HITACHI



Hitachi Systems Security Journal

VOL.82

株式会社 日立システムズ

Hitachi Systems Security Journal VOL.82

TABLE OF CONTENTS

複数のぜい弱性を組み合わせて制御を奪うエクスプロイト・チェーン !!
数百万台の自動車に影響を及ぼす “Perfekt Blue” とは !?
ミハイル・エフドキモフ インタビュー …… 3

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 2026.10 …… 9

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 Windows イベントログ調査
3. Hayabusa によるイベントログ調査 攻撃痕跡調査編 ……10

●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

●ご利用条件
本文書内の文章等すべての情報掲載に当たりまして、株式会社日立システムズ（以下、「当社」といいます。）といたしましても細心の注意を払っておりますが、その内容に誤りや欠陥があった場合にも、いかなる保証もするものではありません。本文書をご利用いただいたことにより生じた損害につきましても、当社は一切責任を負いかねます。
Microsoft、Windows は、マイクロソフト グループの企業の商標です。
NVIDIA は、米国およびその他の国における NVIDIA Corporation の商標または登録商標です。
その他記載の会社名、商品名は、それぞれの会社の商標または登録商標です。
本文書に掲載されている情報は、掲載した時点のものです。掲載した時点以降に変更される場合もありますので、あらかじめご了承ください。
本文書の一部または全部を著作権法が定める範囲を超えて複製・転載することを禁じます。

© Hitachi Systems, Ltd. 2026. All rights reserved.



複数のぜい弱性を組み合わせて制御を奪うエクスプロイト・チェーン!!
数百万台の自動車に影響を及ぼす“Perfekt Blue”とは!?

Mikhail Evdokimov

ミハイル・エフドキモフ インタビュー

2025 年 11 月に開催されたセキュリティ国際会議「CODE BLUE 2025」において、PCA のシニアセキュリティリサーチャーであるミハイル・エフドキモフ（Mikhail Evdokimov）氏が登壇した。同氏は、自動車業界で広く採用されている OpenSynergy 社のカスタム Bluetooth スタック「BlueSDK」に存在する複数のぜい弱性を組み合わせた攻撃手法「Perfekt Blue（パーフェクトブルー）」を披露し、会場の大きな注目を集めた。本稿では、講演の概要を解説するとともに、エフドキモフ氏へのメール・インタビューをお届けする。

撮影＝松沢雅彦／取材・文＝斉藤 健一

複数メーカーの車両に脅威を及ぼす “Perfekt Blue” とは

2025 年 11 月に開催されたセキュリティ国際会議「CODE BLUE 2025」において、PCA Cyber Security（旧 PCAutomotive）のシニアセキュリティリサーチャーであるミハイル・エフドキモフ（Mikhail Evdokimov）氏が登壇した。同氏は、フォルクスワーゲン、メルセデス・ベンツ、シュコダ、BMW、アウディなど、多数の自動車メーカーで広く採用されてい OpenSynergy 社のカスタム Bluetooth スタック「BlueSDK」に存在する複数のぜい弱性を組み合わせた汎用ワンクリック攻撃手法「Perfekt Blue（パーフェクトブルー）」を披露し、会場に大きな衝撃を与えた。

現代の自動車において、車内のインフォテインメント（IVI）システムはエンターテインメントやナビゲーション、各種通信を処理する中心的存在であると同時に、外部からの攻撃者にとって車内システムへ侵入するための主要な標的となっている Perfekt Blue は、メモリ破損や論理的バグといった 4 つのぜい弱性を巧みに連鎖させることで、ASLR や CFI といった標準的な各種セキュリティ軽減策をすべて無効化し、無線経由でシステム上のリモートコード実行（RCE）を達成する技術である。実車を用いたデモではほぼ 100% の安定性で制御を奪うことに成功しており、プライバシー情報の奪取にとどまらず、CAN バスをはじめとする制御系ネットワークへ横展開（ラテラルムーブメント）するための足がかりを与えるという極めて重大な脅威を提示した。

ぜい弱性を連鎖させて完全制御にいたる 3 段階のエクスプロイトチェーン

本攻撃手法は、Bluetooth プロトコル群の不備を突いた不正書き込み、ASLR を突破するアドレス漏えい、任意関数の実行という 3 つの技術的プロセスを連鎖させ、堅牢なシステム保護を無効化してリモートコード実行（RCE）を達成する仕組みである。

1. 不正な書き込み状態の作成

Bluetooth の通信処理（AVRCP）において、使い終わったメモリー領域の後処理が不十分な不備を突き、Use-After-Free 状態を誘発する。ここに、メモリを狙ったデータで埋め尽くす「ヒープスプレー」という手法を組み合わせることで、メモリーの好きな場所へデータを自由に書き込める状態（任意アドレス書き込み）を作り出す。

2. 隠されたメモリー配置の特定（アドレス漏えい）

続いて、セキュリティ機能（ASLR）によってランダム化されているメモリ上の配置を特定するため、内部のアドレス情報を外部へ漏えいさせる処理（アドレスリーク）を行なう。ここでは、シリアル通信を担う「RFCOMM」のテスト機能の不備と、データ転送を担う「L2CAP」で通信 ID に「ゼロ」を設定できてしまう問題を組み合わせる。切断時のチェック処理をうまくスキップさせることで、テストの応答データと一緒に内部のコードアドレスを外部へ漏えいさせ、プログラムの配置基準点（ベースアドレス）の特定を実現する。

3. システムの完全制御（RCE 達成）

特定した配置基準点をもとに、L2CAP のイベント処理を悪用した「好きな関数を呼び出す仕組み」と、AVTP プロファイルの機能を応用した「好きな場所のデータを読み取る仕組み（AAR）」を構築する。これまでそろえた仕掛け（書き込み・読み取り・関数実行）をすべて統合してシステムの管理命令を実行し、ネットワーク（TCP/IP）経由で外部から操作用コマンドを送れる状態（リバースシェル）を起動することで、システムを完全に遠隔操作する「リモートコード実行（RCE）」を達成する。

検証対象にはフォルクスワーゲン ID.4 が使われた。なお、この攻撃手法は、メルセデス・ベンツ、フォルクスワーゲン、シュコダ、BMW、アウディをはじめとする多数の自動車メーカーの車両に対し、ワンクリックでリモートコード実行（RCE）を可能にするものだという。

本誌では、エフドキモフ氏にメールでインタビューを行なった。引き続きその内容をお届けする。



●ミハイル・エフドキモフ
（Mikhail Evdokimov）
PCAutomotive のセキュリティ評価チームのシニアセキュリティリサーチャー。脆弱性調査、エクスプロイト開発、リバースエンジニアリングを専門としている。主な研究分野は、ワイヤレス技術、組み込みシステム、カーネルなど。

エフドキモフ氏へのインタビュー

1. 研究対象の選定について

Q：講演を拝見しました。利用可能なさまざまなプロトコルスタックの中で、なぜ研究対象として具体的に OpenSynergy の「BlueSDK」を選ばれたのでしょうか。また、この一連のぜい弱性を発見するに至った最初の手がかりは何でしたか？

A：Bluetooth 攻撃ベクターは数年前から私の研究テーマであり、この通信プロトコルについてかなり確固たる知識と理解を深めてきました。そのため、BlueSDK の文脈でこの領域の分析を始めるのはごく自然なことでした。例として、私が過去に行った自動車関連の他の講演※1でも、Bluetooth を主な侵入経路として取り上げました。

OpenSynergy の BlueSDK について言えば、完全に偶然でした！私たちは PCA Cyber Security（旧 PCAutomotive）社に代わってフォルクスワーゲン ID.4 について個人的な研究を行なっており、当初、ぜい弱性を発見した時点ではその Bluetooth スタックが BlueSDK に関連しているとは結び付けていませんでした。しかし、追加分析を行なったところ、以前に私たちがエクスプロイトに成功したシュコダ・スペルブ（Skoda Superb）でも（もちろん多少の変更はあるものの）同じ Bluetooth スタックが使用されていることが判明しました。その後、BlueSDK が他にどこで使用されているのか興味を持ち、最終的に認定製品を指定している公開された Bluetooth 認証情報に基づいて OSINT（公開情報調査）分析を実施しました。これにより、認証データ内の詳細に基づいて、Bluetooth フレームワークとして BlueSDK を使用している潜在的にぜい弱なデバイスのリストを作成することができました。

2. エクスプロイト開発における課題

Q：ASLR や Shadow Stack といったさまざまな標準的セキュリティ軽減策をバイパスし、ほぼ 100% の安定性を持つエクスプロイトの作成に成功しました。しかし、講演の中ではエクスプロイトの安定性を向上させるのが「本当に困難だった」とも述べられています。エクスプロイト開発プロセスにおいて最も困難だった具体的な技術的障害について詳しく教えていただけますか？

A：技術的な部分に深く入り込みすぎない範囲でお話しします。エクスプロイトの安定性に関する最大の悩みの種は、Bluetooth スタック全体で利用されている通信の状態を管理する仕組み（有限状態マシン）の制御が複雑で、データを正常に処理し続けるには数多くの細かい条件を満たす必要があったためです。

もう 1 つの大きなハードルは、プロセスヒープ領域内の不安定さでした。これは、メモリ破損のぜ

※ 1 ミハイル・エフドキモフ氏がこれまでに登壇したセキュリティカンファレンスと講演スライド
- **BlackHat Asia 2025**: https://i.blackhat.com/Asia-25/Asia-25-Evdokimov-Remote-Exploitation-of-Nissan-Leaf.pdf
- **BlackHat Europe 2024**: https://i.blackhat.com/EU-24/Presentations/EU-24-Parnishchev-OverTheAirVW.pdf
- **Hexacon 2024:** https://2024.hexacon.fr/slides/Evdokimov-0_click_RCE_on_the_IVI_component.pdf



ミハイル・エフドキモフ氏の講演アーカイブは CODE BLUE 公式サイトから視聴できる
https://archive.codeblue.jp/2025/results/results/#result-9

い弱性を利用してリモートエクスプロイトを行なう際のよくある問題です。ユーザー空間プロセスが実行されると、通常、複数のスレッドが独立してヒープメモリの領域を割り当て・解放するため、ヒープ領域がかなり断片化されます。任意アドレス書き込み（AAW）などの攻撃の足がかり（エクスプロイトプリミティブ）を得るには、与えられたぜい弱性が許すプリミティブに基づいて、適切なヒープレイアウトを「形成」する必要があります。デバッグ機能がなかったため、標準的な Linux ユーザー空間アロケーターに関する十分な知識だけを頼りに完全に手探りで改善を行なう必要があり、この問題の解決には時間がかかりました。

さらに別の問題は、不特定のタイミングで有効化されるタイマーに関連するものでした。数回は完全に問題なく動作したのに、作成したヒープレイアウトに影響を与えるリソースを解放するタイマーが作動し、理由もなくエクスプロイトが失敗するケースを想像してみてください。これらの障害をすべて解決するのは興味深いものでしたが、非常に時間がかかるものでした。そのため「ほぼ」100% の安定性にとどまっています。全体的なエクスプロイトのパフォーマンスに影響を与えることがめったにない問題については、解決を見送る判断をしました。

3. 今後の検証計画について

Q：メルセデス・ベンツ、フルクスワーゲン、シュコダの車両でリモートコード実行（RCE）の実証が行なわれましたが、BMW についてはテスト車両のファームウェアが入手できなかったため、アドレス漏えいの検証にとどまったと報告されています。BMW、アウディ、あるいは BlueSDK を利用している自動車以外の製品について、今後さらなる検証を行なう計画はありますか？

A：アウディに関してはリモートコード実行まで完了しましたが、Linux AppArmor などの他のシステムモジュールのセキュリティメカニズムによって制限されています。つまり、Bluetooth ユーザー空間プロセス内では任意のコードを実行できますが、このプロセスにおけるシステム全体とのやり取りはサンドボックス化されています。
例えば、許可された僅かな場所を除いてファイルシステム上のファイルを操作することはできませんし、すべてのネットワークインターフェイスを介して自由に通信することもできず、ごく一部に限られます。

これは、現在の Bluetooth プロセスの権限を昇格させ、AppArmor サンドボックスから脱出するためのローカル権限昇格（LPE）研究の領域となります。現時点では、BMW 向けのエクスプロイト適合やアウディ向けの LPE 部分を継続する計画はありませんが、将来的に変更される可能性はあります。

4. 自動車サプライチェーンにおけるパッチ適用の課題

Q：講演によると、OpenSynergy 社は数カ月という自動車業界としては非常に迅速なスピードでパッチをリリースしたものの、複雑な自動車サプライチェーンの影響で大幅な遅延が生じ、一部の OEM（自動車メーカー）にパッチが行き届くまでに 1 年近く（2025 年 6 月まで）かかったとのことです。

セキュリティ研究者として、自動車業界における迅速なパッチ適用に向けた課題や解決策についてどのようにお考えですか？

A：課題について言えば、OEM が Tier-1（一次サプライヤー）から特定の ECU を入手し、そこに下位サプライヤーのさまざまなサードパーティソフトウェアが組み込まれている状況を想像してみて

ください。まさに BlueSDK で起きたことですが、サードパーティ製フレームワークのぜい弱性が OEM に報告された際、その OEM は実際の開発者が誰であるかを知らない可能性があり、修正要求がサプライチェーン全体を経由するためにかなりの時間を要します。また、パッチの適用準備が整った後でも、急いでそのまま適用できるわけではありません。通常、セキュリティパッチを適用する前には、さまざまな統合テスト、機能安全テスト、規制適合テストを実施する必要があります。さらに、ディーラーでしかアップデートを受けられない古い車両も存在するため、上流のアップデートにパッチが適用された後でも、多くの車両がぜい弱なまま残ってしまう可能性があります。

私の視点からは、アップデートシステム全体が非常に複雑で多くの関係者が関わっており、パッチ作成の責任や責務が曖昧になっているため、正確なボトルネックを特定するのは困難です。おそらく、車全体のソフトウェアイメージを再構築して再認証する理由がない場合、無線インターフェイスのモジュール式パッチに対して規制上の例外を設けるなど、このシステムに対する小規模で専用の改善を行なう余地はあるでしょう。追加の解決策としては、深刻なリモートぜい弱性の修復を優先することです。一連のテストプロセス全体を経由する代わりに、専用の方法で対応し、特定モジュール向けに反復的なパッチを展開することができます。

さらに、特定フェーズも改善されるべきです。ぜい弱なソフトウェアを担当しているサプライヤーを特定するだけで多くの時間がかかるため、私の考えでは、すべてのサプライヤー関連情報を迅速に保存・アクセスできるようにする必要があります。

5. 車内ネットワークにおける横展開（ラテラルムーブメント）の防止策

Q：インフォテインメント（IVI）システムがワンクリック RCE によって侵害された場合、攻撃者が車内の他の重要なシステムへ横展開（ラテラルムーブメント）を行なうという深刻なリスクを強調されました。今後、自動車メーカーは初期侵入後のような横展開を防ぐために、どのようなアーキテクチャ上のセキュリティ対策を実装すべきだとお考えですか？



エフドキモフ氏は、車載システムのセキュリティ対策として、機能制限や強制アクセス制御（MAC）による攻撃ベクターの縮小、ハードウェア防御の導入を提言。さらに、複雑な自動車サプライチェーンにおける迅速なパッチ適用の仕組みづくりと情報共有の改善を強く訴えた。

A：この問題に対する特効薬はありません。もしあれば、セキュリティ業界は現在のような形で存在していないでしょう。経験上、このようなリモート攻撃に対する最高のセキュリティ対策は、攻撃ベクター自体を狭めること（インターフェイスの無効化、機能のカットなど）、高度な検出・防止機能を持つ EDR フレームワークを使用すること、あるいは Linux の場合であれば SELinux や AppArmor などを活用することにあります。対象システムの各アプリケーションに特定のルールセットを適用する強制アクセス制御（MAC）アプローチを使用して、ファイルシステムへのアクセス、新規プロセスの作成、またはネットワーク通信を制限することで、横展開のための攻撃ベクターをかなり狭めることができます。

また、PAC（ポインター認証コード）や MTE（メモリータギング拡張）などのハードウェアセキュリティ対策を取り入れて、潜在的なぜい弱性のさらなる悪用を防ぐことも非常に大きなアドバンテージになります。当然、言及したこれらの手法

は単体でも優れていますが、すべてを組み合わせて使用することで、潜在的なぜい弱性の多くに対して非常に強力に保護された環境をシステム上に構築できます。明らかにこれらのツールがバグそのものを排除するわけではありませんが、エクスプロイトのプロセスを非常に高コストで時間のかかるものにすることができます。

6. Perfekt Blue 公表後の反響と自動車業界からの反応

Q：講演の中で、Perfekt Blue がさまざまなベンダーの数百万台の車両に影響を与える可能性があることを明らかにし、確認用の潜在的にぜい弱なデバイスのリストをご自身の Web サイトで公開されました。公表および講演後、自動車業界、サイバーセキュリティコミュニティ、または一般の人々からどのような反応やフィードバックがありましたか？

A：全体的な反応はおおむね好意的なものだったと言えます。このような汎用的な攻撃は組み込みデバイスの領域ではめったに見られない（少なくとも、そのような研究が公開されることはまれである）こと、さらにこの攻撃チェーンには Bluetooth プロトコルに非常に特化しつつもこれまで公表されていなかったいくつかのエクスプロイト手法が含まれていたため、この研究はセキュリティコミュニティでかなり大きな関心を集めました。

そのため、一部のセキュリティ関係者にとってこの情報は新しいものでした。自動車業界に関しては、多くのベンダーが責任ある開示（レスポンシブル・ディスクロージャー）の形式ですべてのぜい弱性を報告し、システムにパッチを適用するための十分な時間を提供したことに対して感謝してくれました。

しかし、時としてベンダーは顧客のパニックを防ぐために攻撃の実際の影響を低く見せかけようとすることがありますが、私の意見では、このようなケースを扱う最善の方法は、悪用された場合の潜在的な影響を正直に説明し、可能な限り迅速にパッチを適用することです。問題を沈黙させることが解決策になることは決してありません。

7. 今後の研究について

Q：今後の研究計画や、次に注目しようと考えている具体的なトピックや技術について教えていただけますか？

A：現在、将来の研究に関する具体的な計画はありませんが、無線技術（セルラー /WLAN/ Bluetooth）や OS カーネルに焦点を当てることは間違いありません。私の個人的な意見として、これらのターゲットが最も興味深いためです。全体として、自動車業界は非常に興味深い研究分野です。なぜなら、今日のほとんどの車両には、ブラウザー、カーネル、組み込み、RF など、セキュリティ研究のための多様な攻撃ベクターが含まれているからです。これらのモジュールの多くの実装は各ベンダーごとに（ある意味で）カスタムされているため、ターゲットを手に取るたびに常に新しい発見があり、継続的な学習が求められます。それこそがセキュリティ業界において最も重要なことであると私は考えています。

****

数百万台の車両に影響を及ぼした今回の問題は、複雑なサプライチェーンを抱える自動車業界に大きな教訓を残した。迅速なパッチ適用や情報共有の仕組みづくりは急務であり、コネクテッドカーの安全性を守るため、メーカーは従来の枠組みを超えた柔軟な脆弱性対応と強固な防御策の構築を急ぐべきだろう。

斉藤 健一（さいとう・けんいち）ハッカー専門誌「ハッカージャパン」の創刊から休刊まで（1998 年〜 2013 年）、編集長を務める。現在は、本誌の編集制作を担当する傍ら、サイバーセキュリティ国際会議「CODE BLUE」や、日本ハッカー協会が主催する「Hack Fes.」の運営にも携わっている。

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 2026.10

文＝日立システムズ

サイバー犯罪対策における AML（アンチマネーロンダリング）の重要性

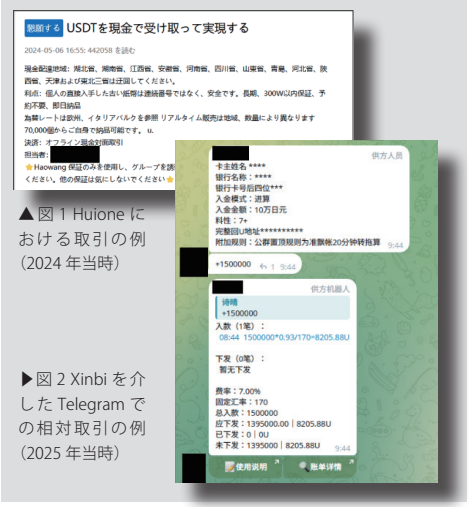
【概要】2026 年 9 月、米司法省はサイバー犯罪者向け市場の Xinbi Guarantee の摘発と暗号資産の凍結を発表した。Xinbi は Telegram 上で資金洗浄などに利用されていた。攻撃抑止には、侵入対策に加え AML 高度化と情報共有による資金遮断が不可欠である。

【本文】2026 年 9 月、米司法省はサイバー犯罪マーケットプレイスの Xinbi Guarantee（以下 Xinbi：シンビ）摘発と約 5,200 万ドルの暗号資産凍結を発表した。Xinbi は Telegram 上で詐欺サイト構築や資金洗浄、人材募集などに悪用され、米財務省 OFAC からも制裁対象に指定されている。

近年のサイバー犯罪は、専門サービスを組み合わせた分業型エコシステムへと変化している。不正アクセスやフィッシング、資金洗浄などを各グループが分業して 1 つの攻撃に関わっているが、こうした組織構造は突発的に生まれたものではない。

2025 年頃までは Huione Guarantee（フイワン）が東南アジアの代表的な犯罪マーケットプレイスとして認知されていた。同サービスではクレジットカード情報やフィッシングキット、詐欺サイト構築、マネーロンダリングなどの犯罪関連サービスが暗号資産を介して取り引きされていた（図 1）。

Xinbi もサービス提供者と買い手の取り引き（相対取引）を補助し、Telegram 上で仲介やエスクロー（取引保証）を提供していた（図 2）。今回の摘発は、犯罪マーケットが依然として大きな役割を持つことを示す。暗号資産は金融機関などの課題として捉えがちで、自組織とは距離のあるテーマに見えることもある。



▲図 1 Huione における取引の例（2024 年当時）

▶図 2 Xinbi を介した Telegram での相対取引の例（2025 年当時）

しかし、フィッシング、ビジネスメール詐欺（BEC）、ランサムウェア等による犯罪収益の多くは暗号資産を経由して移転・隠匿される。サイバーキルチェーンは攻撃の連鎖を断つ防御フレームワークであり、侵入や情報窃取だけでなく犯罪収益の換金段階まで対策することが攻撃者のビジネスモデルに圧力をかける。近年、国際的な法執行機関がマルウェアやボットネットのみならず、暗号資産ウォレット、ミキサーサービス、犯罪マーケットプレイスの摘発に注力している背景には、このような考え方がある。

サイバー犯罪エコシステムにおける暗号資産の重要性を踏まえると、セキュリティと暗号資産を別々に捉えるには限界がある。攻撃対策に加え、犯罪収益の移転・換金を防ぐ AML（Anti-Money Laundering）を強化することで収益化を困難にし、エコシステムの弱体化につながる。金融機関、暗号資産事業者、法執行機関、サイバーセキュリティコミュニティは、連携を強化し、サイバー犯罪抑止に向けた取り組みを進める必要がある。

［情報源］https://www.justice.gov/usao-dc/pr/scam-center-strike-force-conducts-seizures-chinese-run-illicit-scammer-marketplace-and

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 Windows イベントログ調査

3. Hayabusa によるイベントログ調査 攻撃痕跡調査編

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。「Hayabusa によるイベントログ調査」と題し、インシデント調査を行なううえで極めて重要な情報源である Windows イベントログ（以下、「イベントログ」）について、脅威ハンティングツール「Hayabusa」を用いた調査手順や観点をわかりやすく整理しています。本テーマは全 3 回で、イベントログの基礎から平常時ログの確認、攻撃痕跡調査の観点までを、ハンズオンを交えて段階的に解説します。

第 3 回となる今回は「攻撃痕跡調査編」として、サンプル攻撃イベントログを対象に、Hayabusa と Timeline Explorer を用いて、実際のインシデント対応に近い流れでイベントログ調査のハンズオンを行ないます。複数のイベントを時系列で整理し、成果物として簡易的なタイムラインを作成することを最終的な目的とします。

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ことを推奨します。

2. 準備

2.1 Windows 11 とツールの準備

前回までで利用した環境を引き続き利用します。環境の準備ができていない場合は、基本編の 2 章および 4.3 節と、平常時確認編の 2 章を参考に環境構築を行なってください。

本稿のハンズオンで使用するデータの特性上、ローカル環境では Microsoft Defender（Windows セキュリティ）の誤検知が発生する可能性があります。

データ自体に危険性はありませんが、誤検知の履歴を残したくない場合は、基本編の 2 章で記載のとおり Windows サンドボックス上での実行を推奨します。

なお、本稿のハンズオンで使用するツールの想定バージョンは以下のとおりです。

- Hayabusa：v3.9.0
- Timeline Explorer：v2026.5.0

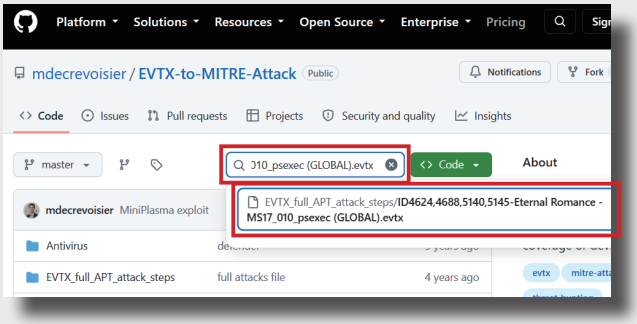
バージョンが異なると本稿の記載と動作が異なる可能性があるため、上記のバージョンを使用することを推奨します。

2.2 サンプル攻撃イベントログの準備

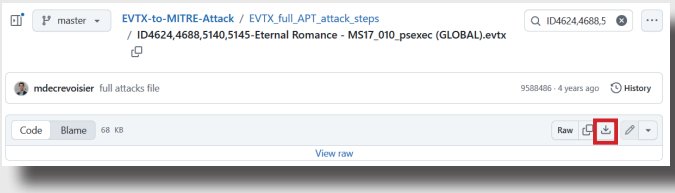
本稿のハンズオンでは、EVTX-to-MITRE-Attack という Windows イベントログのデータセットを利用します。
EVTX-to-MITRE-Attack は、攻撃や不審動作に関連するイベントログが戦術ごとにまとめられた evtx 形式のデータセットです。
下記の URL から、EVTX-to-MITRE-Attack の GitHub リポジトリにアクセスします。

https://github.com/mdecrevoisier/EVTX-to-MITRE-Attack

本稿では、イベントログ調査のハンズオンに用いるサンプル攻撃イベントログ（以下、「サンプルログ」）として、データセットの中の「ID4624,4688,5140,5145-Eternal Romance - MS17_010_psexec (GLOBAL).evtx」というファイルを利用します。
そのため、検索欄に上記のファイル名を入力して、検索結果からファイルのページに遷移します。



ファイルのページのダウンロードボタンをクリックし、ダウンロードした evtx 形式のファイルを、hayabusa.exe と同じフォルダーに配置します。



［注意］サンプルログには攻撃に関係するコマンドラインなどが含まれているため、サンプルログに記載されているコマンドを誤って実行しないよう注意してください。

ファイルのページのダウンロードボタンをクリックし、ダウンロードした evtx 形式のファイルを、hayabusa.exe と同じフォルダーに配置します。
Microsoft Defender がサンプルログを検知する可能性があります。これは、ログに含まれるコマンドを検知するものです。検知が発生した場合、サンプルログファイルや出力 CSV ファイルが削除される可能性があります。その場合は、Windows 設定アプリの「プライバシーとセキュリティ」→「Windows セキュリティ」→「ウイルスと脅威の防止」→「保護の履歴」から当該の検知の履歴（脅威が検疫されました）を選択し、「操作」ボタンからファイルを復元してください。

3. 攻撃が確認された端末のイベントログ調査

本章では、インシデント対応におけるイベントログ調査の流れに沿って、以下の手順でハンズオンを行ないます。

1. 調査目的の整理
2. 調査範囲の設定
3. ログ全体のスキャン
4. ログの消去
5. 認証・ログオン
6. 管理共有アクセス
7. プロセスの作成

なお 3.3 節以降では、各節のログ確認前に [問題] として確認する内容を記載し、その後に [解説] としてログ確認方法や観点を説明する構成となっています。([解答] は各節末に記載)
前回までで得られた知識などを活用し、まずはご自身で [問題] に取り組んでみてください。

3.1 調査目的の整理

インシデント対応では一般的に、調査を始める前に利用者やシステム管理者、監視担当者等にヒアリングを行ない、どういった目的で調査を行なうのかを整理します。
インシデント対応における調査目的の例として、以下のような目的が挙げられます。

①侵害有無の判断：本当にインシデントなのか
②侵入経路の特定：どこから、どのように侵入されたか
③攻撃手口の特定：どのようなプログラムが実行されたか
④影響範囲の特定：どの端末、システム、アカウントに影響が及んだか

上記の調査目的に沿ってさまざまな調査が行なわれますが、中でもイベントログは Windows OS において重要な情報源であり、調査の早い段階で確認されることが多いです。

イベントログからは、ログオン、特権の割り当て、共有フォルダーへのアクセスおよびプロセス作成などの記録を確認できます。これらの記録を時系列で整理することで、主に**①侵害有無の判断**、**②侵入経路の特定**、**③攻撃手口の特定**に必要な手掛かりを得ることができます。
一方で、対象端末のイベントログだけでは十分に判断できない場合も多く、特に**④影響範囲の特定**を行なう際などは、必要に応じて、他端末のイベントログ、EDR ログ、認証基盤およびネットワーク機器のログなどと突き合わせて調査を行ないます。

本稿では、**「ファイルサーバー fs03vuln.offsec.lan において不審な PowerShell 実行が検知された」**という前提のもとで調査を行ないます。
本事象に対する今回の調査目的を、以下のとおり設定します（今回は対象端末のイベントログのみの調査のため④影響範囲の特定は調査目的としません)。

①侵害有無の判断：
検知された PowerShell 実行の前後に、ログ消去や不審なログオンなど、攻撃を示す痕跡があるかの調査
②侵入経路の特定：
PowerShell 実行の起点となったログオンの送信元、アカウント、認証方式と、その後の共有アクセスの調査
③攻撃手口の特定：
ログオン後に作成されたプロセスの親子関係と、実行されたコマンドラインの調査

上記の内容について、イベントログから確認できる事実を抽出し、時系列で整理する方針で調査を進めていきます。

3.2 調査範囲の設定

イベントログをはじめとするログは膨大であり、すべてを詳細に調査するには時間がかかります。そのため、限られた時間で効率的に事実関係を把握するには、まずは事象発生前後のログを確認し、得られた情報をもとに優先順位をつけて調査範囲を広げていくことが重要です。
その際に、調査範囲を適切に設定するためには、関係者へのヒアリングを通してインシデントに関する情報を適切に収集する必要があります。

ヒアリングにおいて確認すべき情報の例を以下に示します。

分類	確認すべき情報の例
全体	インシデントの認知・発生日時、判断理由、アンチウイルスソフト検知情報、症状・被害、進行状況、暫定対策、調査目的・範囲、通信ログの有無
調査対象システム	ホスト・ネットワーク情報（ホスト名・IP アドレス・ドメイン名・構成図）、保守ベンダー、症状・被害、稼働状況、発覚後のシステム変更内容、端末用途・形状、OS、パッチ適用状況、主要アプリ、平時の利用方法、ネットワーク接続、アカウント数、ハードディスク情報（種別・容量・構成・ファイルシステム・暗号化）、USB 書き出し制限、ディスクコピー・端末持ち帰り可否、電源断・サービス停止可否

今回の調査では以下の調査範囲とし、サンプルログは調査範囲で抽出されたファイルであるという想定でハンズオンを進めます。

事項	調査範囲
対象ホスト	fs03vuln.offsec.lan
対象ログ	セキュリティログ
対象期間	2021-04-22 17:50 - 17:52 頃
対象ユーザー	OFFSEC\admmig、FS03VULN$

3.3 ログ全体のスキャン

イベントログ調査を行なうために、Hayabusa の csv-timeline コマンドを用いて、検知ルールを適用したタイムラインの CSV を出力します。
Hayabusa を起動するため、Windows キー +R で「ファイル名を指定して実行」ダイアログを開き、「cmd」と入力後 Ctrl+Shift+Enter を押して、管理者権限でコマンドプロンプトを起動します。
次のコマンドを実行し、Hayabusa のフォルダーに移動します。

```
cd C:¥Tools¥hayabusa
```

次のコマンドを実行します（Hayabusa v3.9.0 を想定）。

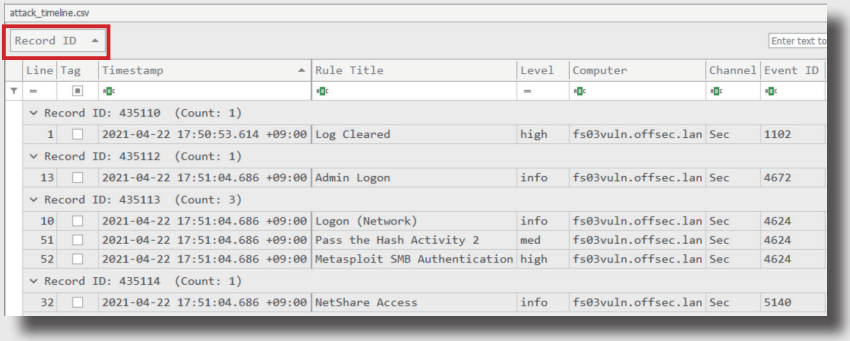
```
hayabusa.exe csv-timeline -f "ID4624,4688,5140,5145-Eternal Romance - MS17_010_psexec (GLOBAL).evtx" -o attack_timeline.csv
```

スキャンウィザード画面が表示されますが、検知ルールは「informational」以上のレベルが読み込まれる「5. All event and alert rules」を選択し、以降の選択はすべて「n」（no）を入力します。

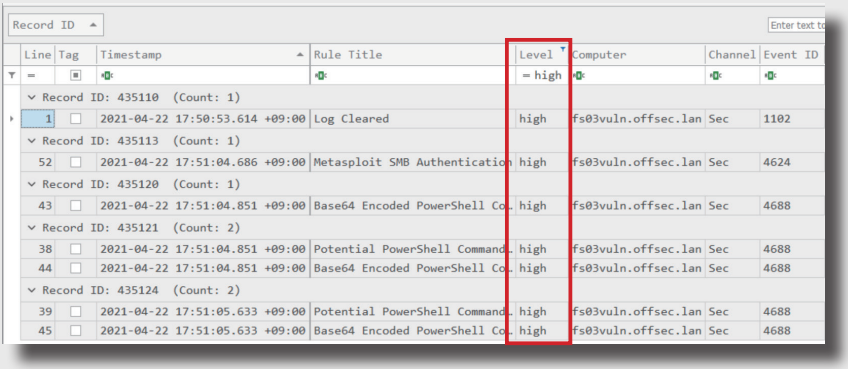
次に、タイムラインにおいて検知ルールのレベルが高いイベントを確認します。
レベルが高いという理由だけで異常と判断することはできませんが、調査の優先順位を決めるためには有用な情報です。
レベルが高いイベントを手掛かりとして関連イベントを追跡していくことで、効率的にイベントログ調査を行なうことができます。

Timeline Explorer を起動し、画面左上の「File」から「Open」をクリックして hayabusa.exe のフォルダーにある「attack_timeline.csv」を開きます（Timeline Explorer の操作方法は前回までの記事をご参照ください）。
Timestamp 列ヘッダーをクリックし、昇順に並べ替えます。

タイムラインを確認する際の注意点として、Hayabusa のタイムラインでは、1 つのイベントが複数の検知ルールに一致する場合があるということを考慮する必要があります。
同一イベントに対する複数の検知を見やすくするため、Record ID 列でグループ化を行ないます。



Level 列ヘッダー右上のフィルターアイコンを用いて、「high」でフィルターを適用することでレベルが高いイベントのみを確認します。



Line	Tag	Timestamp	Rule Title	Level	Computer	Channel	Event ID
=				= high			
Record ID: 435110 (Count: 1)							
1		2021-04-22 17:50:53.614 +09:00	Log Cleared	high	fs03vuln.offsec.lan	Sec	1102
Record ID: 435113 (Count: 1)							
52		2021-04-22 17:51:04.686 +09:00	Metasploit SMB Authentication	high	fs03vuln.offsec.lan	Sec	4624
Record ID: 435120 (Count: 1)							
43		2021-04-22 17:51:04.851 +09:00	Base64 Encoded PowerShell Co…	high	fs03vuln.offsec.lan	Sec	4688
Record ID: 435121 (Count: 2)							
38		2021-04-22 17:51:04.851 +09:00	Potential PowerShell Command…	high	fs03vuln.offsec.lan	Sec	4688
44		2021-04-22 17:51:04.851 +09:00	Base64 Encoded PowerShell Co…	high	fs03vuln.offsec.lan	Sec	4688
Record ID: 435124 (Count: 2)							
39		2021-04-22 17:51:05.633 +09:00	Potential PowerShell Command…	high	fs03vuln.offsec.lan	Sec	4688
45		2021-04-22 17:51:05.633 +09:00	Base64 Encoded PowerShell Co…	high	fs03vuln.offsec.lan	Sec	4688

[問題]

（1）セキュリティイベントログの消去が記録された時刻はいつですか。
（2）不審な認証の手掛かりとなるイベントの検知ルールおよび時刻は何ですか。
（3）Base64 でエンコードされた PowerShell コマンドが最初に記録された時刻はいつですか。

[解説]

確認できたレベルが高いイベントの時刻および検知ルールを以下に示します。説明は、各検知ルールの定義ファイルに記載されています（GitHub リポジトリから検索）。

Timestamp（時刻）	Rule Title（検知ルール）	Description（説明）
17:50:53.614	Log Cleared	ユーザーがセキュリティイベントログを消去しました
17:51:04.686	Metasploit SMB Authentication	Metasploit による SMB 認証に似た特徴を持つ NTLM ネットワーク認証を検知し、アラートを生成します
17:51:04.851 17:51:04.851 17:51:05.633	Base64 Encoded PowerShell Command Detected	Base64 でエンコードされた文字列をデコードするために使用される FromBase64String 関数が、コマンドライン内で使用されたことを検知します
17:51:04.851 17:51:05.633	Potential PowerShell Command Line Obfuscation	難読化に使用される可能性のある特殊文字が、PowerShell のコマンドライン内で繰り返し使用されたことを検知します

上記の検知内容から、一旦の推測として次のような事象が起きた可能性が考えられます。

・ログ消去後、外部端末から SMB 経由で対象ホストへアクセスし、難読化された PowerShell を実行した。

以降の調査では、検知された各イベントに関連するイベントを調査しながら、タイムライン（以下、「調査結果タイムライン」）を作成し、推測が正しいかを検証していきます。

[解答]
（1）17:50:53.614
（2）検知ルール：Metasploit SMB Authentication、時刻：17:51:04.686
（3）17:51:04.851

3.4 ログの消去

攻撃者は、侵入後の活動を隠すためにイベントログを消去する場合があります。
ログが消去されている場合、消去以前の記録は残っていないため、その期間の事象はイベントログだけでは確認できません。そのため、調査の早い段階でログ消去の痕跡の有無を確認し、イベントログから調査できる期間を把握しておくことが重要です。

主なログ消去イベントとして、セキュリティイベントログのイベント ID 1102（セキュリティ監査ログの消去）や、システムログのイベント ID 104（イベントログの消去）がありますが、今回はセキュリティイベントログのため、Event ID 列「1102」でフィルターを適用します。その際、前節で適用した Level のフィルターは解除します。

[問題]
（1）イベントログ消去を示す痕跡はありますか。イベントログが消去されている場合には、どのユーザーのログオンセッションでログが消去されましたか。
（2）ログ消去イベントに記録された LID は何ですか。

[解説]
以下のイベントが確認できます。
以降の説明では、以下のイベントを「イベント A」とします。

項目	値
Timestamp	2021-04-22 17:50:53.614 +09:00
Event ID	1102
Rule Title	Log Cleared
Level	high
SubjectDomainName	OFFSEC
SubjectLogonId	0x7424d
SubjectUserName	admmig

イベント A から、17 時 50 分 53 秒に OFFSEC\admmig ユーザーのログオンセッションでセキュリティ監査ログが消去されたことが確認できます。
ログオンセッションとは、ユーザーがシステムにログオンしてからログオフするまでの一連の状態を管理する仕組みです。

イベント A のみでは、正規のユーザーによる操作であったのか、攻撃者による痕跡隠蔽であったのかは断定できません。しかし、調査対象期間においてログ消去のイベントが発生している場合は、攻撃活動の一部である可能性も考慮し、調査の優先度が高いイベントとして調査結果タイムラインに記載しておきましょう。

[解答]
（1）痕跡の有無：有、ユーザー：OFFSEC\admmig
（2）0x7424d

3.5 認証・ログオン

次に、対象ホストに対する不審な認証・ログオンの有無を確認します。また、記録されているイベントを確認し、どこから、どのアカウントでログオンされたのかを調査します。
今回は、3.3 節で得られた検知ルールのレベルが高いイベントのうち、イベント ID 4624（ログオン成功）でフィルターを適用し、不審なログオンイベントを確認します。
前節で適用した Event ID 列「1102」のフィルターを解除し、「4624」でフィルターを適用します。また、Level 列「high」でフィルターを適用します。

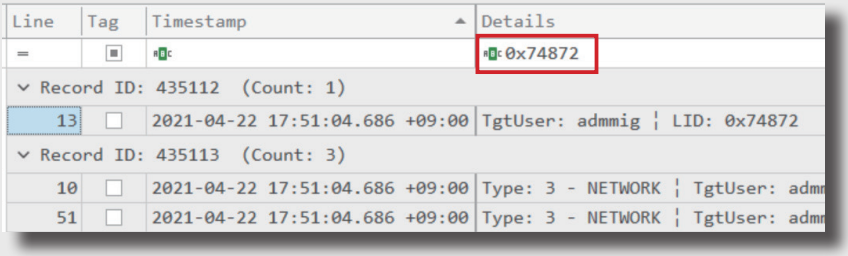
[問題]
（1）通常利用時に想定されない認証方式等の、不審なログオンの痕跡はありますか。また、痕跡にはどの IP アドレスから、どのユーザーを対象としたログオンが成功したと記録されていますか。
（2）（1）のログオンセッションで記録されている、他の 2 つのイベントのイベント ID は何ですか。

[解説]
以下のイベントが確認できます。
以降の説明では、以下のイベントを「イベント B」とします。

項目	値
Timestamp	2021-04-22 17:51:04.686 +09:00
Event ID	4624
Rule Title	Metasploit SMB Authentication
Level	high
TgtUser	Admmig
TargetDomainName	OFFSEC
Logon Type	3 – NETWORK
SrcIP	10.23.123.11
AuthenticationPackageName	NTLM
LID	0x74872

イベント B から、17 時 51 分 04 秒に 10.23.123.11 から、OFFSEC\admmig を対象とした NTLM 認証によるネットワークログオンが成功したことが確認できます。

なお、LID はログオンセッションを識別する ID です。イベント ID 4624 では、ログオンしたアカウントのログオン ID は TargetLogonId に記録されます。関連するイベント ID 4672 や 5140 では、SubjectLogonId と照合することで、同じログオンセッションの活動を追跡できます。
イベント B のログオンセッションにおける関連イベントを確認するために、同じ LID のイベントを確認してみましょう。
現在適用している Event ID 列と Level 列のフィルターを解除し、Details 列ヘッダーの下の入力欄に「0x74872」と入力します。



LID 0x74872 のログオンセッションのイベントとして、イベント B と同時刻に以下の 2 つのイベントが確認できます。

Timestamp	Event ID	Rule Title
17:51:04.686	4672	Admin Logon
17:51:04.686	5140	NetShare Access

1 行目のイベント ID 4672（特権割り当て）のイベントは、イベント B と同じログオンセッションで同時刻に発生しています。よって、10.23.123.11 からのネットワークログオンに特権が割り当てられたと考えられます。
2 行目のイベントは次節で確認します。

[解答]
（1）痕跡の有無：有、IP アドレス：10.23.123.11、対象ユーザー：OFFSEC\admmig
（2）4672、5140

3.6 管理共有アクセス

前節で確認した不審なログオンのイベント B における送信元 IP アドレス（SrcIP）10.23.123.11 が、攻撃に関係している可能性があると仮定して調査を行ないます。
また、不審なログオンのイベント B は SMB 認証の可能性があると検知されています。SMB 認証経由の侵入では、共有フォルダーの一種である管理共有（IPC$、ADMIN$、C$ など）を悪用してリモート操作を実行する手法が広く用いられるため、本節では共有フォルダーへのアクセスについて確認します。

前節で適用したDetails列のフィルターを解除し、Details列ヘッダーの下の入力欄に「10.23.123.11」と入力します。その後、Event ID列「5140」（ネットワーク共有オブジェクトアクセス）および「5145」（共有オブジェクト詳細アクセス）でフィルターを適用します。

[問題]

（1）10.23.123.11 からアクセスされた共有フォルダー（ShareName）は何ですか。

（2）（1）で確認したイベントのうち、イベント ID 5145 のイベントには、どのファイルへのアクセスが記録されていますか。

[解説]

以下のイベントが確認できます（すべてのイベントの SrcIP は 10.23.123.11 です）。

Timestamp	Event ID	Rule Title	ShareName（共有名）	LID	Path（アクセス先パス）
17:51:04.686	5140	NetShare Access	IPC$	0x74872	-
17:51:04.780	5140	NetShare Access	ADMIN$	0x3e7	-
17:51:04.780	5145	NetShare File Access	ADMIN$	0x3e7	System32¥WindowsPowerShell¥v1.0¥powershell.exe
17:51:04.796	5140	NetShare Access	IPC$	0x3e7	-

管理共有とは、Windows が既定で作成する非表示の共有フォルダーであり、ShareName はアクセスされた共有リソースの共有名を示します。IPC$ は、SMB セッションの確立や名前付きパイプを利用したリモート管理通信の際に利用され、ADMIN$ は接続先端末のシステムフォルダー（通常は Windows フォルダー）へアクセスするための管理共有として利用されます。
また、3 行目のイベント ID 5145 のイベントの Path より、powershell.exe へのアクセスがあったことが確認できます。一方で、本イベントはファイルアクセスを示すものであり、PowerShell の実行そのものを示すものではありません。

1 行目の IPC$ アクセスは、前節の不審なログオン（イベント B）と同一のログオンセッション（LID 0x74872）で発生しています。しかし、2 行目以降のアクセスは LID が 0x3e7 に変化しており、SID も S-1-5-18（SYSTEM）となっています。したがって、1 行目のアクセスと 2 行目以降のアクセスの関連性については、本イベント単独では断定できません。一方で、同じ SrcIP からのほぼ同時刻のアクセスという点で、2 行目以降も攻撃との関連性が疑われます。そのため、調査結果タイムラインに記載しておき、次節で追加調査を行ないます。

3 行目のイベント詳細を以下に示します。
以降の説明では、以下のイベントを「イベント C」とします。

項目	値
Timestamp	2021-04-22 17:51:04.780 +09:00
Event ID	5145
Rule Title	NetShare File Access
Level	info
ShareName	ADMIN$
Path	System32¥WindowsPowerShell¥v1.0¥powershell.exe
SrcIP	10.23.123.11
LID	0x3e7

[解答]
（1）IPC$、ADMIN$
（2）System32¥WindowsPowerShell¥v1.0¥powershell.exe

3.7 プロセスの作成

前節では、不審なログオンのイベント B の SrcIP 10.23.123.11 に関して調査したところ、10.23.123.11 から ADMIN$ を通じて PowerShell 実行ファイルへアクセスしたとみられるイベント C が確認できました。
本節では、イベント C 後の PowerShell 実行の有無を含めたプロセス作成の流れを、LID および PID を用いて確認していきます。

まず、イベント C との関連が疑われるプロセスを確認するため、イベント C と同じログオンセッション（LID 0x3e7）で、イベント C の直後に記録されたイベント ID 4688（プロセス作成）のログを確認します。

前節で適用した Event ID 列と Details 列のフィルターを解除し、Details 列ヘッダーの下の入力欄に「0x3e7」と入力します。その後、Event ID 列「4688」でフィルターを適用します。

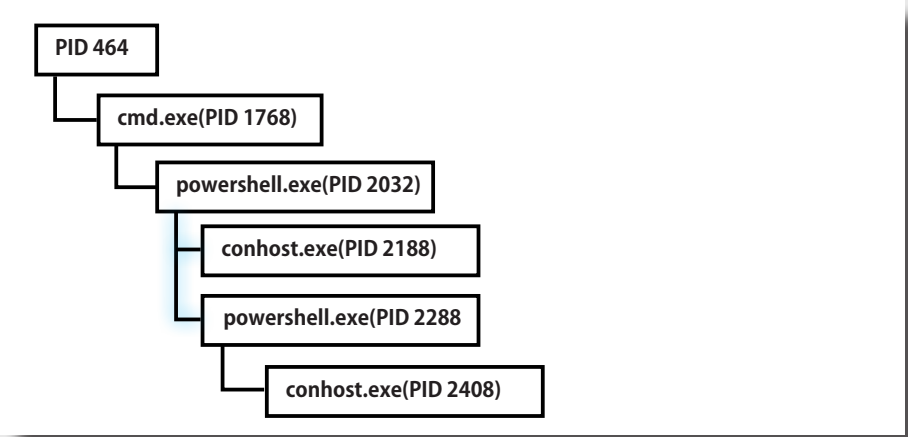
[問題]
（1）イベント C の直後に作成されたプロセスの PID は何ですか。
（2）確認できたイベントからプロセスツリー（プロセスの親子関係を木構造で表したもの）を作成してください。

[解説]
以下のイベントが確認できます（すべてのイベントのイベント ID は 4688、LID は 0x3e7 です）。

Timestamp	PID （プロセス ID）	ProcessId （親プロセス ID）	Proc （プロセスパス）
17:51:04.851	1768	464	C:¥Windows¥System32¥cmd.exe
17:51:04.851	2032	1768	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe
17:51:04.867	2188	2032	C:¥Windows¥System32¥conhost.exe
17:51:05.633	2288	2032	C:¥Windows¥SysWOW64¥WindowsPowerShell¥v1.0¥powershell.exe
17:51:05.633	2408	2288	C:¥Windows¥System32¥conhost.exe

PID は、各プロセスを識別する ID であり、イベント ID 4688（プロセス作成）のログに記録されます。
イベント ID 4688 のログには、そのプロセスの PID だけでなく、親プロセス（作成の元となったプロセス）の PID やコマンドラインも記録されているため、プロセス同士の関係や実行内容を確認する際には、イベント ID 4688 のログで PID を追跡していきます。（親プロセスの PID は Extra Field Info 列の ProcessId に記録）
ただし、イベント ID 4688 の記録は標準設定では無効となっているため、記録するにはあらかじめ設定しておく必要があることにご注意ください。

確認結果より、プロセスの親子関係は以下のようなプロセスツリーで整理できます。



この結果から、イベントCでPowerShell実行ファイルへアクセス後、同じログオンセッションでcmd.exeおよびPowerShellの実行があったことが確認できます。（conhost.exeはPowerShell実行に付随して作成されるプロセスです。）

次に、PowerShellで実行されたコマンドラインを確認します。
コマンドラインは、Details列のCmdlineに記録されています。
PID 1768のcmd.exeのプロセスのコマンドライン冒頭は次のとおりです。

```
C:¥Windows¥system32¥cmd.exe /b /c start /b /min powershell.exe …
```

/cは指定されたコマンドを実行した後にcmd.exeを終了する指定、start /bは新しいウィンドウを開かずにプログラムを開始する指定、/minはウィンドウを最小化して起動する指定です。つまり、cmd.exeはPowerShellを目立たない形で起動するために使用されています。
起動されたPID 2032、2288のPowerShellのプロセスに関して、コマンドラインを確認し読み解くと、以下のような動作が行なわれたことがわかります。

・PID 2032：
64ビット環境かどうかを判定し、32ビット版のPowerShellをPID 2288として起動した。
・PID 2288：
Base64エンコードおよびgzip圧縮によって難読化されたデータをメモリー上でデコード・展開した。

本稿の範囲外のため、本ハンズオンでは難読化された処理内容の解析（デコード）までは扱いませんが、不審なログオンのイベントBとの関係を考慮すると、攻撃者によって侵害後に何らかの処理が実行された可能性が高いと判断できます。
そのため、これらのプロセス作成のイベントに関しても調査結果タイムラインに記載しておきましょう。

[解答]
（1）1768
（2）プロセスツリーは [解説] で記載

4. タイムラインの作成

対象端末で発生した事象を時系列で把握するために、3章で確認したイベントを調査結果タイムラインとしてまとめます。

○調査結果タイムライン
対象ホスト：fs03vuln.offsec.lan
対象日：2021年4月22日
タイムゾーン：JST（+09:00）

Timestamp	Event ID	内容	関連付けの根拠
17:50:53.614	1102	OFFSEC\admmig ユーザーのログオンセッションでセキュリティ監査ログが消去された（イベント A）	User：admmig LID：0x7424d
17:51:04.686	4624	10.23.123.11 から、OFFSEC\admmig を対象としたネットワークログオンが成功した（イベント B）	Record ID：435113 LID：0x74872
17:51:04.686	4672	イベント B のログオンに特権が割り当てられた	LID：0x74872
17:51:04.686	5140	10.23.123.11 から管理共有 IPC$ へアクセスした	IP：10.23.123.11 LID：0x74872
17:51:04.780	5140	10.23.123.11 から管理共有 ADMIN$ へアクセスした	IP：10.23.123.11 LID：0x3e7
17:51:04.780	5145	10.23.123.11 から PowerShell 実行ファイルへアクセスした（イベント C）	IP：10.23.123.11 LID：0x3e7
17:51:04.851	4688	イベント C のログオンセッションで cmd.exe が起動し、PID 2032 として PowerShell を最小化して起動した	LID：0x3e7 PID：1768
17:51:04.851	4688	64 ビット環境かどうかを判定し、32 ビット版の PowerShell を PID 2288 として起動した	PID：2032 親 PID：1768
17:51:05.633	4688	Base64 エンコードおよび gzip 圧縮によって難読化されたデータをメモリー上でデコード・展開した	PID：2288 親 PID：2032

また、3.1 節で設定した調査目的に対する調査結果を、以下に整理します。

①侵害有無の判断：
不審なネットワークログオン、管理共有へのアクセスおよび難読化された PowerShell の実行が一連の事象として確認されました。これらのイベントが短時間に連続していることから、対象ホストが侵害された可能性が高いと考えられます。

②侵入経路の特定：
送信元 IP アドレス 10.23.123.11 から、OFFSEC\admmig を対象とした NTLM 認証によるネットワークログオンが成功し、その後、SMB 経由で管理共有 IPC$ および ADMIN$ にアクセスしたことが確認されました。

③攻撃手口の特定：
管理共有へのアクセス後、cmd.exe から PowerShell が起動され、Base64 エンコードおよび gzip 圧縮によって難読化されたデータがメモリー上でデコード・展開されたことが確認されました。

調査結果より、今回の事象は、10.23.123.11 から管理者アカウントを使用して SMB 経由で対象ホストの管理共有 IPC$、ADMIN$ へアクセスし、難読化された PowerShell を実行したという事象であると考えられます。

以上の調査により、3.1 節で設定した調査目的に沿って、イベントログから確認できる情報を整理することができました。
しかし、今回の調査では確認できなかった情報も多くあります。
確認できなかった情報および行なうべき追加調査の例を以下に示します。

・ログ消去以前のログ
→ EDR、認証基盤、ファイアウォール等、対象ホスト以外に保存されたログを確認する。

・10.23.123.11 の詳細
→資産管理情報や DHCP ログから送信元端末を特定し、当該端末の EDR 記録を確認する。

・PowerShell で展開された処理の詳細
→難読化された処理内容について、安全な解析環境でのデコードを検討する。

・外部サーバーとの通信の有無
→ファイアウォール、プロキシ、DNS、EDR 等の通信ログを確認し、対象ホストから外部 IP アドレスやドメインへの通信が発生していないか調査する。

・情報が実際に持ち出されたか
→通信ログを確認し、外部へのファイル転送や通常より大きなデータ送信が発生していないか調査する。

5. おわりに

今回はここまでとなります。
「Hayabusa によるイベントログ調査 攻撃痕跡調査編」では、サンプル攻撃イベントログを対象に、Hayabusa と Timeline Explorer を用いて攻撃の痕跡を調査するハンズオンを実施しました。
ハンズオンでは、Hayabusa による検知ルールのレベルが高いイベントを調査の手掛かりとして、時刻、IP アドレス、LID、PID 等の情報から認証・ログオン、管理共有へのアクセス、プロセス作成といった関連イベントを追跡した後、タイムラインとして確認できた事象を整理しました。

イベントログ調査では、個々のイベントや検知ルール名だけで事象を判断するのではなく、複数のイベントを時系列で整理し、ログから確認できる事実と推測を区別することが重要です。また、イベントログだけでは確認できない情報もあるため、必要に応じて EDR やネットワーク機器のログ等の追加調査によって推測を検証する必要があります。

本稿で紹介した Hayabusa を用いた調査の流れや観点を、今後のインシデント調査やイベントログ調査において活用していただければ幸いです。

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

※本カタログに記載されている会社名、製品名は、それぞれの会社の登録商標または商標です。
※本カタログに記載されている内容、仕様については、予告なく変更する場合があります。
※本製品を輸出する場合には、外国為替および外国貿易法ならびに、米国の輸出管理関連法規などの規制を御確認の上、必要な手続きをお取りください。なお、ご不明な場合は、当社営業にお問い合わせください。

Printed in Japan