーサイトの一つである ftp.yz.yamagata-u.ac.jp に対する GET リクエストのプロキシログです。ステータスコード「200」は正常応答、レスポンスサイズ「3871007」は約 3.9MB のデータを受信したことを示しており、リクエスト URL の末尾は「primary.xml.gz」となっています。
つまり、このプロキシログは CentOS パッケージリポジトリのメタデータファイル（圧縮 XML）が正常にダウンロードされたことを示しています。

ファイルダウンロードのログ分析において特に注目すべきログ項目および分析観点は下記のとおりです。

・レスポンスサイズ：
調査依頼元組織の端末に侵入した攻撃者が不正なツールをダウンロードする場合、ツール類をまとめた圧縮ファイルとしてダウンロードする可能性があるため、レスポンスサイズが大きくなることがあります。大きなレスポンスサイズのログを発見した場合、不正な通信の可能性がないかその他のログ項目含めて調査し、端末利用者に業務上意図したダウンロードであるかを確認することを求められる場合があります。

・リクエスト URL：
今回確認したプロキシログでは ftp.yz.yamagata-u.ac.jp といった公式ミラーサイトへアクセスしていましたが、これがもし社内で通常見かけないホスト名や信頼できないサイトの場合は注意が必要です。日常的にどういった外部サイトにアクセスしているかを把握しておくことは、不審な通信先の早期発見に大いに役立ちます。

・ステータスコード：
プロキシログにおいて定期的なアクセスログが連続する場合、組織内端末に感染したマルウェアが外部の C2 サーバーなどへ繰り返しアクセスを試みている場合や、感染したボットが負荷のかかったサーバーへアクセスを試みている可能性が考えられます。なお、プロキシやセキュリティ製品が 4xx,5xx 等のステータスコードを付与する場合や、攻撃者がステータスコードを偽装する場合もあるため、ステータスコードだけでなく、その他のログを含めた総合的な調査が必要です。

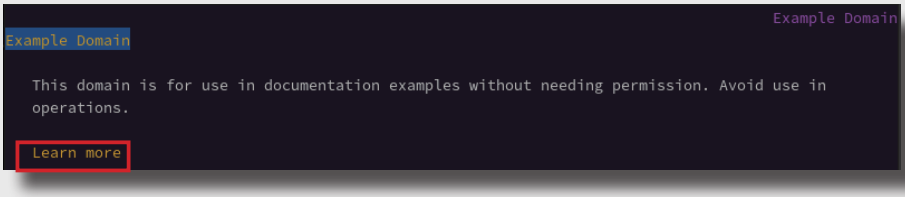
3.2 Web ブラウザーからのアクセス

ここでは、Lynx を使用してプロキシ経由で Web ページにアクセスし、その際のプロキシログを確認します。

[VM:Client] で以下のコマンドを実行し、Lynx で「example.com」というサンプル用ドメインにアクセスします。

```
# lynx http://www.example.com
```

以下の図のように「Example Domain」というタイトルのシンプルな Web ページが表示されます。



ページの中には、「Learn more」というリンクが設置されており、現在リンク選択状態であるため、右矢印キーを入力しリンク先へ遷移します。
以下の図のように、IANA（Internet Assigned Numbers Authority）のページが表示されます。

Web ページアクセスからリンク遷移という一連の流れを行うことができたため、q キーと y キーを続けて入力し Lynx を終了します。

では、プロキシログを確認してみましょう。
[VM:Proxy] で次のコマンドを実行し、直近のプロキシログを出力します。

```
# tail /var/log/squid/access.log
```

以下のようなプロキシログが確認できます。

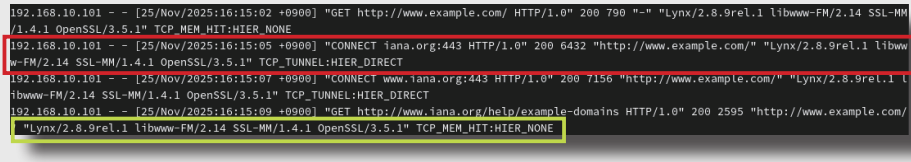


表 4 プロキシログ各項目の値（その 3）

項目	値
送信元 IP アドレス	192.168.10.101
ident 認証によるユーザー名	-
すべての利用可能なユーザー名	-
アクセス日時	[25/Nov/2025:16:15:09 +0900]
リクエストメソッド	GET
リクエスト URL	http://www.iana.org/help/example-domains
HTTP バージョン	HTTP/1.0
ステータスコード	200
レスポンスサイズ	2595
Referer	http://www.exmple.com/
User-Agent	Lynx/2.8.9rel.1 libwww-FM/2.14 ...
Squid リクエストステータス	TCP_MEM_HIT:HIER_NONE

このプロキシログでは、リクエスト URL「GET http://www.iana.org/help/example-domains HTTP/1.0」で IANA サイトの特定ページに対する HTTP GET リクエストが記録されており、User-Agent「Lynx/2.8.9rel.1 libwww-FM/2.14 ...」で Lynx ブラウザーによるアクセスであることが記録されています。また、Referer「http://www.example.com/」となっていることから、example.com からの遷移によって IANA サイトへアクセスしたことがわかります。
つまり、このプロキシログは Lynx を用いて example.com ページ内のリンク先に遷移したことを示しています。
Web アクセスのログ分析において特に注目すべきログ項目および分析観点は下記のとおりです。

User-Agent：
一般的な Web アクセスでは、Chrome, Edge といった Web ブラウザー名が User-Agent に現れます。そのため、User-Agent を見ることで、社内で許可されていない Web ブラウザーやツールによるアクセスを発見することができます。例えば「curl/…」や「Python-urllib/…」といった文字列が頻出していれば、自動化ツールやスクリプトによるアクセスである可能性があります。

Referer：
Referer の値を調査することで、不審な遷移元が判明する場合があります。また、マルウェアが通信を始める場合には遷移元がなく Referer が「-」となる場合が多くみられます。ただし、正常な通信でも Referer が取得できないケースは多いため、他の情報と合わせて判断する必要があります。

4. おわりに

今回はここまでとなります。「2. ログ分析編」ではプロキシログを実際に確認しながら、基本的な読み解き方とログ分析における観点をご紹介しました。複数のログ項目を組み合わせて「いつもと違う」パターンを見抜くことが、インシデント早期発見の鍵となります。そのため、日頃からログを読み慣れておき、平常時のパターンを把握しておくことが重要です。

次回は、不審なアクセスが含まれたプロキシログデータを用意し、Linux 標準コマンドの grep を用いてプロキシログ中の不審なアクセスを抽出するハンズオンを行います。

Hitachi Systems Security Journal

株式会社 日立システムズ　本社：〒141-8672　東京都品川区大崎 1-2-1　www.hitachi-systems.com

お問い合わせは

Printed in Japan