グローバルセキュリティ動向四半期レポート

2024 年度　第 2 四半期

目次

1. エグゼグティブサマリー

本レポートは、NTTDATA-CERTが期間中に収集したサイバーセキュリティ関連情報に基づき、その四半期におけるグローバル動向を独自の観点で調査・分析したものです。

注目トピック『内部不正による情報漏洩：特徴と求められる対策』

内部不正による情報漏洩は、企業にとって深刻なセキュリティリスクです。特徴として、通常業務と見分けにくい行為や保有している権限や能力の悪用が多く、退職時や退職目前に行われることが多いです。対策としては、アクセス制限や特権管理、ログ監視などの技術的対策に加え、社員教育やルールの徹底が重要です。

内部不正の防止には、技術と教育の両面からのアプローチが求められます。企業はこれらの対策を統合し、持続的な防御策を構築することが重要です。また、内部不正の特性に応じた対策として、隠蔽タイプには「機会」・「動機」・「正当化」の３側面に基づく不正のトライアングルへの干渉、大胆タイプにはアクセス制限や即時停止システムの導入が推奨されます。内部不正対策ソリューションとして、DLPや特権管理システム、UEBAの活用が効果的です。

脆弱性『急増するパブリックAPIの脆弱性対策』

パブリックAPIの利用拡大に伴い、脆弱性が増加し、サイバー攻撃のリスクが高まっています。特に認証や認可の設定不備が多く、Broken Object Level Authorization（BOLA）などの脆弱性が悪用されています。

対策として、セキュアバイデザインのアプローチを採用し、開発初期段階からセキュリティを組み込むことが重要です。具体的には、多要素認証やOAuth 2.0の実装、ランタイム保護、セキュリティテスト、セキュリティ・ポスチャー管理、APIのカタログ化、APIガバナンスの強化などを推奨します。また、セキュアバイデザインでは解決できない問題への対策として、APIディスカバリーや自動化されたセキュリティツールの活用、セキュリティ教育の強化も必要です。

脆弱性『SSVCを用いた脆弱性トリアージ手法の改善』

SSVC（Stakeholder-Specific Vulnerability Categorization）は、脆弱性対応の優先度を評価するフレームワークで、脆弱性の深刻度を評価する指標であるCVSS（Common Vulnerability Scoring System）の課題を解決します。SSVCは決定木を用いて脆弱性の影響を評価し、対応の優先度を「Immediate」「Out-of-cycle」「Scheduled」「Defer」の4段階で示します。これにより、脆弱性対応の判断が視覚的かつ論理的に行えます。

AI技術の進化に伴って脆弱性を狙ったサイバー攻撃が短時間で発生するため、迅速に脆弱性対応しなければなりません。脆弱性対応の事前準備と工夫がポイントです。具体的には、SSVCの4つの決定点のうちExposureやMission Impactに関係するサーバやネットワーク機器、ソフトウェアをグループ化して評価値を用意しておいてから、SSVCを使って脆弱性対応の優先度を決定します。少人数でも効果的な脆弱性対応が可能となり、組織のセキュリティ対応力を向上することができます。

2. 注目トピック『内部不正による情報漏洩：特徴と求められる対策』

NTTデータ SL事業本部 セキュリティ＆ネットワーク事業部　栗本 重彰

内部不正による情報漏洩は、独立行政法人・情報処理推進機構（IPA）が毎年発表しているセキュリティ10大脅威（組織）でも、IPAが発表を始めた2016年からは毎年10位以内に入り、2024年は3位でした。内部不正による情報漏洩は、企業が常に注視しなければならない深刻なセキュリティリスクです［1］。内部不正による情報漏洩が発生してしまうと、企業は信頼を損ない経済的な損失につながるだけでなく、法的な問題に発展するおそれもあります。

本記事では、年々巧妙化を続ける内部不正による情報漏洩に関して、実際によくある事例やその特徴を中心に記述し、予防的対策や持続的な防御策として採用することが多い対策も解説します。

2.1. 内部不正による情報漏洩事例

内部不正による情報漏洩は、個人が起こす場合から複数人が関与して起こす場合、悪意の強さの違いなど、さまざまなケースがあります。以下に主な内部不正のパターンを例示します［2］［3］。

① 退職予定の社員が機密情報／個人情報を不正に持ち出す
② 委託社員が委託元企業の機密情報／個人情報を不正に持ち出す
③ 産業スパイが機密情報／個人情報を不正に持ち出す
④ 複数の社員が結託して機密情報／個人情報を不正に持ち出す
⑤ 期限までに資料作成が終わらないので土日に自宅で作業するために、機密情報を持ち出す
⑥ 会社に恨みがある社員が情報を破壊する、偽情報を流布するなどの嫌がらせをおこなう

内部不正の主体や目的はさまざまです。これらの内部不正に少なからず共通して言える特徴は、「通常業務と見分けにくい行為」や「保有している権限や能力を使って通常業務の一環で実施できる行為」である点です。悪意や故意で行なった内部不正と通常の業務行為を判別しづらいため、内部不正は周囲の目を欺き、監視をくぐり抜けやすくなります。次の章では、その特徴を具体的に説明します。

2.2. 近年における内部不正事案の特徴

通常業務と見分けにくい内部不正は、いくつもの巧妙な手口を使います。内部不正の事例を整理して、以下のような巧妙な手口の例を表 2-1へまとめました。

① 不正に持ち出したい機密情報／個人情報を、通常業務で扱う情報・行為の中に紛れ込ませる
② 一度に大量の情報を持ち出すのではなく、少量の情報を複数回に分け継続的に持ち出す
③ 内部不正を検知、制御するシステムやルールの抜け穴を使った手段で実施する

表 2-1:通常業務と見分けが付きにくい内部不正行為の例

①	【行為概要】 持出したい情報を、通常業務で取り扱うファイルや行為の中に不正に紛れ込ませる	【通常業務と見分けが付きにくい理由】 周囲からは、通常業務の同等もしくはその一環に見えてしまう
	【具体例】 ✓ 外部送付する打合せ文書内に、関係のない機密情報を含むページを挿入する ✓ 機密ファイルを、大量の一般情報のファイルと一緒に圧縮してzipファイルとして持出す	
②	【行為概要】 少量の情報を複数回に分けて、継続的に持出す	【通常業務と見分けが付きにくい理由】 一度に大量の情報を持ち出さないため、目立たない
	【具体例】 ✓ 機密文書を、毎日数ページずつ印刷する ✓ 毎週社外展開する議事録内に、機密情報を分割し手数行ずつ挿入し、持出す ※上記①のポイントを組合せた事例	
③	【行為概要】 システム／ルールでは禁止や検知がされない抜け穴をついた手段で実施する	【通常業務と見分けが付きにくい理由】 即時に検知がされないため、異常行為として判断されず、通常行為の中に埋もれてしまう
	【具体例】 ✓ システムで検知するキーワードのみ、伏字や他の言葉に変更する ✓ システムで検知しないファイル／拡張子に変更する	

ある仮の環境を設定し具体的な行為をシミュレーションして、上記のポイントを解説します。

2.2.1. 仮の組織環境における不正持ち出しシミュレーション

実際に発生した内部不正事例や知見をもとに、内部不正事案の特徴を踏まえた不正持ち出しのシナリオを記述します。まず、以下に仮の組織環境を設定します(図 2-1参照)。

- A社（委託元企業）が、新製品開発業務（機密情報）のプロジェクトをB社（委託先企業）へ依頼。
- B社の社員X氏がA社のプロジェクトへ参画。A社の委託社員用業務アカウントを持ち、一部の機密情報にアクセスして業務を行っていた。
- A社とB社は、同じコミュニケーションツール（例：Microsoft Teams等）を利用しており、各社のテナントを連携して情報共有を実施。
- ただし、A社から社外に機密情報を持ち出そうとすると、A社が導入しているDLP（Data Loss Prevention:情報が持ち出されるのを防ぐセキュリティツール）のルールで検知／遮断する。そのために、基本的にA社－B社間でやり取りを行う情報には機密情報を含まない。

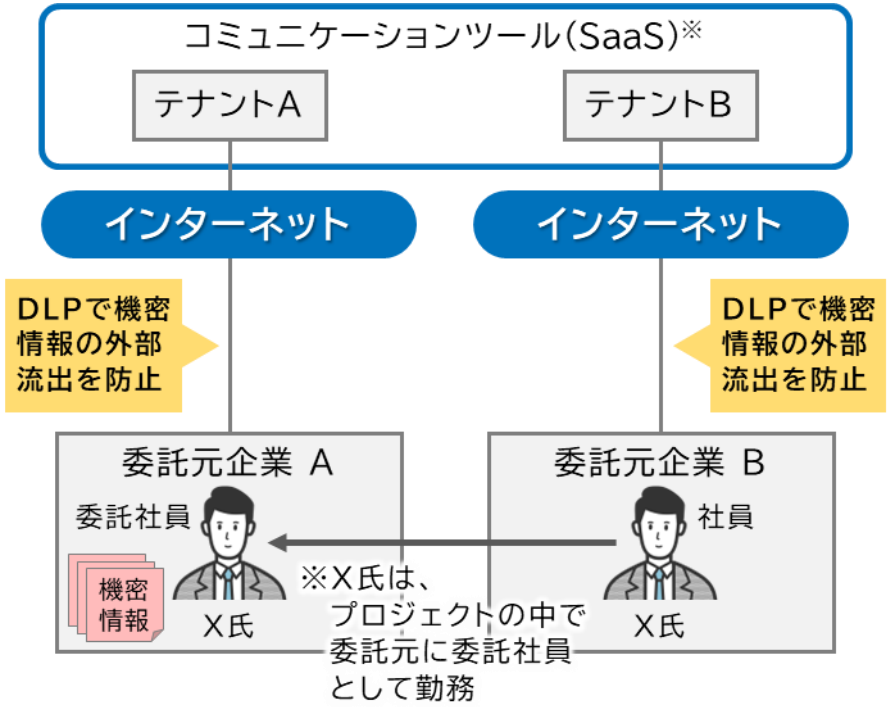


図 2-1:仮の組織環境

2.2.2. 不正持ち出しのヒント

この環境にて、X氏が、委託元企業A社の機密情報の不正持ち出しをすることを試みます。

- X氏は、通常業務の中で、プロジェクトに関わる情報をA社環境からB社環境へコミュニケーションツールを使って連携している。複数ファイルをZip形式でまとめて連携。
- ある時、X氏がA社環境からB社環境へ連携しようとしたZipファイル内に機密情報が意図せず含まれており、DLPがキーワードを検知して、遮断した。
- X氏は、この検知からDLPの検知／遮断ルール、DLPが検知するキーワードを知ることができた。

このケースのように、通常業務の中でDLPのルールの一部を知ることができます。

2.2.3. 不正持ち出しの実行

X氏はB社から別のC社に転職することになり、A社のプロジェクト3か月後に離任することになった。そこでX氏は、転職先のC社で情報を活用するために、プロジェクト離任及び転職までの３か月の間に、A社の機密情報を持ち出そうと試みます。内部不正による情報持ち出しは、退職時／退職直前に行われることが多いです [4]。X氏は、委託先A社の環境から以下の手段を用いて情報持ち出しを実施します（図 2-2参照）。

- X氏は、持ち出したい文書の中から、DLPが検知するキーワードをDLPが検知しない言葉へ変更した
- A社環境からB社環境へ連携するZipファイル群の中へ、上記の細工した機密情報を含むファイルを紛れ込ませて持ち出した。
- 複数の機密情報ファイルをZipファイルへ入れて一度に持ち出すのではなく、Zipファイルへ少しずつ入れて持ち出した。
- 3か月後のプロジェクト離任まで継続的に機密情報ファイルの持ち出しを実施した。
- 機密情報ファイルをA社からB社にまで持ち込んだ後、B社の環境でファイルを少しずつ印刷し、機密情報を私物化した。

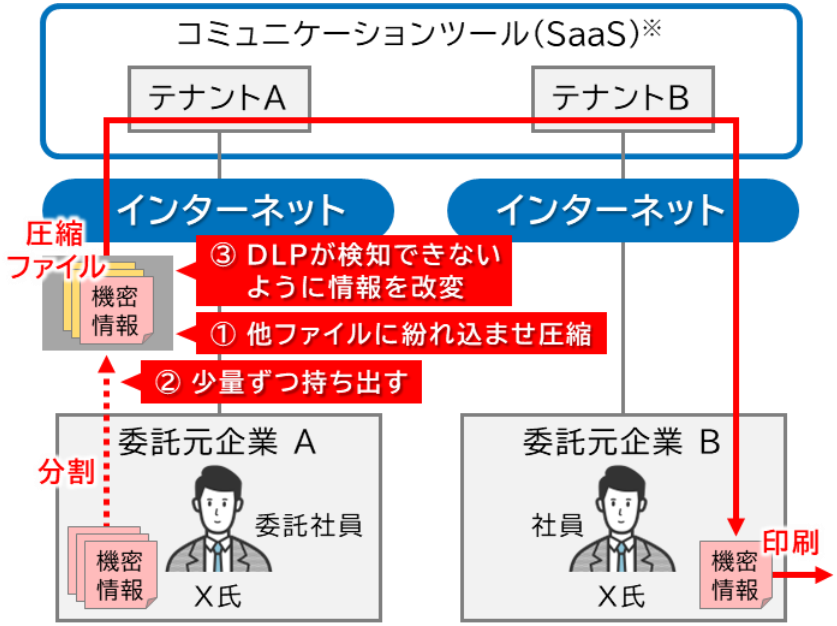


図 2-2:不正持ち出し手段

この不正持ち出しのシナリオは、前述の①通常業務で取り扱うファイルに紛れ込ませる、②少量の情報を複数回に分けて持ち出す、③検知しないように情報を改変してシステム／ルールの抜け穴をつく、といった通常業務と見分けが付きにくい不正アクセスの方法を使って作成しています。

また、X氏は、機密情報ファイルを自社のB社環境へ持ち出した後、紙へ印刷して自宅へ持ち出します。近年、デジタル化やリモートワークを背景に、資料の紙への印刷が減っているため、印刷処理ログの監視や機密情報の印刷制御のセキュリティ対策を行うことが少なくなってきています。この点も抜け穴として悪用しやすくなっています。

2.2.4. 不正持ち出し後

X氏は、結果、不正であっても通常業務を実施しているようにしか見えず、周りからも怪しまれずDLPでも検知されなかったので、機密情報を持ち出すことに成功しました。しかしこの後、持ち出した機密情報も十分に活用できず、最終的には不正持ち出しが判明することになります。

- X氏は、新しいC社の環境で持ち出した機密情報を活用することを試みるが、その情報と同分野でより最新の技術／情報が世間で発表され、持ち出した情報が程なく古いものとなり、結果C社では十分に活用できない状態になった。
- また後日、A社にて定期的に実施される「退職者／プロジェクト離任者チェック（監査）」にて、X氏のPC／ファイルサーバアクセス履歴が改めて確認される手続きが踏まれることになった。その際に、X氏が「通常業務上ではアクセスする必要がないはずの機密情報にもアクセスしていた」ことが判明し、A社内で疑いを持たれ、さらに追加の詳細調査が行われることになった。最終的には、過去の持ち出しファイル一覧も履歴もチェックされ、業務上予定されていない機密情報ファイルの持ち出しが行われていたことが判明し、不正が露呈する結果となった。

内部不正で持ち出した機密情報は、転職先の競合他社内で堂々と使うことはできず、その価値も長期的に担保できるとは限りません。また不正持ち出しが発覚すれば、A社やB社からのX氏への損害賠償請求、場合によっては逮捕や書類送検などの刑事的な罰則が発生します。よって現実は、不正に持ち出した機密情報ファイルを使って利益を得ることが難しく、あとから内部不正が発覚して責任を追求されるケースがあります。

2.3. 内部不正の特性別のセキュリティ対策

このような内部不正への対策は、さまざまな団体が発表しているガイドライン等が参考になります［5］［6］。内部不正は、以下のような特性によって、内部犯の行動が異なるため、セキュリティ対策は難しいのです。この内部不正の特性からセキュリティ対策の方針を定め、それらをもとに具体的な方法を検討する方法を推奨します。内部不正の特性は、不正を見つからないようにこっそり実施する隠蔽タイプと、見つかる前提で内部不正を実施する大胆タイプに大きく分かれます。それぞれに有効なセキュリティ対策が異なります。

2.3.1. 隠蔽タイプの内部不正のセキュリティ対策

まず、できる限り内部不正を見つからないように実施したい隠蔽タイプの内部不正は、慎重に内部不正を実施します。内部不正が発覚して職を失ったり、法的罰則や損害賠償を請求されたりするリスクをおそれる場合が、隠蔽タイプです。隠蔽タイプは、一般的に内部不正を行う要因とされる「不正のトライアングル」へ干渉するセキュリティ対策を実施すれば、一定の防止効果が期待できます［7］。以下に、不正のトライアングルの「機会」・「動機」・「正当化」の３側面に基づくセキュリティ対策例を挙げます。

- 『機会』：内部不正が可能な環境／手段をできる限り奪う
  （例）機密情報に対するアクセス制限、特権の管理　等
- 『動機』：内部不正を実施することが難しい状況であること、及び見返りが少ないことを理解させる
  （例）社員の端末・セキュリティ機器のログ監視及びその周知
- 『正当化』：理由付け・犯罪の弁明をさせない状況を作るためにルール化・教育を徹底する
  （例）社員への教育・ルールの徹底　等

これらのセキュリティ対策を実施すれば、内部不正を働こうとする意志を弱めることにつながり、内部不正を防止できます。

2.3.2. 大胆タイプの内部不正のセキュリティ対策

一方、内部不正が見つかっても良いと思っている場合は、不正のトライアングルのセキュリティ対策のうち、教育や監視による牽制などの対策は効果がありません。例えば、内部不正が見つかったらすぐ会社を辞めてしまえばよいと考えている産業スパイならば、あらゆる方法で機密情報を持ち出して、内部不正が見つかった場合は、行方をくらましまいます。

こういった教育や牽制の効果がない大胆タイプは、あらゆる重要な行為を基本的に禁止する方法が有効な対策です。機密情報は基本的にアクセスを禁止とし、機密情報へアクセスするときは必ず管理者が都度内容を確認して許可を出します。また、アクセスした機密情報を追跡したり、内部不正を検知した場合には、即時にその行為を停止したりすることも重要です。これらを実現するには、権限管理、アクセス制限を行うシステム、及び内部不正の検知時に即時に操作を停止するシステムの導入と運用が必要になります。

2.4. 内部不正対策ソリューション

現在、内部不正対策は、情報漏洩対策のDLP、権限管理の特権管理システム等、さまざまなソリューションがあります。それらを効率的に利用すれば、大胆タイプだけでなく、隠蔽タイプの内部不正も含めて、費用対効果のよい内部不正対策を実現できます［8］［9］。

近年、内部不正の検知に効果が高いと言われているソリューションがUEBA(User and Entity Behavior Analysis)製品です。UEBAは、セキュリティ機器や各ITデバイスのログを集約して監視、機械学習を利用して分析を行います。機械学習の結果、通常業務を行っているように見えるが普段の業務とは異なる怪しい、内部不正の疑いがある行為をすばやく検知することができます。具体的な例は、以下の通りです [10]。

- 普段は業務でアクセスしていないサーバへアクセスしている
- 普段は業務でアクセスしていないWebサイトへファイルをアップロードしている
- 普段のメール送信回数よりもメール多い回数が多い
- 普段は利用しない時間帯にシステムを利用している
- 普段とは異なるIPアドレス、拠点からシステムへアクセスして業務を行っている

これらの行為は、禁止している行為ではなく、所有している正当な権限を使った行為で、一見、通常業務に見えます。しかし、部分的には普段とは異なる行為のため、内部不正のおそれもあります。UEBAは、大量のログを監視、分析して、このような正当な権限を使った行為だが、普段とは異なる部分をすばやく自動的に検知できます。

大胆タイプの内部不正は、教育や牽制の効果がないため、システムで強制的に禁止したり、制御したり、管理するセキュリティ対策でなければ防止できません。UEBA製品の検知結果を使ってSOAR製品が他製品と連携して、自動対応すれば、大胆タイプの内部不正も対策できます。

内部不正の手段や技術は日々巧妙化しています。企業は、DLPや特権管理システム、UEBA等の内部不正対策ソリューションを使って、内部不正を効率的に防止、検知して、重要な情報資産を守らなければなりません。

3. 脆弱性『急増するパブリックAPIの脆弱性対策』

NTTデータ SL事業本部 セキュリティ＆ネットワーク事業部　板垣 毅

近年、パブリックAPI（アプリケーション・プログラミング・インターフェース）の利用は、クラウドサービスの普及やマイクロサービスアーキテクチャへの移行に伴い、急速に拡大しており、企業のビジネス成長を支える重要な技術の一つとなっています。しかし、同時にパブリックAPIの脆弱性も増加しているため、それを狙うサイバー攻撃の脅威も高まっています。パブリックAPIのサービス提供事業者にとって、適切なセキュリティ対策を講じることは不可欠です。本レポートでは、急増するパブリックAPIの利用に関連する脅威と、パブリックAPIのサービス提供事業者に必要なセキュリティ対策を解説します。

3.1. パブリックAPIへのサイバー攻撃

API自体はコンピュータ技術の黎明期より利用されており、決して新しい技術ではありません。しかし、クラウドサービスの普及やマイクロサービスアーキテクチャへの移行に伴い、APIの利用、特にインターネットからアクセス可能なパブリックAPIの利用が急増しています。パブリックAPIを利用すれば、システム間の連携が容易になり、サービスの提供が迅速化するといったメリットがあります。その一方で、パブリックAPIの脆弱性を狙ったサイバー攻撃も増加しています。

2024年6月31日にAkamai Technologies, Inc.が発表した報告書によると、アジア太平洋地域におけるパブリックAPIとアプリケーションへのサイバー攻撃は、65%増加しています [11]。しかし、パブリックAPIのセキュリティに対する企業の意識はまだ低く、セキュリティ対策も不十分なため、大きなリスクを抱えています。パブリックAPIには、以下のサイバー攻撃のリスクがあります。

- **不正アクセス：**
  認証情報を盗んでパブリックAPIへ不正にアクセスする攻撃。アクセスポリシーの誤設定が原因で不正アクセスに成功する場合が多い
- **ブルートフォース攻撃：**
  パスワードを総当たりで試行して不正にログインする攻撃
- **インジェクション攻撃：**
  SQLインジェクションやクロスサイトスクリプティング（XSS）など、悪意のあるコードをパブリックAPIに挿入する攻撃
- **DDoS攻撃：**大量のリクエストを送信してパブリックAPIを過負荷にし、サービスの停止を引き起こす攻撃
- **レイヤー7 DDoS攻撃：**
  アプリケーション層を狙ったDDoS攻撃。攻撃者は、大量のHTTPリクエストを送信したり、Webアプリケーションの特定の機能を過剰に利用してWebサーバやアプリケーションサーバの処理能力を圧迫して、サービスの提供を妨害したりする。少量のトラフィックでも大きな影響を与えることができる
- **APIの悪用：**
  APIのビジネスロジックを悪用して不正な操作を行う攻撃

過去には、パブリックAPIの脆弱性を悪用した以下のサイバー攻撃で、Facebook

やX(旧Twitter)をはじめ、多くの企業や組織で情報流出が発生しています。IBMの「2023年データ侵害のコストに関する調査レポート」 [12] によると、パブリックAPIからの情報漏洩は、他のセキュリティ侵害の情報漏洩と比べて10倍以上の被害をもたらします。

（1）2021 年 6 月から 2022 年 1 月の間に攻撃者が X（旧 Twitter）のパブリック API のゼロデイ脆弱性を悪用して、X のアカウント情報を不正に取得した。攻撃者は、X のパブリック API へターゲットのユーザのメールアドレスや電話番号を入力して、そのユーザの ID を入手した。入手した ID を使用して、公開されているアカウント情報（ID、名前、ユーザ名、位置情報、認証ステータス、電話番号、メールアドレス）5.4 百万件を窃取した。攻撃者は、窃取したアカウント情報をハッキングフォーラムで販売して、その後、無料で公開した [13]。

（2）2024 年 2 月 15 日に Facebook のパスワードリセット処理の認証の不備が原因で、攻撃者が不正にログインして、ユーザのアカウントを乗っ取った。攻撃者は、パブリック API を使ってパスワードリセットを要求した。Facebook のシステムは、Facebook の通知を使って、パスワードリセット用の 6 桁の認証コードを正規のユーザへ送信した。Facebook のシステムは、パスワードリセット時に 6 桁の認証コードの入力を求めましたが、試行回数の上限を設定していなかった。攻撃者は、パブリック API で 000000 から 999999 までのすべてのパターンを試して、不正ログインしてパスワードをリセットして、アカウントを乗っ取った。乗っ取ったアカウントから、ユーザの個人情報が漏洩して、フィッシング攻撃やその他の詐欺行為が発生した [14]。

最近では、DeepSeekの関係者がパブリックAPIを悪用してOpenAIから生成AIに関する大量のデータを盗み出した疑いがあります [15]。このようにパブリックAPIを悪用すると短時間で大量の情報窃取が可能なため、パブリックAPIの脆弱性は企業にとって重大な脅威です。

3.2. パブリックAPIの脆弱性

パブリックAPIに多い脆弱性は、認証や認可の設定に関する不備です。OWASP API Security Top 10レポート [16] でも、認証や認可に関連する脆弱性が上位にあります。例えば、同レポートの1位は、Broken Object Level Authorization（BOLA）と呼ばれる脆弱性です。BOLAは、APIへリクエストメッセージを送信するときにオブジェクトレベルの認可設定が欠如しているため、攻撃者がそのAPIリクエストメッセージのユーザ識別子を正規のユーザの識別子へ書き換えて送信して、不正にリクエストメッセージを実行できる脆弱性です。パブリックAPIに関係した多くの情報漏洩インシデントは、このBOLAを悪用しています。

BOLAを悪用したサイバー攻撃を図 3-1を使って説明します。一般ユーザ YYYさんは、アカウントを認証してパブリックAPIへアクセス可能になったあと、自分のメッセージID “YYY” を使ってパブリックAPIへリクエストメッセージを送信して、自分のプロファイル情報を取得します。一般ユーザ ZZZさんも、同様に自分のメッセージID “ZZZ” を使って自分のプロファイル情報を取得します。悪意のあるユーザXXXは、認証後にメッセージID “XXX”の代わりに、他人のメッセージID “YYY”や“ZZZ”を使って作成したリクエストメッセージを送信して、他人のプロファイル情報を不正に取得できます。

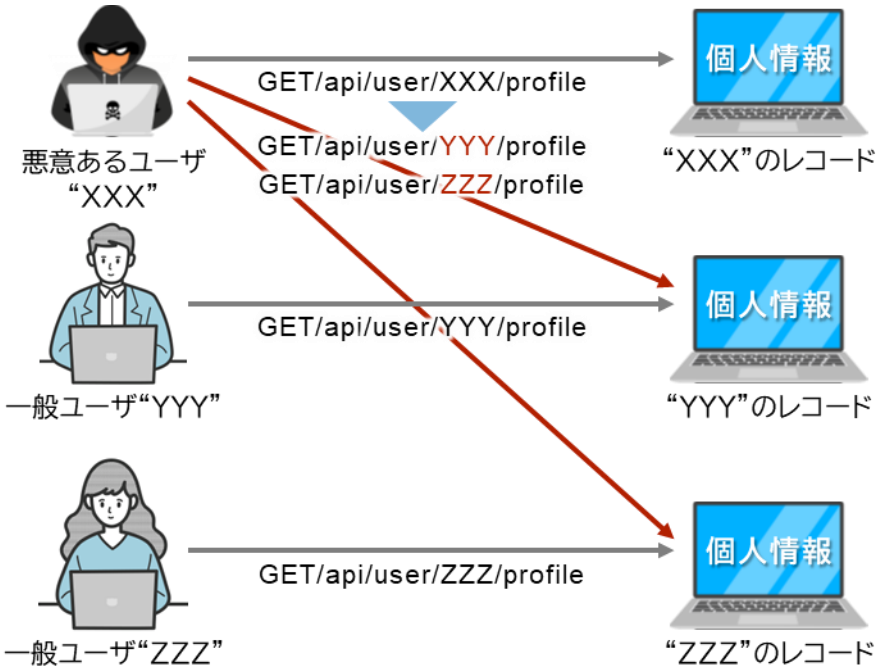


図 3-1：BOLAの脆弱性を悪用したサイバー攻撃の例

このように、パブリックAPIの脆弱性を悪用したサイバー攻撃は、プログラムを使って自動でリクエストメッセージを作成してパブリックAPIへ送信するだけなので、短時間で大量のサイバー攻撃を行うことができます。パブリックAPIへのサイバー攻撃以外のあるサイバー攻撃の場合、例えば攻撃者は、VPN機器の脆弱性を攻撃してVPN接続できるIDとパスワードを入手したあと、手動でVPN経由で内部ネットワークへ侵入します。内部ネットワークを調査して攻撃対象のADサーバやデータベースサーバを見つけたら、別のサイバー攻撃でサーバへ不正ログインして機密情報を盗んだり、ランサムウェアを実行したりします。このサイバー攻撃方法と比較して、パブリックAPIへのサイバー攻撃は手動操作の部分が少ないのです。少ない作業量で大量の機密情報を窃取できるため、短時間で大きな被害が発生する点も特徴です。

3.3. パブリックAPIのセキュリティ対策

パブリックAPIのサービス提供事業者は、システムの開発時にパブリックAPIへ脆弱性を作り込んでしまわないようセキュリティ対策をします。システムやアプリケーションの設計段階からセキュリティを考慮し、脆弱性を未然に防ぐアプローチである「セキュアバイデザイン」が、脆弱性の作り込みに対するセキュリティ対策です。以下に、システム開発者や組織が採用できるAPI を強化するためのセキュリティ対策の戦略的プラクティスを紹介します [17] [18]。

- **認証と認可の実装：**
  多要素認証（MFA）、OAuth 2.0、APIキーを使った認証方式を使用するときは、それぞれの仕組みを理解して、安全に設計、実装する。認可のスコープ設定やロールベースアクセス制御（RBAC）にもとづいた細かなアクセス権限の設定、アクセストークンへ有効期限の設定などを行う
- **ランタイム保護：**
  稼働中のAPIをリアルタイムで保護するための一連のセキュリティ対策。APIで受信したデータを厳密に検証し、SQLインジェクションやクロスサイトスクリプティング（XSS）などの攻撃を防ぐ。APIへのリクエスト数の制限やトークン検証をおこなう。APIをリアルタイムで監視して、不審なアクセスや異常なパターン、異常な振る舞いを検知して対応する。攻撃者の攻撃パターンを認識し、異常なリクエストをブロックする
- **セキュリティテスト：**
  開発プロセスの早い段階でセキュリティテストを行って、APIの脆弱性を

特定して対処する。CI/CD パイプラインにセキュリティテストを統合する

- **セキュリティ・ポスチャー管理：**
  パブリックAPIの設定や実装に潜む脆弱性を評価して修正する
- **APIのカタログ化とドキュメント管理：**
  すべての API のインベントリを作成、維持して、APIを効率的に管理する。詳細な APIの ドキュメントを作成して、管理する
- **APIガバナンスとAPIセキュリティ：**
  APIガバナンスやAPIセキュリティのポリシを策定して、APIの設計、実装、デプロイ、運用を統制する。APIの設計や実装の方針やセキュリティ対策の方針、セキュリティテスト、セキュリティ・ポスチャー管理、API カタログなど、上記のAPIのセキュリティ対策に必要な要素をポリシへ含める

設計および開発フェーズの上流工程にセキュア設計やセキュアコーディングに詳しいセキュリティ技術者を加えて、APIの開発の初期段階からセキュリティ対策を意識して、上記のセキュリティ対策の戦略的プラクティスの実践を確実にします。またAPIへ脆弱性の作り込みを未然に防ぐために、積極的にシフトレフトのセキュリティ戦略を採用します。脆弱性を発見したら、その脆弱性を作り込んだ原因とタイミングを特定して、上流工程で脆弱性の作り込みを対策します。

しかし、セキュアバイデザインでも、脆弱性を完全に排除することは困難です。実際の開発現場においては、ミスによる脆弱性の作り込みだけでなく、不要になり本来削除されるべきだが、忘れ去られて放置された「ゾンビAPI」や開発や運用を効率化するためにルールを違反して作成した「シャドーAPI」の問題があります。ゾンビAPIやシャドーAPIの問題は、APIガバナンスとAPIセキュリティだけでは解決できません。APIディスカバリーを行えば、企業が保有するAPIを特定してカタログ化して、ゾンビAPIやシャドーAPIを特定できます。

先進的なツールやソリューションを導入して、APIセキュリティを強化する方法もあります。APIゲートウェイやAPIプラットフォームです。APIゲートウェイは、APIに対するアクセスを一元管理して、認証やレートリミット、リクエストのログ記録などを行います [19]。APIプラットフォームとは、APIを提供する企業や組織が、APIを公開、管理するための基盤です。APIキーやトークンを使った認証やIPフィルタリングなど、APIを保護するセキュリティ機能も提供します。APIを提供する企業や組織は、APIの公開や管理を効率化でき、APIを利用する開発者は、APIを検索できる便利なツールです [20]。

3.4. まとめ

本記事では、パブリックAPIの利用拡大に伴う脅威とサービス提供事業者に必要なセキュリティ対策を解説しました。具体的には、不正アクセス、インジェクション攻撃、DDoS攻撃、APIの悪用などのサイバー攻撃が増加しており、XやFacebookのAPI脆弱性を悪用した事例を紹介しました。パブリックAPIには、認証や認可の設定に関する不備が多く、特にBroken Object Level Authorization（BOLA）などの脆弱性が悪用されて、大量の情報が漏洩しています。

そこで、セキュアバイデザインのアプローチを採用して、開発の初期段階からセキュリティを組み込むことを提案しました。認証と認可の実装やランタイム保護、APIガバナンスとAPIセキュリティのポリシ策定などのセキュリティ対策や、積極的なシフトレフトのセキュリティ戦略を推奨しました。

しかし、セキュアバイデザインでも解決できない問題も存在します。これらの問題を解決するために、APIディスカバリーの導入、自動化されたセキュリティツールの活用、セキュリティ教育の強化が必要です。開発者や運用担当者のセキュリティ教育を強化して、セキュリティ意識を高めて、脆弱性の作り込みを未然に防ぐ努力をしましょう。これらの方針を採用して、パブリックAPIのセキュリティを強化すれば、より安全なパブリックAPIを提供できます。

4. 脆弱性『SSVCを用いた脆弱性トリアージ手法の改善』

NTTデータグループ C&I技術部 情報セキュリティ推進室　村田 直樹

サイバー攻撃の高度化やシステム環境の複雑化により、脆弱性管理は組織の安全性を確保するための重要な課題となっています。脆弱性情報の公開数の増加に対して、組織には、効率的かつ適切な脆弱性管理手法の導入が求められています。本稿では、SSVC（Stakeholder-Specific Vulnerability Categorization）という脆弱性対応の優先度評価に関するフレームワークを取り上げます。SSVCは米国土安全保障省（DHS）のサイバーセキュリティ・インフラストラクチャセキュリティ庁（CISA）でも活用されており、将来的に脆弱性トリアージの標準となる可能性があります。脆弱性トリアージ手法を導入している各システムの脆弱性対応の担当者にとって、現段階からSSVCへの理解を深めていくことは、将来的に価値があるでしょう。本フレームワークを例に、弊社内プロジェクトで脆弱性の管理方針の検討時に生じた課題とそれに対する考え方を掘り下げ、脆弱性対応の優先度を評価する時に検討すべきポイントを整理します。

4.1. 脆弱性情報の件数増加とその影響

昨今、脆弱性情報の公表件数が増加し続け、それに伴い、大量の脆弱性情報を管理する手法の検討が重要な課題となっています [21]。この課題の背景には、技術の進化とともに複雑化するシステム環境や、多様化するサイバー攻撃手法が関連しています。脆弱性情報の公表件数の増加により、各システムの脆弱性対応を行う担当者にも課題が発生します。もし担当者がセキュリティ分野の専門家でない場合や、日常業務の多忙さから脆弱性管理に十分なリソースを割けない状況では、脆弱性対応の優先度の適切な判断が難しく、脆弱性対応を後回しにしてしまう場合があります。このような状況は、弊社のプロジェクトにおいても例外ではありません。実際に現場のプロジェクトが脆弱性管理をできるようにするには、専門知識がなくても誰でも効率的に取り組める運用フローでなければなりません。

本稿では、私があるプロジェクトへSSVC（Stakeholder-Specific Vulnerability Categorization）というフレームワークを活用して、脆弱性情報から対策の優先度を評価する方法を導入した事例を紹介します。SSVCは、決定木を使って脆弱性の対応の優先度を機械的に決定できる方式です。SSVCの決定木の構造を理解すれば、優先度の判断を間違えた箇所の特定や修正が容易です。このSSVCを活用して、脆弱性情報をもとに対応の優先度を評価する手順を整理しました。その過程で、いくつかの課題に直面しましたため、その時に整理したSSVCの使い方や考え方を説明します。

4.2. 脆弱性の評価手順の改善

4.2.1. 脆弱性評価システム(CVSS)の課題

弊社のあるプロジェクトにおいて、脆弱性のトリアージ基準を整理するためにSSVC（Stakeholder-Specific Vulnerability Categorization）というフレームワークを活用しました [22] [23]。これまでの脆弱性の深刻度を評価するシステムCVSS（Common Vulnerability Scoring System） [24]には、3つの問題があります。

1. CVSSスコアから脆弱性の対応方法がわからない：
   脆弱性の深刻度を表すCVSSスコアを算出できるが、各システムの脆弱性対応の担当者は、この値から脆弱性の具体的な対応方針を決定することができません。CVSSには、CVSSスコア（深刻度）から、脆弱性対応期限などの具体的な方針、わかりやすい対策手順の示唆がありません。
2. 計算方法が複雑でCVSSスコアの添削や修正が難しい：
   CVSSは、計算式が複雑で脆弱性評価項目とCVSSスコアの関係を理解できません。そのため、CVSSスコアの添削や修正が難しいです。1つ目の問題点にも関連しますが、例えばCVSSスコアと脆弱性の対応期限の変換表を定義しておけば、CVSSスコアから脆弱性の対応期限を決定できます。それを使って「AV：NでPR：Nなど脆弱性評価項目が・・・なのでCVSSスコアは10.0となる。そのため12時間以内の脆弱性対応が必要」と判断できます。しかし、脆弱性管理の担当者は、AVとPRなどの脆弱性評価項目から対応期限を12時間以内に決定した構造を理解できません。12時間以内という結果が直観的に妥当ではないと思った時に、AVとPRなどの間違っている脆弱性評価項目を特定して値を修正することは大変です。
3. 現状評価基準や環境評価基準による再評価が困難：
   現状評価基準のCVSSスコアは、脆弱性の被害有無を実際のシステムで検証した結果や脆弱性を狙ったサイバー攻撃の発生有無などのサイバー攻撃の流行状況が変化したときに、その情報を収集して再計算しなければならなりません。環境評価基準のCVSSスコアは、システム環境特性の変化など、脆弱性に関係するシステムの状況が変化したときも、その情報を収集して再計算しなければなりません。この2つの評価基準のCVSSスコアを再計算する際に収集しなければならない情報は、簡単に手に入らなかったり、手間と時間がかかったりします。

4.2.2. SSVCの適用による課題解決

SSVCを活用すれば、このCVSSの3つの問題を改善できます。SSVCは、パッチを提供するベンダーやパッチを適用するユーザなどの脆弱性対応を行うステークホルダーごとに脆弱性の優先度や対応方針を決定できるフレームワークです。SSVCは、決定木（図 4-1 参照）を使用して脆弱性の影響や緩和策などの要因を評価して、4つの対応の優先度「Defer（対応しない）」「Scheduled（定期対応）」「Out-of-cycle（迅速対応）」「Immediate（緊急対応）」を示します。図 4-1は、ソフトウェアやシステムの導入、設定、運用、および保守を行う組織やシステム管理者が利用する決定木「Deployer Tree」です。各ステークホルダーの脆弱性対応担当者は、決定木を見れば、視覚的かつ論理的に脆弱性の対応優先度の判断のプロセスを理解できます [25], [26]。

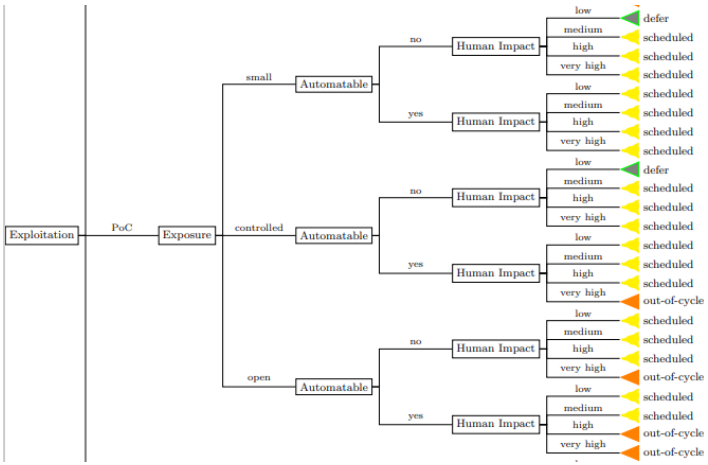


図 4-1：SSVCv2.1 Deployer Treeの決定木（Exploitation=PoCのみ抜粋） [22]

1つ目の「CVSSスコアから脆弱性の対応方法がわからない」という問題は、SSVCを使えば、解決します。SSVCは、決定木（Decision Tree）を用いて、脆弱性の深刻度の数値ではなく、脆弱性対応の優先度を出力します。SSVCの脆弱性対応の優先度は、高い順に「Immediate」「Out-of-cycle」「Scheduled」「Defer」の4段階です。以下に脆弱性対応の4段階の優先度を説明します。各システムの脆弱性対応の担当者は、SSVCで決定した優先度に沿って脆弱性に対応すれば、問題は解決します。

表 4-1：脆弱性対応の優先度

No.	脆弱性対応の優先度	説明
1	Defer	現時点では対応しない
2	Scheduled	定期的なメンテナンスで対応
3	Out-of-cycle	緩和策または修正策を迅速に適用
4	Immediate	可能な限り迅速に修正策を適用。必要であれば、通常業務を一時停止する

2つ目の「計算方法が複雑でCVSSスコアの添削や修正が難しい」という問題は、決定木の導出過程が明示されているというSSVCの特徴により解決します。決定木の構造を理解すればチェックや間違いの特定、修正が可能となるためです。

3つ目の「現状評価基準や環境評価基準による再評価が困難」という問題も、2つ目の問題と同様に決定木を使えば解決します。SSVCの決定木には、システムのネットワーク接続状態を現す項目があります。システムのネットワーク接続状態が変化した場合は、その項目を使った決定木の分岐点である決定点（Decision Point）から先を更新するだけで複雑な再計算なしに、脆弱性対応の優先度を判定できます。

4.2.3. 2つの決定点の判断方法

以降は、具体例を使いながら、SSVCの4つの決定点（表 4-2参照）のうち、ExposureとHuman Impactの2つの決定点の判断方法を説明します。

表 4-2：SSVCv2.1の決定点（Decision Point） [22]

決定点 (Decision Points)	説明
Exploitation	脆弱性の悪用状況（または悪用可能性）を評価
Exposure	システムの外部露出度を評価
Automatable	攻撃の自動化が可能かどうかを評価
Human Impact	Situated Safety ImpactとMission Impactを組み合わせて評価
Situated Safety Impact	物理的な安全性や健康に与えるリスクを評価
Mission Impact	重要なビジネスプロセスやミッションに対する影響を評価

（1） 説明に用いるシステム例

ここで説明に使用するシステム例は2つです。例１は、図 4-2で示したパブリッククラウド上に構築されたシステムで、本番環境と検証環境の2つの環境で構成しています。例2は、図 4-3で示したオンプレミス基盤上に構築したシステムです。

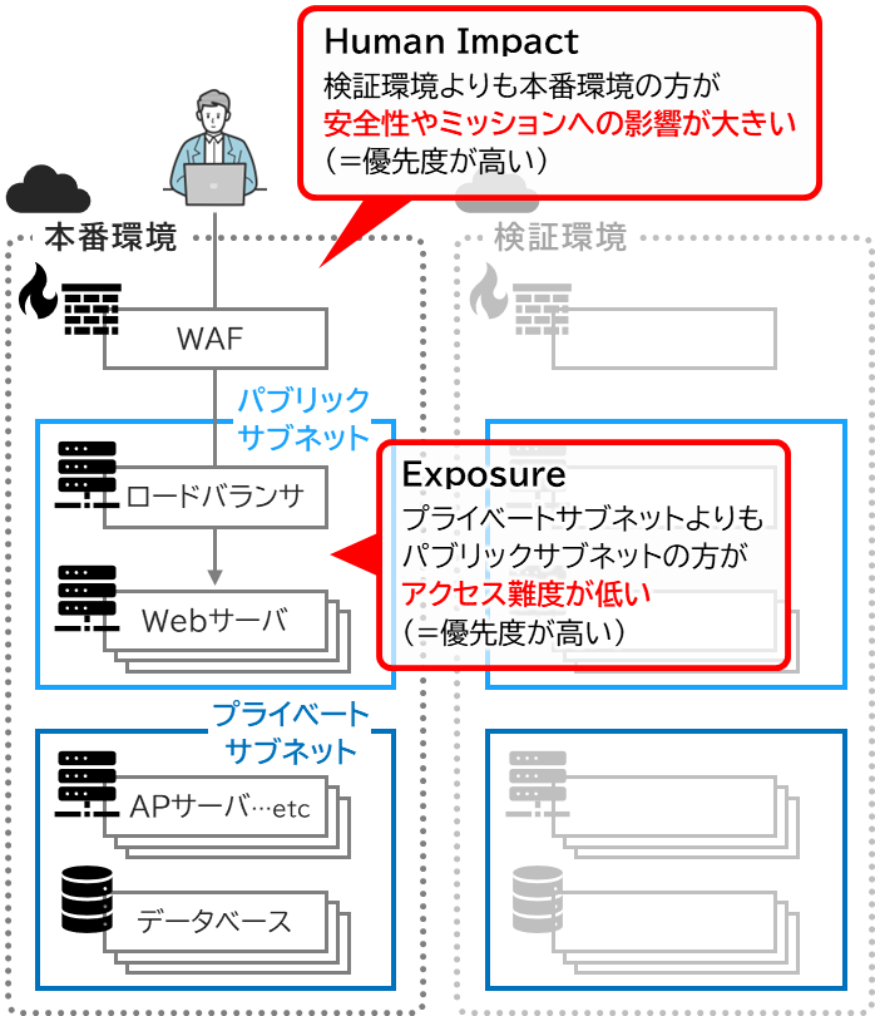


図 4-2：例1 パブリッククラウド上のシステム（概略）

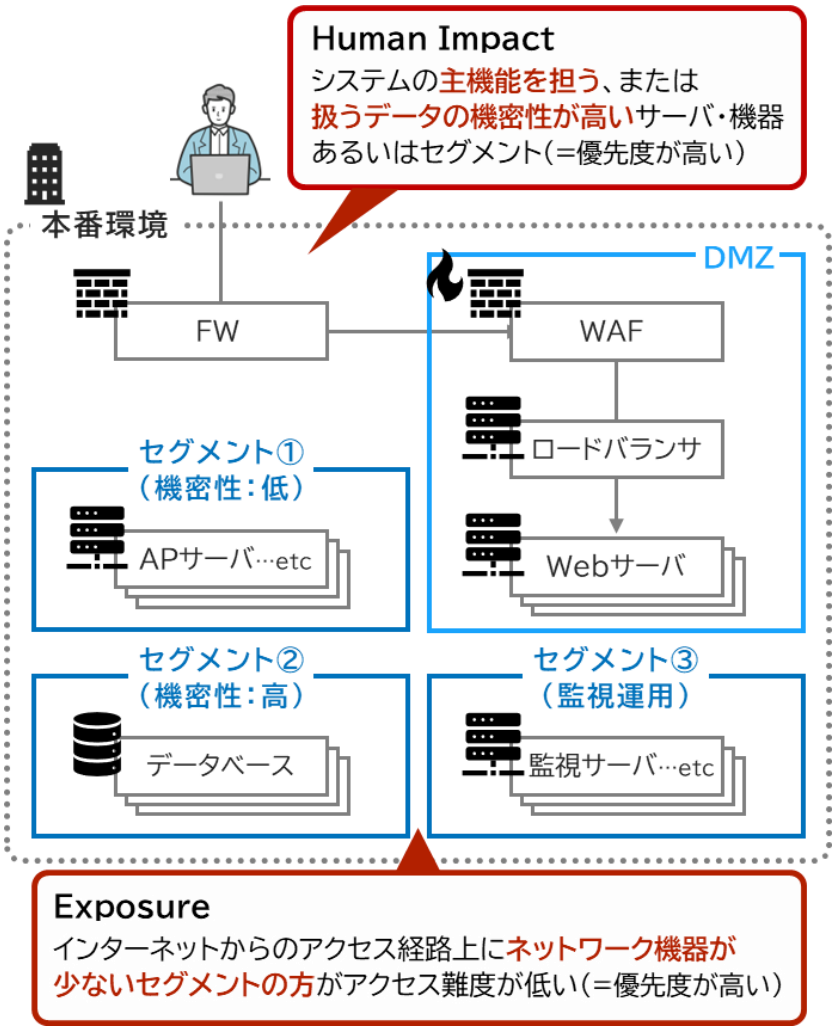


図 4-3：例2 オンプレミス基盤上のシステム（概略）

（2） SSVC の Exposure の判断方法

Exposure は、インターネット上から脆弱性のあるサーバやネットワーク機器、ソフトウェア等へのアクセスの難度を表します（表 4-3参照）。インターネットから直接アクセス可能な場合は「Open」、Firewallなどでアクセス制御していてインターネットから直接アクセスできない場合は「Controlled」、インターネットから隔離され閉域内に存在する場合は「Small」です。

図 4-2のパブリッククラウドのシステムでは、パブリックサブネットに配置しているロードバランサやWebサーバは「Open」と判定します。プライベートサブネットに配置されているAPサーバ/Webアプリケーションやデータベースは「Controlled」と判定します。

図 4-3のオンプレミス基盤上のシステムでは、DMZに配置したWebサーバやネットワーク機器はインターネットから直接のアクセスが可能なため「Open」、その他の内部セグメントのサーバはFirewall等でアクセス制御できているため「Controlled」と判定します。攻撃者がインターネットから脆弱性があるサーバやネットワーク機器、ソフトウェアへアクセスしやすいほど危険なため、脆弱性対応の優先度が高くなります。

表 4-3：Exposureの評価値 [22]

No.	評価値	説明
1	Open	インターネットまたはアクセスを制限または制御できない可能性のあるネットワーク
2	Controlled	アクセス制限及び緩和策が実施されているネットワーク
3	Small	インターネットへの入出力がない閉域環境

（3） SSVC の Human Impact の判断方法

Human Impactは、表 4-2のように脆弱性が悪用された場合の安全性への影響を表す「Situated Safety Impact」と組織のミッションへの影響「Mission Impact」を掛け合わせた結果で表現します。

① Situated Safety Impact：

Situated Safety Impactは、表 4-4に示す通り、脆弱性が「身体的被害」、「環境への被害」、「心理的被害」、「経済的被害」の4つの被害カテゴリ「Harm Categories」へ与える影響を5段階で評価します。その4つのHarm Categoriesの中で最も影響が大きい評価値をSituated Safety Impactに採用します。例えば、脆弱性が人命に関わる病院や航空などのシステムに存在する場合は、Physical Harmの影響が最も大きくなりやすく、このPhysical Harmの評価値を採用することが多くなります。電気や水道などの生活インフラ関連のシステムの場合はEnvironmentの影響が大きく、評価値に採用することが多くなります。

表 4-4：Situated Safety Impactの被害カテゴリと評価値 [22]

被害カテゴリ（Harm Categories）	説明	評価値
Physical Harm	システム利用者への身体的な影響	1. None 2. Minor 3. Major 4. Hazardous 5. Catastrophic
Environment	自然環境や公衆衛生に関するリスクを含む外部環境への影響	

Financial	関係者の経済的損失
Psychological	関係者への心理的被害

② Mission Impact：

Mission Impactは、表 4-5に示す通り、組織のミッションに対する脆弱性の影響を評価します。脆弱性が、ミッションを達成するために必要な機能へ与える影響の程度を表 4-5に示す4段階で評価します。

表 4-5：Mission Impactの評価値 [22]

No.	評価値	説明
1	None, Degraded	ほとんど影響がない。非必須機能の劣化が、最終的に重要な機能を損なう可能性がある
2	Crippled	重要な機能を直接サポートする活動が機能不全に陥る
3	MEF Failure	ミッションに不可欠な機能のいずれかが、許容範囲を超えて長期間に渡って機能しなくなる
4	Mission Failure	複数またはすべてのミッションの必須機能が失敗する

Situated Safety Impactの評価値とMission Impactの評価値が決まったら、その組み合わせからHuman Impactの評価値を決めます。事前にSituated Safety ImpactとMission Impactのそれぞれの評価値の組み合わせから、Human Impactの評価値を決める判定ロジックを用意しておきます。例えば、Situated Safety Impactが「Major（重大）」で、Mission Impactが「Mission Failure」の場合は、その脆弱性が人々や環境に与える影響は非常に大きいと判断して、Human Impactの評価値は「Very High」と判断します [27]。

表 4-6：Human Impactの評価値 [22]

No.	Situated Safety Impact	Mission Impact	Human Imapact	説明
1	None/Minor	None/Degraded/ Clippled	Low	影響がほとんどない
2	None/Minor	MEF Failure	Mediu m	基幹業務には影響がない
	Major	None/Degraded/ Clippled		
3	Major	MEF Failure	High	1つの基幹業務に長期間影響が出る
	Hazadous	None/Degraded/Clipped/MEF Failure		
4	None/Minor/Major/ Hazadous	Mission Failure	Very High	複数の基幹業務が停止し、基幹業務が続行不能かつ回復不能となる
	Catastrophic	None/Degraded/Clipped/MEF Failure/ Mission Failure		

4.3. 脆弱性評価手順の最適化

4.3.1. SSVC導入後の課題

あるプロジェクトで、このSSVCをそのまま適合して脆弱性対応の手順を整理していくと、以下の課題が浮かび上がりました。

CVSSに比べるとSSVCの脆弱性の評価フローはよりシンプルにわかりやすくなっています、しかしSSVCは、システム特性に応じた評価基準の定義に曖昧な部分が残っています。この曖昧な部分に対応できない場合は、決定木を分岐する決定点の判断を間違ってしまい、誤った優先度に判定してしまうおそれがあります。

次に脆弱性対応の担当者は、脆弱性対応の優先度を決定するためには、サーバ毎にネットワークの接続状態を調べたり、脆弱性を悪用したサイバー攻撃の発生有無を調査したりしなければなりません。しかしサーバ数が多い場合は、脆弱性対応の担当者1人では、すべてのサーバを調査する時間が足りません。例えば、2024年第2四半期に発生したregreSSHion [28]の脆弱性は、システム内の複数のサーバが影響を受けました。

脆弱性対応の担当者は、脆弱性を悪用したサイバー攻撃の発生有無を調査しなければなりません。しかし脆弱性対応の担当者が、脆弱性の悪用状況の情報を収集する方法を知らない場合もあります。

以下が、SSVCの脆弱性対応の優先度に影響する課題の例です。

1. 評価対象のシステムのスコープや業務影響を定義できていない
2. 脆弱性対応の担当者1人では、すべてのサーバを調査する時間が足りない
3. 脆弱性の悪用状況等の外部の情報を得ることが難しい

4.3.2. 評価対象の精査と課題解決

ExposureとMission Impactの評価方法を工夫して、課題2の「脆弱性対応の担当者1人でサーバを調査できない」を解決する方法を説明します。

（1） Exposure を工夫した課題解決

脆弱性対応の担当者1人で複数のサーバすべてを調査することができないのであれば、複数のサーバやネットワーク機器、ソフトウェアをグループにまとめて、グループ毎に脆弱性対応の優先度を決定する方法を推奨します。

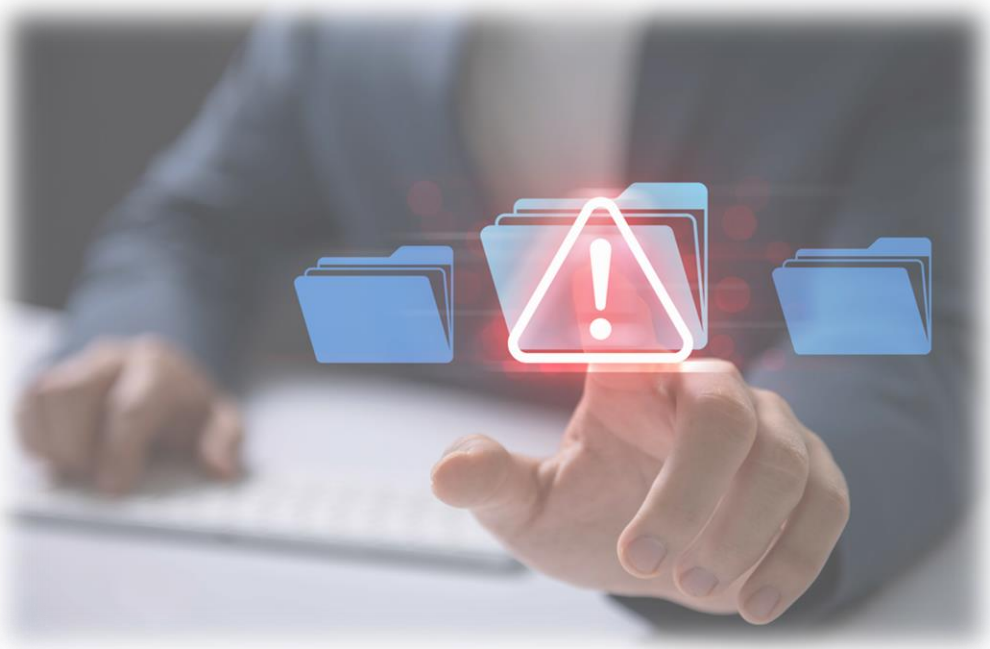
SSVCは柔軟なフレームワークなので、Exposureの適用範囲も自由に決定できます。図 4-2や図 4-3のように、インターネット上からサーバやネットワーク機器、ソフトウェア等への通信プロトコル毎にアクセス難度別で、評価対象を事前にグループ化しておきます。脆弱性の悪用リスクが大きいExposureの評価値がOpenのグループから優先的に決定木を使って脆弱性対応の優先度を決定します。サーバやネットワーク機器、ソフトウェア単位で脆弱性対応の優先度を決定したい場合は、脆弱性対応の優先度が高いグループから順に、もう一度サーバやネットワーク機器、ソフトウェア単位で決定木を使って脆弱性対応の優先度を決定します。

（2） Human Impact の Mission Impact を工夫した課題解決

課題1の「評価対象のシステムのスコープや業務影響を定義できていない」は、Exposureに加えてHuman ImpactのMission Impactも工夫すれば、解決できます。Mission Impactは、システムを構成するサーバやネットワーク機器毎に変わる場合があります。例えばシステムの本番環境と検証環境でMission Impactの評価値が異なります。本番環境の方が、脆弱性でサービスが停止して多くのユーザが使用できないおそれがあるため、影響が大きくなります。本番環境を構成するサーバやソフトウェアも、システムの機能要件の実現に該当する部分と非機能要件に該当する部分で、Mission Impactの評価値が異なります。例えばログ管理サーバや運用監視サーバは、非機能要件に該当するため、脆弱性の影響があってもシス

テムの主機能への直接的な影響はあまりありません。よってMission Impactの評価値は、NoneやDegradedになります。

あらかじめ、サーバやネットワーク機器単位、ソフトウェアやプログラム単位でMission Impactを評価しておけば、課題1で困ることはありません。あらかじめMission Impactの評価値を準備できなかった場合でも、Exposureの脆弱性対応の優先度が高いグループから順にMission Impactを評価すれば、リスクが高い部分から順に脆弱性対応の優先度を決定できます。



4.4. まとめ

SSVCを使って脆弱性を評価すれば、脆弱性の対応の4段階の優先度を決定できます。しかし、脆弱性の悪用状況に関する情報が足りずに十分な評価ができない場合もあります。または、システム内のサーバやネットワーク機器、ソフトウェアが多く、すべての機器の脆弱性の対応の優先度を決定できない場合もあります。その場合は、ExposureとMission Impactの評価方法を工夫して、あらかじめサーバやネットワーク機器、ソフトウェアをネットワークのアクセス制御の状況別にグループ化したり、Mission Impactを個別に評価したりしておけば、優先度の決定作業を効率化できます。

特にAI技術の急速な進化に伴い、脆弱性の公開後に脆弱性を悪用した攻撃コードや攻撃ツールが短期間で作成されるようになっています。このような状況では、脆弱性対応の迅速性が一層重要となります。そのため、本記事で整理したように、あらかじめExposureやMission Impactに関係するサーバやネットワーク機器、ソフトウェアをグループ化して評価値を用意しておいてから、SSVCを使って脆弱性対応の優先度を決定すれば、少ない人数でも対処可能です。このような工夫と事前準備が、脆弱性の対応力を向上する鍵になるでしょう。

5. タイムライン

NTTデータグループ C&I技術部 情報セキュリティ推進室　寺師 悠平
NTTデータグループ C&I技術部 情報セキュリティ推進室　田中 稜太郎

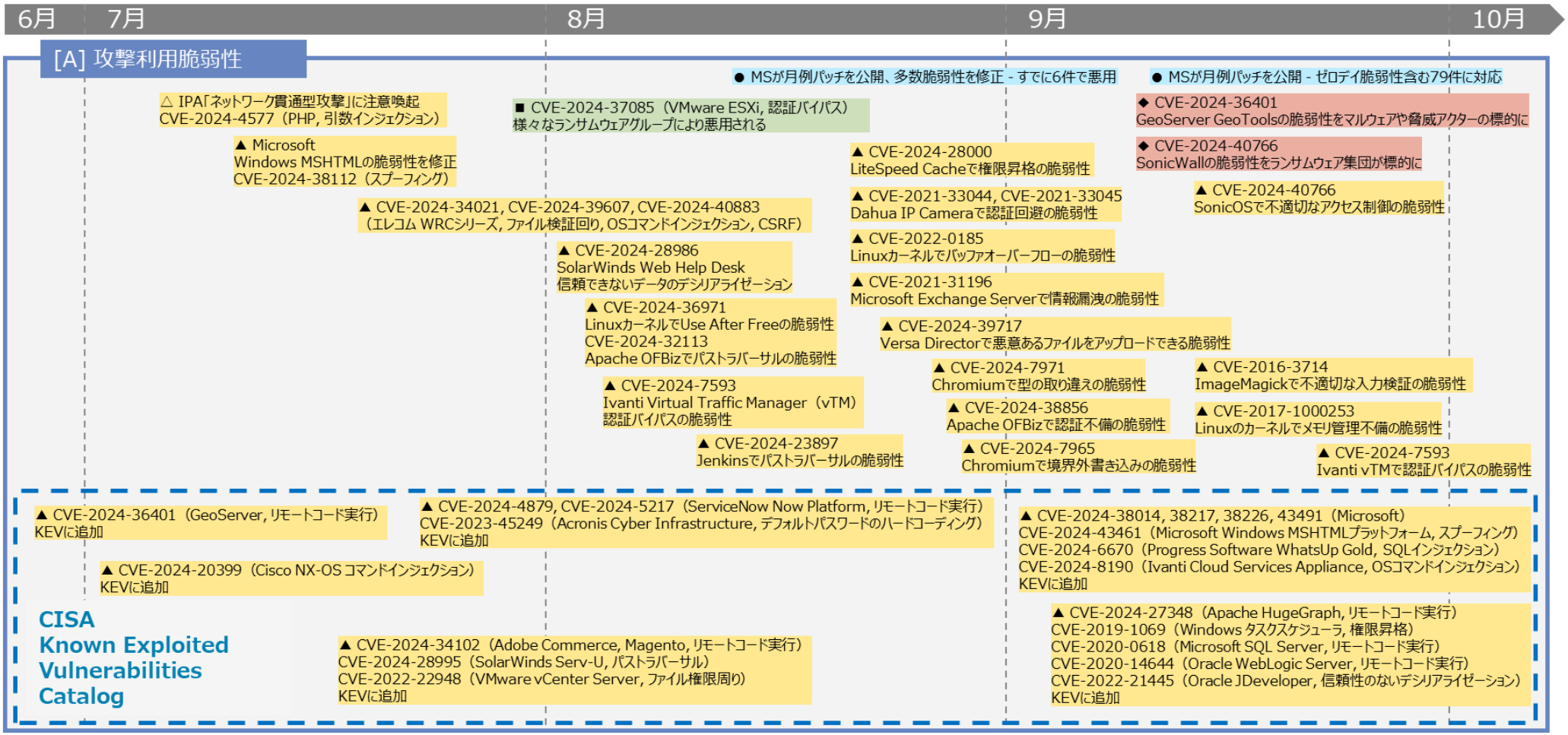
2024年度第2四半期の[A]攻撃利用脆弱性のカテゴリでは、脆弱性のPoC公開後わずか22分でサイバー攻撃が発生したというニュースが特徴的でした。また脆弱性以外のカテゴリでは、パスワードリスト攻撃のニュースが比較的多く見つかりました。2024年度第1四半期のタイムラインでも取り上げたAI関連の事象では、生成AIによるビジネスメール詐欺の増加のニュースがありました。

※タイムラインに記載している日付は事象発生日ではなく、記事掲載日の場合があります。

△□◇○:国内
▲■◆●:世界共通・国外

△▲:脆弱性
□■:事件・事故

◇◆:脅威
○●:対策

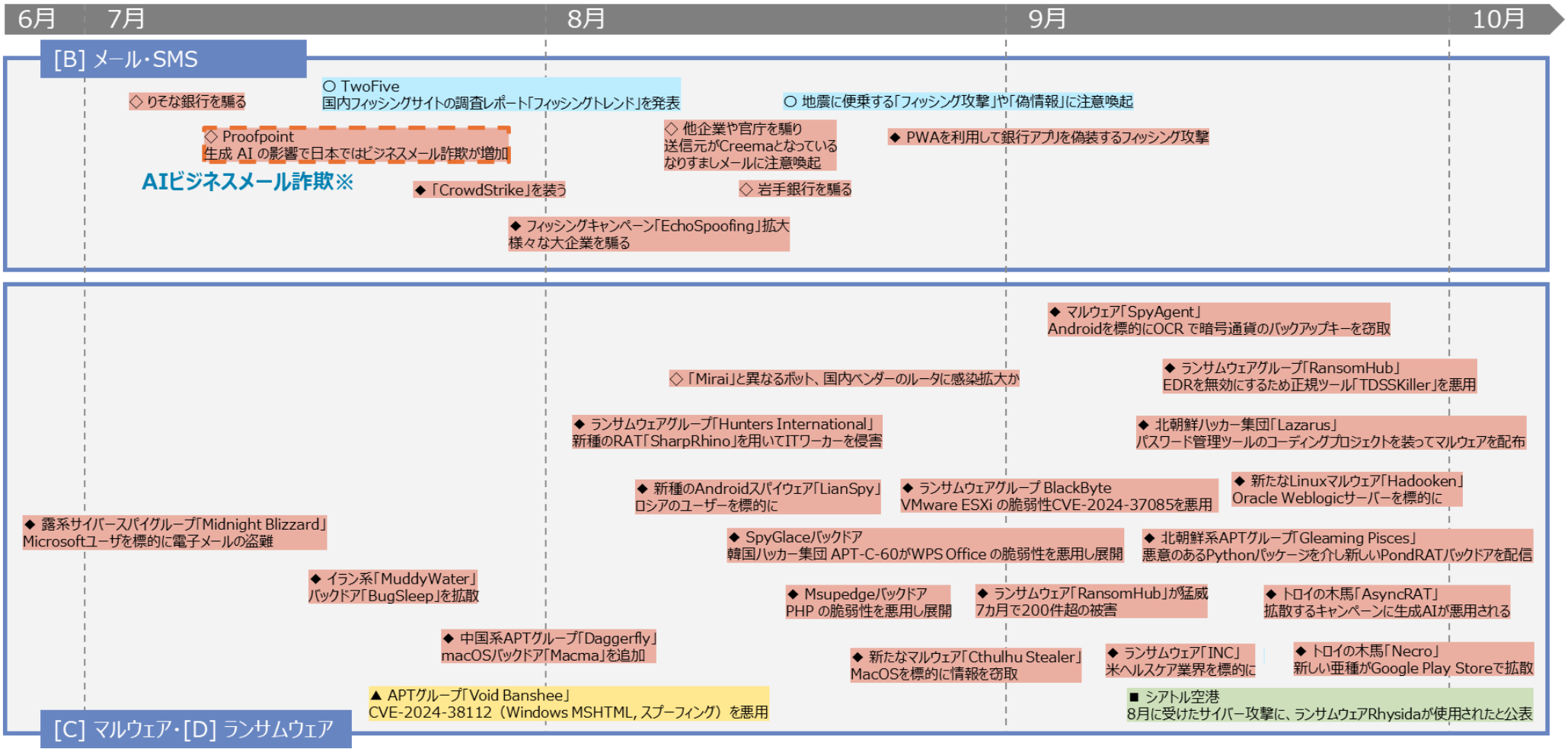


※タイムラインに記載している日付は事象発生日ではなく、記事掲載日の場合があります。

△□◇○:国内
▲■◆●:世界共通・国外

△▲:脆弱性
□■:事件・事故

◇◆:脅威
○●:対策

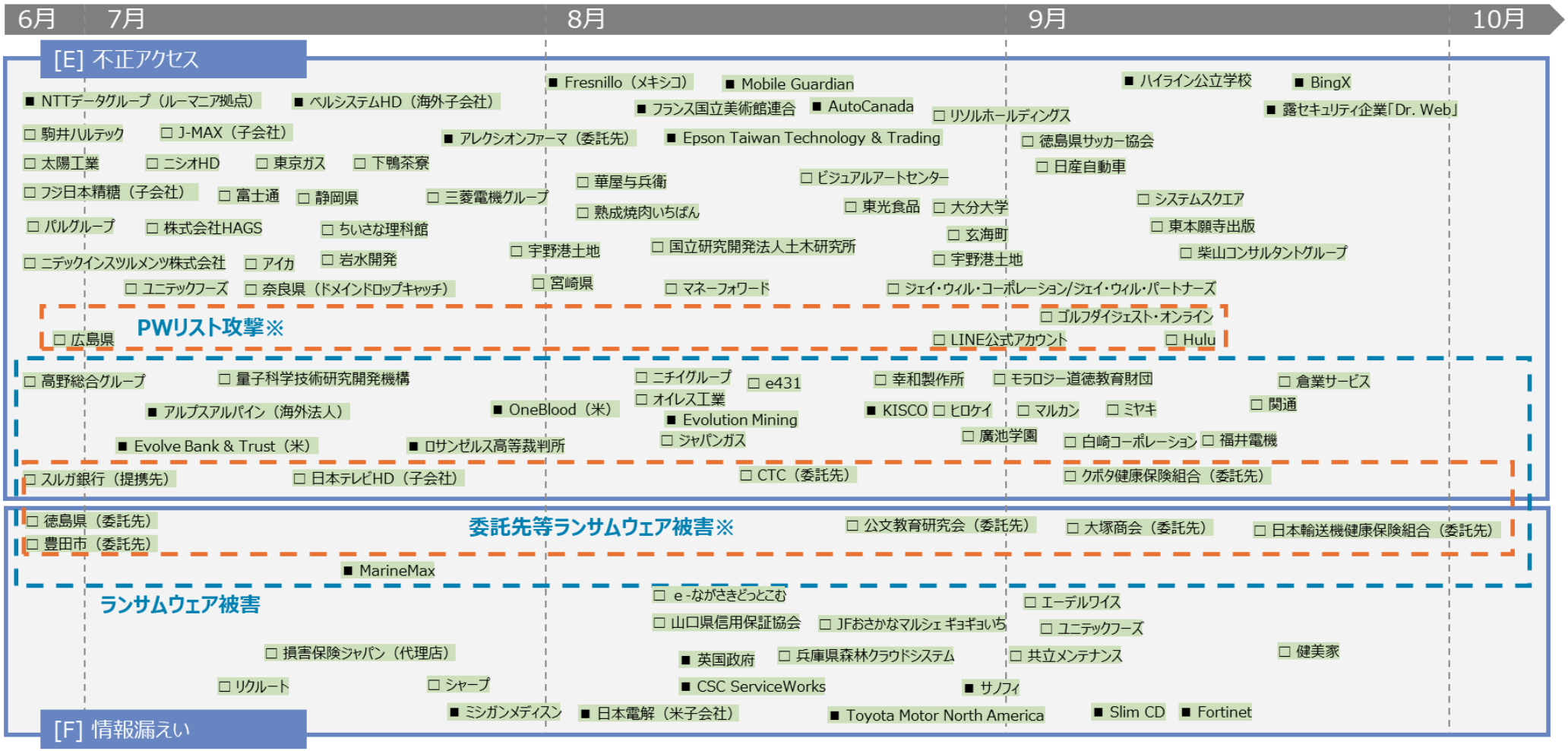


※タイムラインに記載している日付は事象発生日ではなく、記事掲載日の場合があります。

△□◇○:国内
▲■◆●:世界共通・国外

△▲:脆弱性
□■:事件・事故

◇◆:脅威
○●:対策

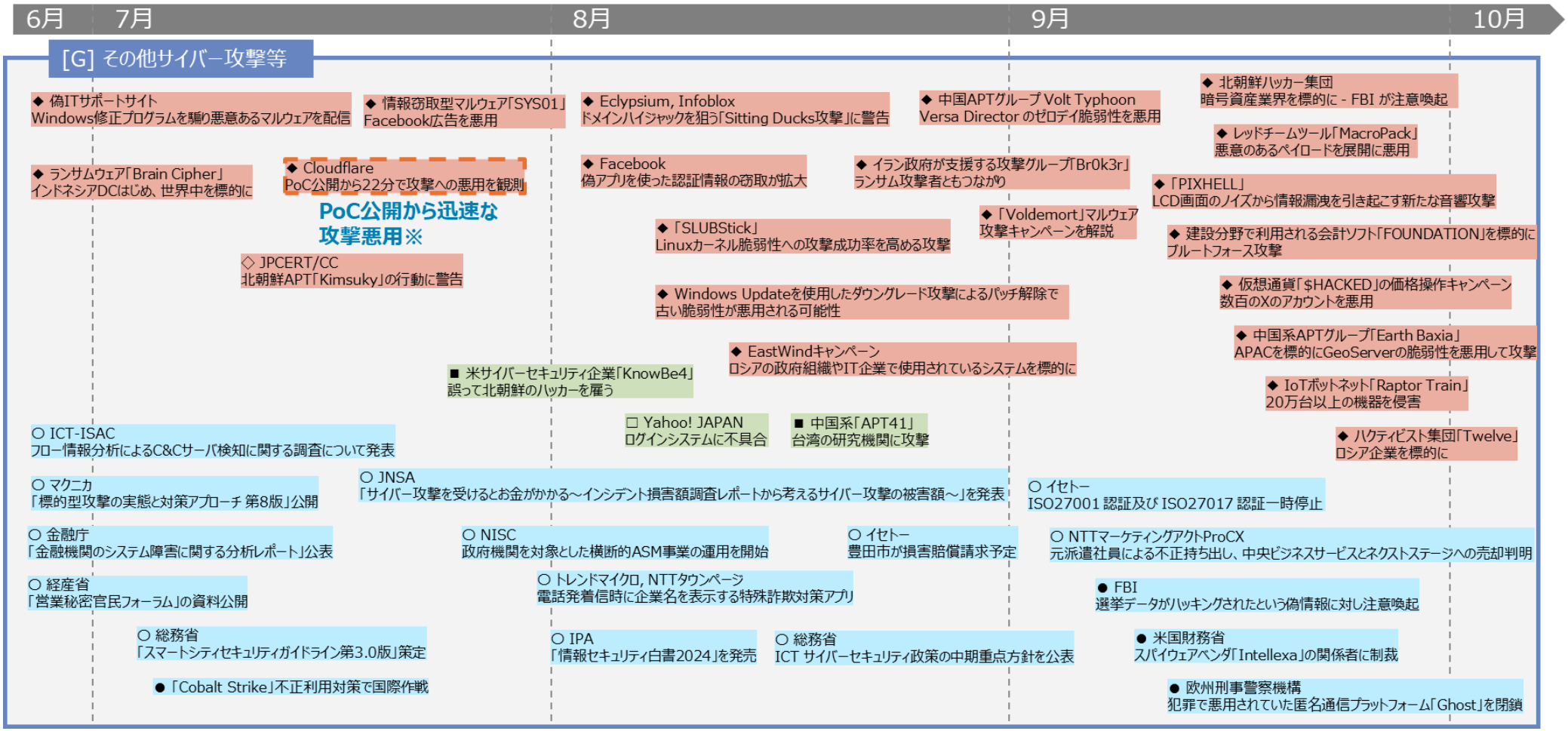


※タイムラインに記載している日付は事象発生日ではなく、記事掲載日の場合があります。

△□◇○:国内
▲■◆●:世界共通・国外

△▲:脆弱性
□■:事件・事故

◇◆:脅威
○●:対策



パスワードリスト攻撃/ブルートフォース攻撃

成りすましログインやアカウントの乗っ取り被害に関するトピックが、過去と比べて増加しています。[E]不正アクセスで取り上げたゴルフダイジェスト・オンラインやHuluの事例では、個人のアカウントが狙われました。[G]その他サイバー攻撃等で取り上げたFOUNDATIONの事例では、企業で使用している共用アカウントが狙われました。

IPAの調査 [29]にもあるように、推測しにくいパスワードを利用しているユーザは、パソコン利用者で71.3%、スマホ利用者で65.1%と高い割合です。よって、パスワードをデフォルトのままにしたり、あるいは推測しやすいパスワードを設定したりすることが原因で、ブルートフォース攻撃により不正ログインが発生している割合は減っていると推測します。しかし、複数サービスでパスワードを使いまわしていないユーザは、パソコン利用者で58.1%、スマホ利用者で46.6%と、上記と比較して少なくなっています。つまり、複数サービスでパスワードを使いまわしているユーザは依然として多いため、あるサービスから流出したIDとパスワードを悪用したパスワードリスト攻撃が成功した確率が高いと推測します。

パスワードは使い回さず、異なるサービスでは異なるパスワードを設定しましょう。パスワードを覚えることが大変であれば、パスワード管理ツールの活用がおすすめです。また、もしパスワードを突破されてしまった際の対策として、多要素認証の利用が有効です。適切なパスワード設定を用いて、パスワードリスト攻撃やブルートフォース攻撃からアカウントを守りましょう。

ゼロデイ攻撃の加速

[A]攻撃利用脆弱性のタイムラインから分かるように、各製品やサービスの脆弱性を悪用したサイバー攻撃が多く発生しています。こうした状況の中で、Cloudflareは、アプリケーションセキュリティレポート [30]にて、ゼロデイ攻撃の増加と、公開された脆弱性（CVE）を攻撃する武器の開発期間の短縮について警告しています。特にJetBrains TeamCityの認証バイパスの脆弱性（CVE-2024-27198）の事例では、公開後22分でサイバー攻撃が確認されました。

このスピードは、人間がWAFルールを作成したり、パッチを作成してデプロイしたりするスピードよりも速いです。つまり、サイバー攻撃の発生スピードに、人間が実施する防御策のスピードが追いつかなくなってきています。

各組織は、こうしたサイバー攻撃の発生スピードに対応できる方法を導入する必要があります。例えば、Cloudflareが取り組んでいるように、AIを活用していくことも一つの手でしょう。

委託先のランサムウェア被害

委託先や子会社でランサムウェア被害が発生して、委託元が情報漏えい被害を公表するケースが多く発生しました。特にイセトーやヒロケイで発生したランサムウェア感染被害の影響を受けた委託元が、個人情報漏えいの被害を公表するケースを多く観測しました。

独立行政法人情報処理推進機構（IPA）も「情報セキュリティ10大脅威 2024 [組織]」 [1]の第2位に「サプライチェーンの弱点を悪用した攻撃」を挙げています。

自組織のセキュリティ対策を強固にしたとしても、サプライチェーンの組織の中に適切な対策を行っていない組織があると、その組織がランサムウェア攻撃を受けて、そこから被害が広がります。

このような被害を予防するためには、ISO/IEC 27001等のセキュリティ対策に関連する国際的な認証を取得している委託先を選定することです。併せて、ランサムウェア攻撃の被害が発生した場合に備えて、契約で責任範囲を明確にすることやインシデント発生時の緊急連絡プロセスを明確にすることも重要です。

生成AIを利用したビジネスメール詐欺脅威の増大

近年、ChatGPTをはじめとする生成AIツールの普及に伴い、サイバー攻撃、特に世界中でビジネスメール詐欺（BEC）が広がり、手口も急速に高度化しています。Proofpoint [30]によると、生成AIの多言語対応を背景に、日本におけるBECは、前年比で35%増加と世界で最も高い増加率でした。攻撃者は、生成AIの多言語対応や文法的な精度、標的に合わせたメッセージ生成能力を使って、詐欺メールに特有の不自然な文法や誤字脱字をほぼ排除し、巧妙な文章を作成できます。

BECの被害を防ぐために、最低でもSPF、DKIM、DMACといったメールの送信元アドレスのドメインの正当性やメール本文改ざんを確認する仕組みを導入しましょう。しかし、最近の高度なBECは、BECの攻撃メールをSPFなどのメールセキュリティ対策で排除されないように、メールの送信元アドレスのドメインを詐称しないメールでBECを行います。そのため、メールセキュリティ対策の仕組みだけでなく、これまでも行なってきた従業員全体への定期的なBECのセキュリティ教育やトレーニングが重要になってきました。緊急の経費の支払い指示や緊急で機密情報を要求する文面などの最新のBECのパタンや手口を共有して、従業員がBECを自律的に識別して報告できるようにしましょう。

参考文献

[1] “情報セキュリティ10大脅威 2024,” IPA独立行政法人 情報処理推進機構, 24 1 2024. [オンライン]. Available: https://www.ipa.go.jp/security/10threats/10threats2024.html.

[2] “内部不正はなぜ起こるのか？～その発生のメカニズムを探る～,” Trend Micro, 5 12 2024. [オンライン]. Available: https://www.trendmicro.com/ja_jp/jp-security/24/l/expertview-20241205-01.html.

[3] “内部不正のプロが解説！これを読めば内部不正のすべてが分かる,” ジュピターテクノロジー株式会社, 31 5 2021. [オンライン]. Available: https://blog.jtc-i.co.jp/2021/05/ekran-5.html.

[4] “組織における内部不正防止対策,” IPA 独立行政法人 情報処理推進機構, 22 5 2015. [オンライン]. Available: https://sccs-jp.org/archives/symposium19/wp-content/uploads/sites/11/4874105411851011.pdf.

[5] “組織における内部不正防止ガイドライン,” IPA 独立行政法人 情報処理推進機構, 6 4 2022. [オンライン]. Available: https://www.ipa.go.jp/security/guide/insider.html.

[6] “秘密情報の保護ハンドブック,” 経済産業省, 2 2016. [オンライン]. Available: https://www.meti.go.jp/policy/economy/chizai/chiteki/pdf/handbook/full.pdf.

[7] “不正リスクへの理解を深める ー「不正のトライアングルの活用」,” 日本システム監査人協会, 5 12 2018. [オンライン]. Available: https://www.saaj.or.jp/kenkyu/pdf/238Shiryo.pdf.

[8] “独立行政法人情報処理推進機構,” サイバーセキュリティ対策・内部不正防止対策, 20 6 2022. [オンライン]. Available: https://www.meti.go.jp/policy/economy/chizai/chiteki/pdf/forum/reiwa4_forum/06_220620_IPA.pdf.

[9] “「企業の内部不正防止体制に関する実態調査」報告書,” 情報処理推進機構, 6 4 2023. [オンライン]. Available:

https://www.ipa.go.jp/security/reports/economics/ts-kanri/20230406.html.

[10] “従業員の不正を防ぐ！内部不正検知システムUEBA／XDR,” NTTデータ, 17 11 2021. [オンライン]. Available: https://www.nttdata.com/jp/ja/trends/data-insight/2021/1117/.

[11] A. Technologies, “Akamai 脅威レポート,” 7 8 2024. [オンライン]. Available: https://www.akamai.com/ja/newsroom/press-release/web-attacks-against-apis-and-applications-in-asia-pacific-grew-last-year.

[12] IBM, “IBM Report: Half of Breached Organizations Unwilling to Increase Security Spend Despite Soaring Breach Costs,” 24 7 2023. [オンライン]. Available: https://newsroom.ibm.com/2023-07-24-IBM-Report-Half-of-Breached-Organizations-Unwilling-to-Increase-Security-Spend-Despite-Soaring-Breach-Costs.

[13] Trendmicro, “HUNTING THREATS ON TWITTER - How social media can be used to gather actionable threat intelligence,” 30 July 2019. [オンライン]. Available: https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/hunting-threats-on-twitter.

[14] 後藤大地, “Facebookにゼロクリックの脆弱性、アカウント乗っ取りの危険,” 2 3 2024. [オンライン]. Available: https://news.mynavi.jp/techplus/article/20240302-2896170/.

[15] 竹内薫, “DeepSeekがデータ不正利用か　OpenAIとMicrosoft調査,” 日経新聞社, 29 1 2025. [オンライン]. Available: https://www.nikkei.com/article/DGXZQOGN293P40Z20C25A1000000/.

[16] T. O. W. A. S. P. (OWASP), “OWASP API Security Top 10,” 2023. [オンライン]. Available: https://owasp.org/API-Security/editions/2023/en/0x00-header/.

[17] M. Tucci, “セキュアな設計で API の安全性を確保する（原題：Defend Your APIs: Secure by Design）,” 8 5 2024. [オンライン]. Available: https://www.xlsoft.com/jp/blog/blog/2024/05/08/smartbear-33-post-63110/.

[18] A. L. a. 2. m. Dionisio Zumerle, “Market Guide for API Protection,” Gartner, Inc., [オンライン]. Available: https://www.gartner.com/en/documents/5471595.

[19] LIBRUS株式会社, “APIセキュリティの全体像: 重要性から最新のベストプラクティスまで,” 7 3 2024. [オンライン]. Available: https://cybersecurity-jp.com/contents/librussc/202/#API-5.

[20] アスピック, “APIプラットフォームの比較14選。違いや選び方は？,” 8 1 2025. [オンライン]. Available: https://www.aspicjapan.org/asu/article/29892.

[21] S. Abbasi, “2024 Midyear Threat Landscape Review | Qualys Security Blog,” Qualys, 06 08 2024. [オンライン]. Available: https://blog.qualys.com/vulnerabilities-threat-research/2024/08/06/2024-midyear-threat-landscape-review#conclusion.

[22] Carnegie Mellon University, “GitHub - CERTCC/SSVC: Stakeholder-Specific Vulnerability Categorization,” Carnegie Mellon University, 28 09 2023. [オンライン]. Available: https://github.com/CERTCC/SSVC?tab=readme-ov-file.

[23] 菊. 美紀子, “セキュリティ脆弱性評価の新たな指標SSVCとは？ | DATA INSIGHT | NTTデータ - NTT DATA,” 株式会社NTTデータグループ, 19 01 2023. [オンライン]. Available: https://www.nttdata.com/jp/ja/trends/data-insight/2023/0119/.

[24] 独立行政法人 情報処理推進機構 セキュリティセンター, “共通脆弱性評価システムCVSS v3概説 | 情報セキュリティ | IPA 独立行政法人 情報処理推進機構,” 独立行政法人 情報処理推進機構, 05 04 2022. [オンライン]. Available: https://www.ipa.go.jp/security/vuln/scap/cvssv3.html.

[25] NTT Communications, “CVSSを逆から読むと？脆弱性対応の意思決定に使えるSSVCについて - NTT Communications Engineers' Blog,” NTT Communications, 14 12 2024. [オンライン]. Available: https://engineers.ntt.com/entry/202412-ssvc/entry.

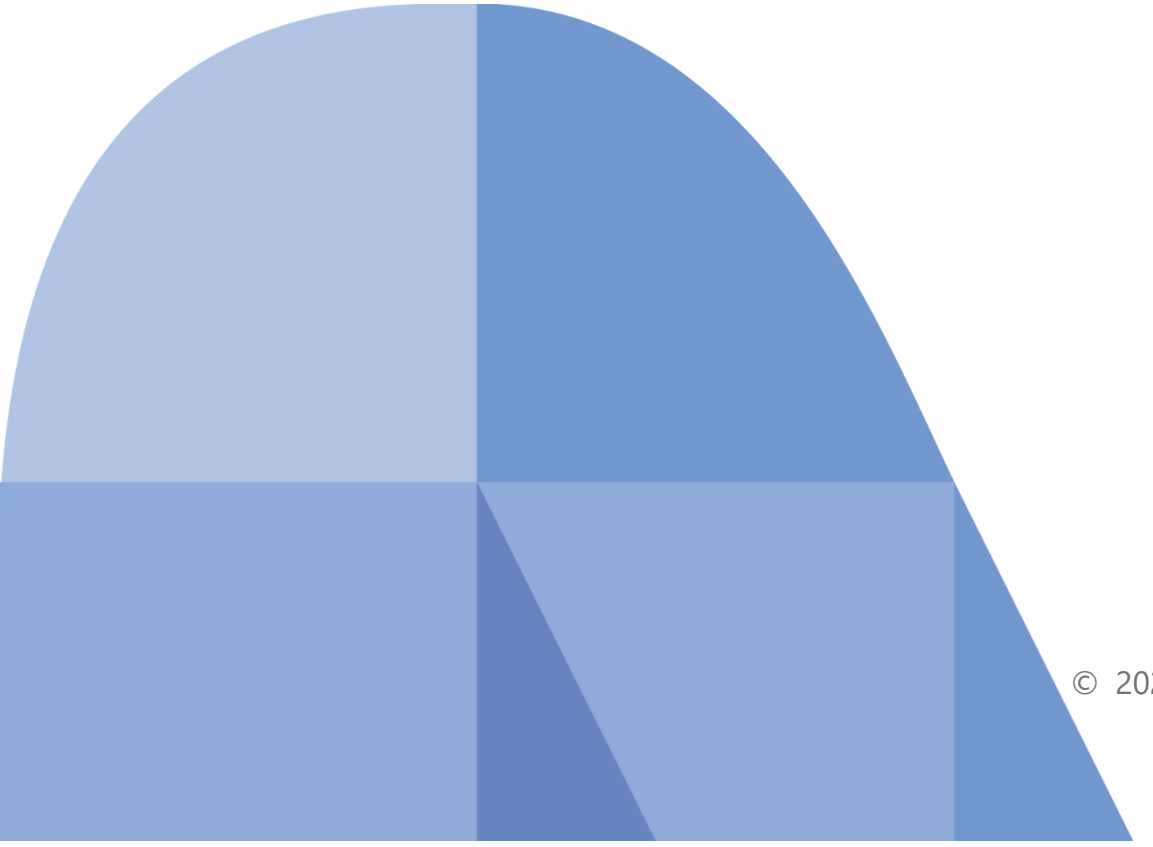
[26] 兼. 翼, “SSVC v2.1徹底解説：脆弱性対応フレームワークの変更点と進化 | FutureVuls Blog,” フューチャー株式会社, 21 11 2024. [オンライン]. Available: https://vuls.biz/blog/articles/20241121a/.

[27] 松. 翔. 村上 純一, “ユーザー企業におけるSSVCの導入と留意点 | PwC Japanグループ,” PwCコンサルティング合同会社, 09 03 2023. [オンライン]. Available: https://www.pwc.com/jp/ja/knowledge/column/awareness-cyber-security/ssvc-introduction.html.

[28] B. Jogi, “regreSSHion: Remote Unauthenticated Code Execution Vulnerability in OpenSSH server | Qualys Security Blog,” Qualys, 01 07 2024. [オンライン]. Available: https://blog.qualys.com/vulnerabilities-threat-research/2024/07/01/regresshion-remote-unauthenticated-code-execution-vulnerability-in-openssh-server.

[29] 独立行政法人情報処理推進機構, “2022年度 情報セキュリティの倫理と脅威に対する意識調査ー【脅威編】ー,” 2 2023. [オンライン]. Available: https://www.ipa.go.jp/security/reports/economics/hjuojm0000007fh1-att/000108321.pdf..

[30] Cloudflare, “Application Security report: 2024 update,” Cloudfr, 11 7 2024. [オンライン]. Available: https://blog.cloudflare.com/application-security-report-2024-update/.



2025年3月12日発行

（執筆）
栗本 重彰
板垣 毅
村田 直樹
寺師 悠平
田中 稜太郎

（編集者）
大嶋 真一
大谷 尚通
大串 智美
金澤 瑠維
中山 知香
澤田 貴順

株式会社NTTデータグループ C&I技術部 情報セキュリティ推進室
nttdata-cert@kits.nttdata.co.jp