セキュリティアラート対応大全

~アラートに埋もれる脅威を見逃さないために~

Takumi Ishibashi
Swimlane, Inc. Customer Success Manager

7+ years experience of SOC Analyst for Business on Product Vendors
GNFA / GCTI / GREM / GCDA / GEIR / GXPN

D1-4 on Internet Week 2024

本資料/セッションの対象者層

- 普段、セキュリティ製品から発報される大量のアラートに苦慮されているセキュリティご担当者の皆様
- セキュリティ製品導入直後で、これからどんな監視・運用が必要になるのか知っておきたいご担当者の皆様
- セキュリティご担当者の苦労を理解したい皆様
- セキュリティ運用の現場に興味関心をお持ちの皆様
- SOC Analyst に相談事のある皆様

本資料の位置づけ

- セキュリティ監視のエントリー資料
  - SOC Analyst レベル的にはTier1 ~ Tier2 相当
- セキュリティの企画後、導入されたセキュリティ製品・ソリューションの監視をする際の参考資料になれば
- 製品導入後、次のような課題を更に解決していくべきであることには留意
  - 製品では見つからない脅威への対処（脅威ハンティング等）
  - 継続的な（セキュリティ）システムアーキテクチャのブラッシュアップ
  - 脆弱性管理
  - チームマネジメント（特に、スキルアップとバーンアウト対策）
  - etc…
- InternetWeek の精神に則り、本資料はベンダーフリーであり、そして特定の営利企業やその製品・サービスに対する勧奨・依存が含まれないように努めます

お品書き

1. 大まかなセキュリティの変遷 (5 min)
    a. ネットワーク
    b. + エンドポイント
    c. + アイデンティティ
2. 説明にあたっての前提知識 (10 min)
    a. True Positive と False Positive
    b. Cyber Kill chain
    c. MITRE ATT&CK
    d. 解析・対応のおおまかなステップ
3. 各製品の概要とアラート対応事例 (合計50 min、途中休憩あり)
    a. Network Security 製品全般
    b. Identity and Access Management
    c. Endpoint Detection and Response
    d. その他（Email / 脅威インテリジェンス / ASM / SSPM / 改ざん検知 / etc..）
4. 判断を誤らせる要素とその対策 (10 min)
    a. システムアーキテクチャの問題（ログ遅延 / 情報不足 / 頻発する誤検知 / etc…）
    b. オペレータや組織の問題（認知バイアス etc…）
5. まとめと質疑応答 (5 min)

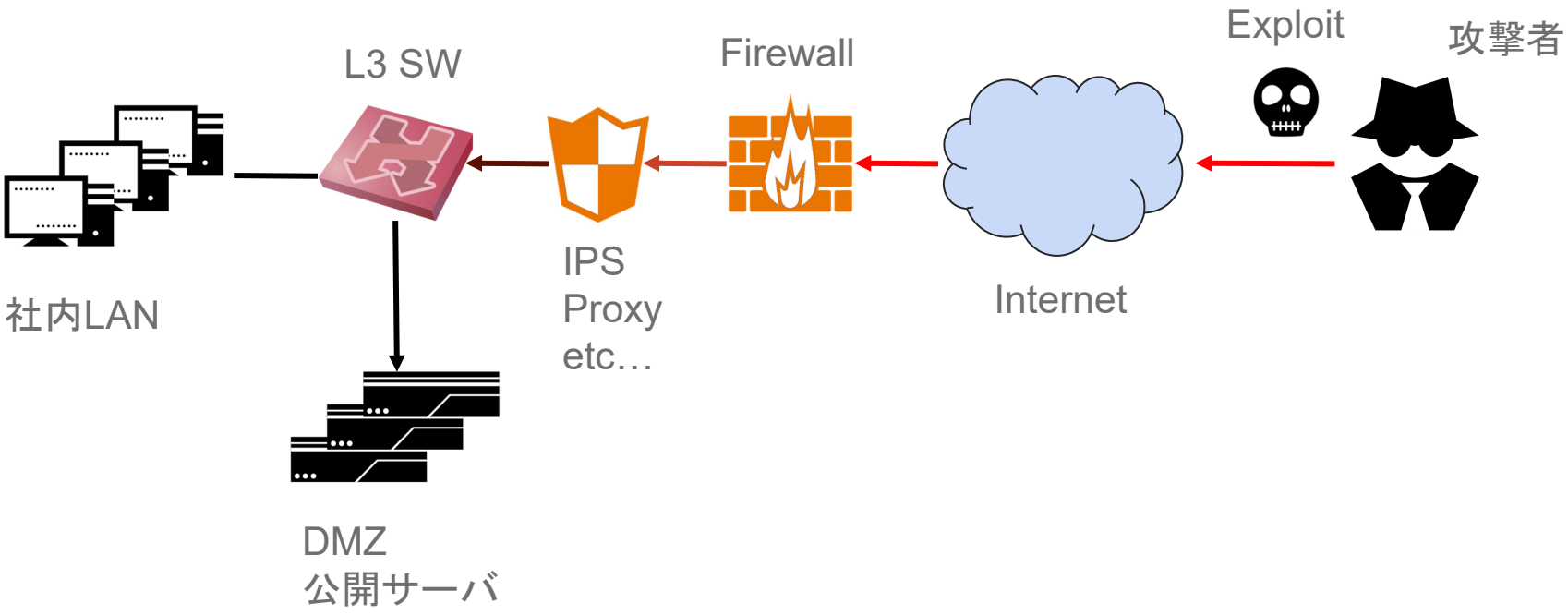
#1
大まかなセキュリティ監視の変遷

※
厳密には正確な歴史ではありません

a. ネットワーク
b. + エンドポイント
c. + アイデンティティ

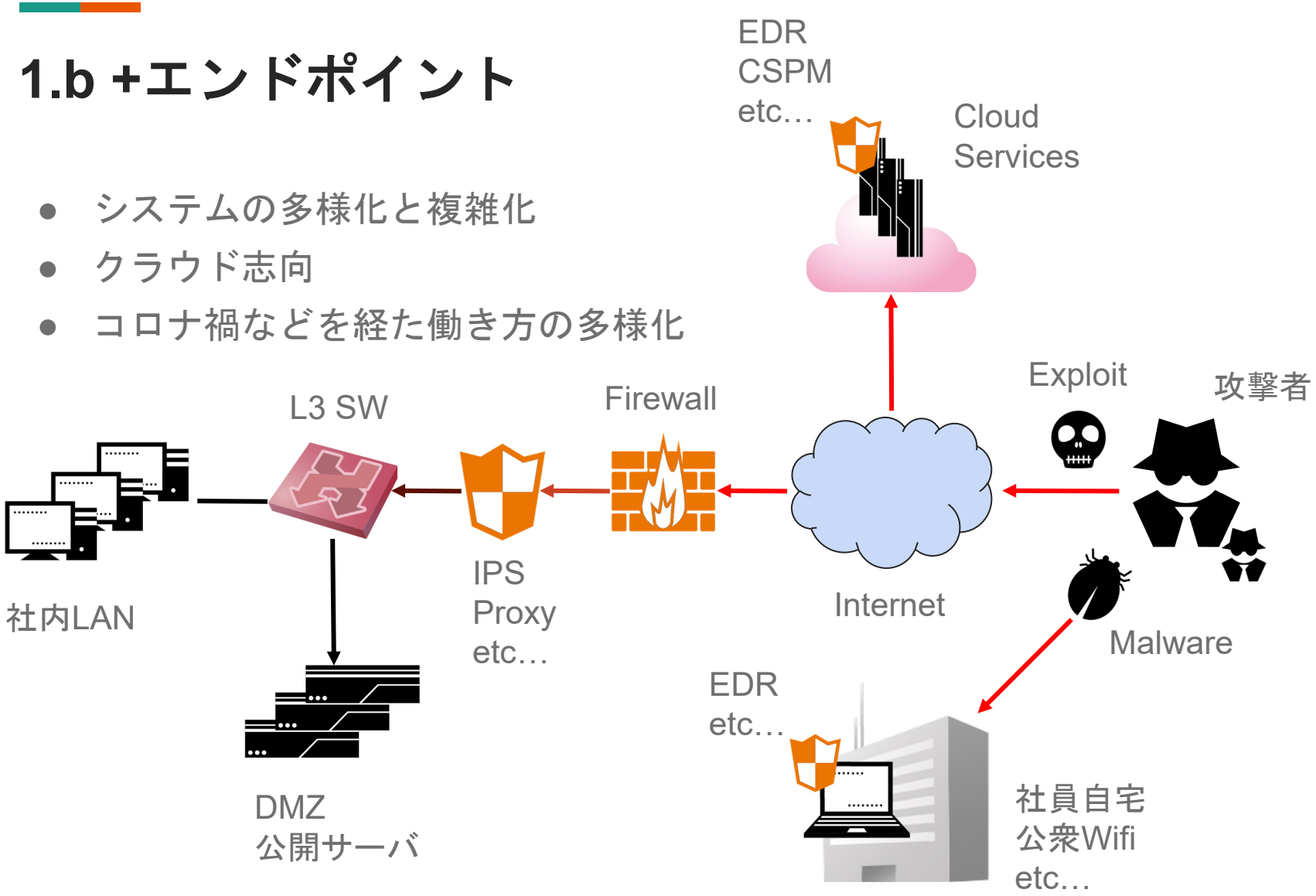
1.a ネットワーク

- セキュリティが話題に上がり始めた頃の原始的システムと攻撃者との関係性



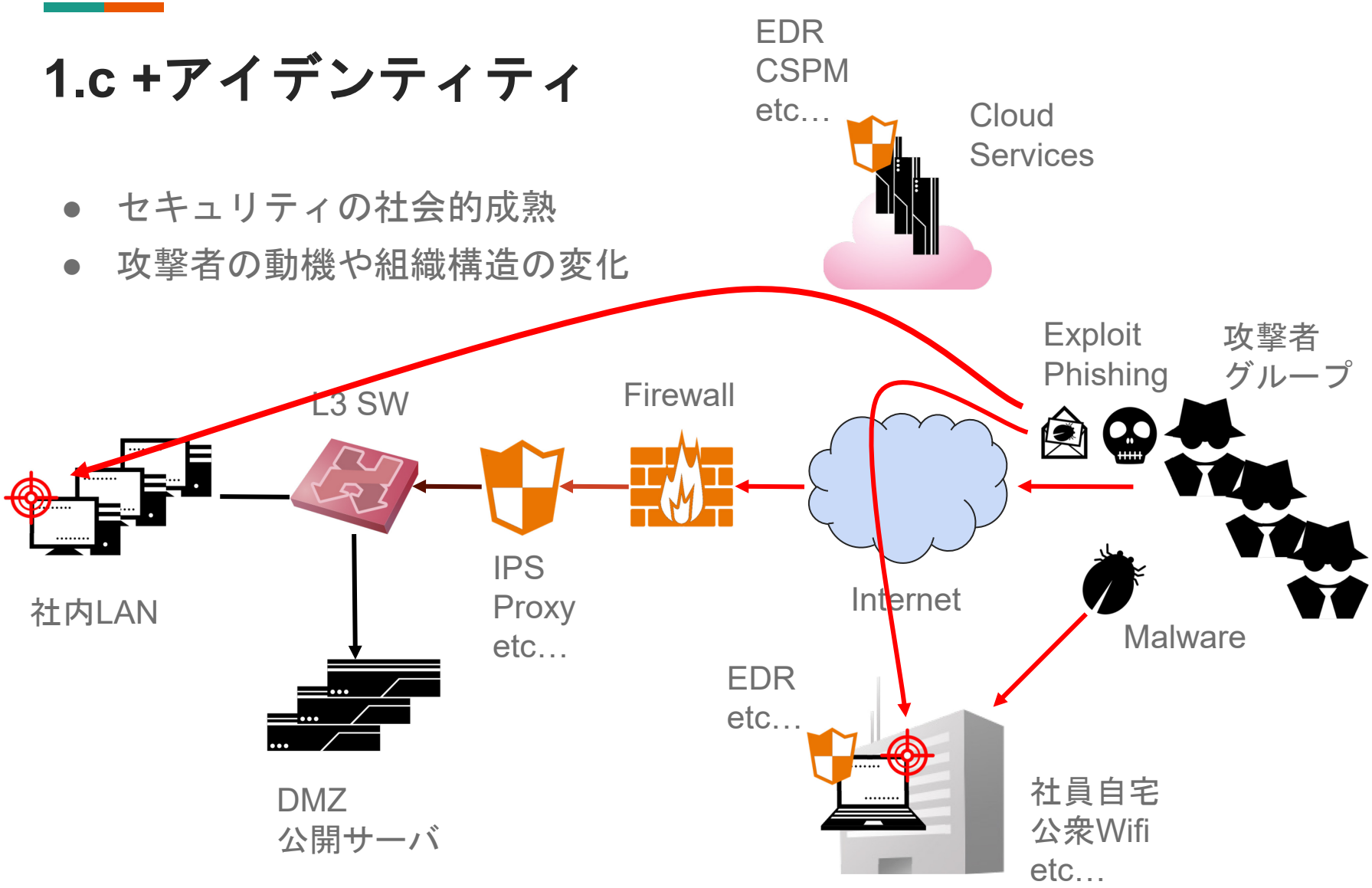
1.b +エンドポイント

- システムの多様化と複雑化
- クラウド志向
- コロナ禍などを経た働き方の多様化



1.c +アイデンティティ

- セキュリティの社会的成熟
- 攻撃者の動機や組織構造の変化



#2
説明にあたっての前提知識

a. True Positive と False Positive
b. Cyber Kill Chain
c. MITRE ATT&CK
d. 解析・対応のおおまかなステップ

2.a.1 True Positive と False Positive (1/4)

- セキュリティアラートの最も基本的な分類が（正）検知と誤検知
- 簡単な2値分類のように思うが、これがなかなか難しい
  - 問1 : 何に基づいて正と誤に分類するのか？
  - 問2 : 何のために正と誤を分類するのか？
  - 問3 : この分類問題に曖昧さは含まれないか？

2.a.2 True Positive と False Positive (2/4)

- 問1 : 何に基づいて正と誤に分けるか？
  - 例1 : 検知した事象・物に悪意がある
    - 一番馴染み深い分類手法
    - しかし、バックアップの削除であったりログの削除その他管理者業務に類似した事象の検知はどのように考えますか？
  - 例2 : 検知した事象・物が利用者（ユーザないしはその所属組織）の意図に反する
    - 主体によって判断がブレる
    - エンドユーザがフリーウェアや違法ソフトを使った場合は？
  - 例3 : 検知が、検知ルールの意図した通りのものかどうか
    - 検知ロジックの詳細を確認できる必要がある
    - ビジネスに必要なものが意図通りに検知された場合は？

2.a.3 True Positive と False Positive (3/4)

- 問2 : 何のために正と誤に分類するのか？
  - 例1 : 監視システムの効果測定のために正の検知数を測定
    - 検知数が多いほどそのシステムの効果が高いという考え方
    - この結果の過多によって利益が増減する人はいませんか？
  - 例2 : 検知ロジックの精度測定のために誤の検知率を測定
    - 誤検知率が高いほどそのロジックは精度が低いという考え方
    - マルウェア検知はこれでいいかもしれないが、管理者業務と攻撃者の振る舞いが一致してしまうようなものはどうする？
      - 前のスライドでも記載したバックアップ削除など
- 問3 : この分類問題に曖昧さは含まれないか？
  - 常に厳密な調査ができていればいいが、本当に？
  - 担当者に問い合わせても１ヶ月連絡がないなんて…ありませんか？

2.a.4 True Positive と False Positive (4/4)

- なぜこの話をしたか？
  - 解析にあたって、まず、悪意があるかとか、セキュリティベンダーがどう判断したかではなく、自組織内で起きてほしくないことはなにかを意識していただきたかった
  - この資料でも、基本的には、悪性有無ではなく業務意図の観点を重視

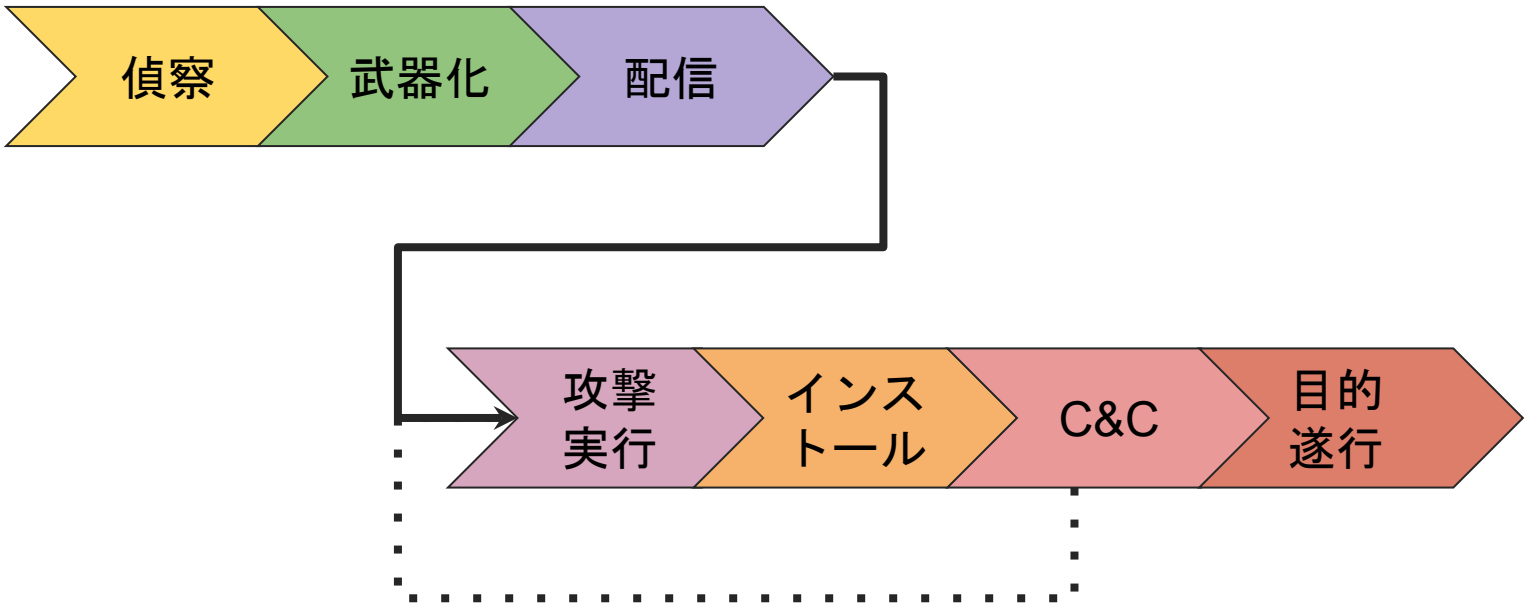
本当に True Positive と False Positive の分類が自組織に必要か
本当にその情報はその用途に適切か常に熟考してください
そしてその定義を関係者全員が共有しているか検証してください

- 個人的に推奨する分類例（というか、記録事項？）
  - 検知事象の組織への悪影響の有無とその度合い
  - 解析時点での検知内容の正確性・確度

2.b.1 Cyber Kill Chain (1/2)

- サイバー攻撃のライフサイクルモデル
  - https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html
- 攻撃者が攻撃を完了するまでの各ステップをモデル化したもの
  1. Reconnaissance : 偵察、脆弱性スキャン等
  2. Weaponization : 武器化、マルウェアやPoCの準備
  3. Delivery : 配信、フィッシングメールの送付等
  4. Exploitation : 攻撃実行、脆弱性が付かれて悪性コードが実行される等
  5. Installation : インストール、マルウェアやバックドアのインストール等
  6. Command and Control : 司令・制御、C2サーバとの通信
  7. Actions on Objectives : 目的の遂行、データの窃盗やシステム破壊
- 検知された事象が後半であればあるほど危険な状態

2.b.2 Cyber Kill Chain (2/2)



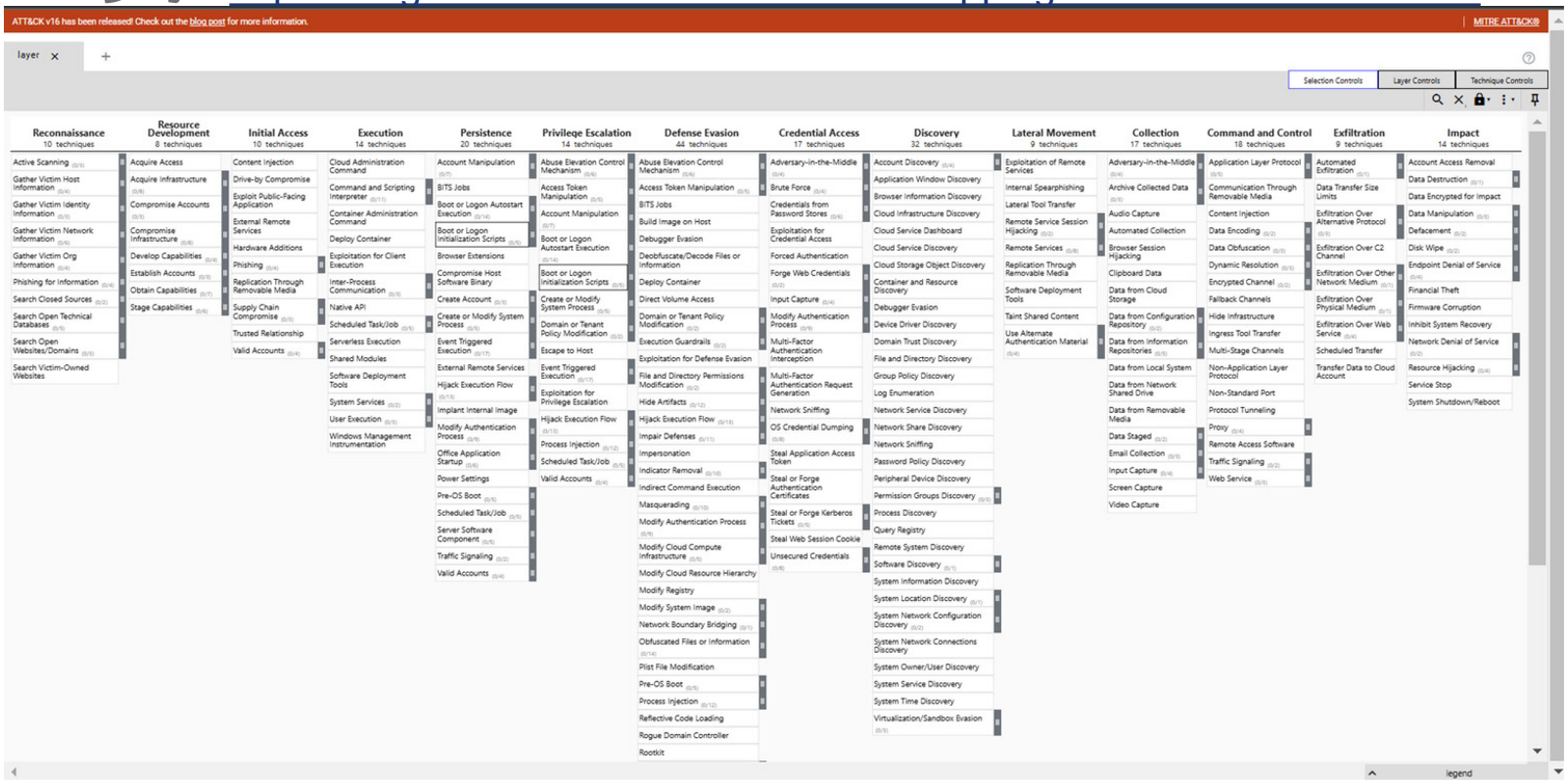
- 配信フェーズまでは（基本的に）インシデントの発生を意味しない
- マルウェアによっては攻撃実行からC&Cまでを複数回繰り返す
- 後半は確実に検知出来るようにしつつも、可能な限り前半で止める
- 簡潔で直感的に理解しやすいが攻撃の詳細を表現・共有するのは苦手

2.c.1 MITRE ATT&CK (1/2)

- MITRE によって作成された攻撃者の戦術・手法を分類・記述したフレームワーク
- Cyber Kill Chain との比較
  - 攻撃者の用いる手法の詳細まで明確に表現可能
  - 攻撃手法だけでなく、それらに対する防御手法についても提供されており個別の事案に対して実行すべきアクションが明確
  - 反面、詳細すぎるので情報量に圧倒されがち + 状況がわかりにくい
- こういうユースケースが便利
  - 自社システムにおけるセキュリティ対策の網羅性の確認
    - ツールの例 : DeTT&CT + MITRE ATT&CK Navigator
  - 検知があった際の検知目的や解析方法の確認
    - 今日の用途はこちら

2.c.2 MITRE ATT&CK (2/2)

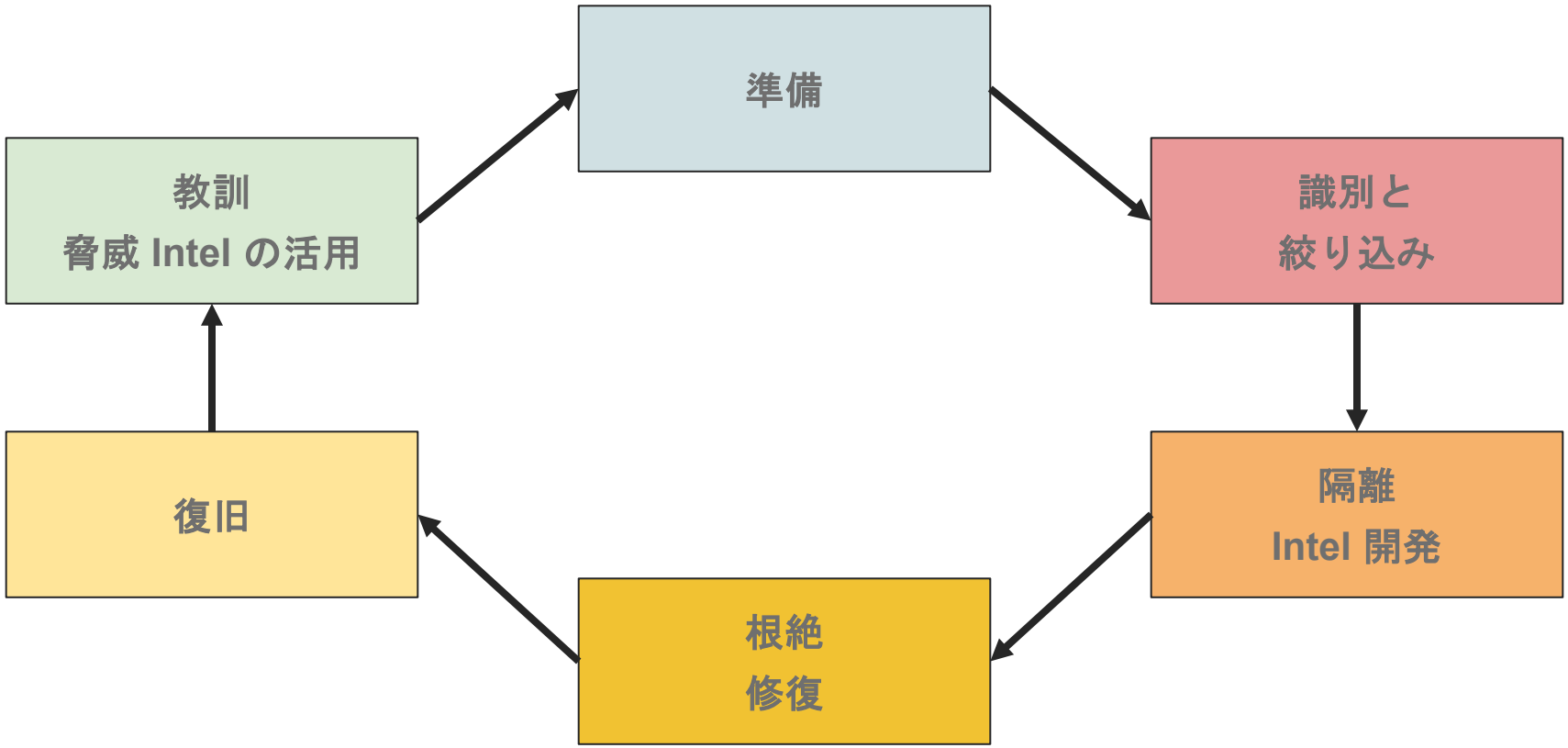
- 下記出典：https://mitre-attack.github.io/attack-navigator/
- 参考：https://blog.nviso.eu/2022/03/09/dettct-mapping-detection-to-mitre-attck/



2.d.1 解析・対応のおおまかなステップ (1/4)

- SANS® Institute によればインシデントレスポンスは次の6つのプロセスから構成
  1. Preparation : 準備
  2. Identification and Scoping : 識別と絞り込み
  3. Containment / Intelligence Development : 隔離 / Intel 開発
     - Intelligence Development の意訳 : 情報収集・分析の手順等の開発
  4. Eradication / Remediation : 根絶 / 修復
  5. Recovery : 復旧
  6. Lessons Learned / Threat Intel Consumption : 教訓 / 脅威 Intel の活用
- 出典 :
  - SANS 関連コース資料 (FOR508, FOR608)
  - https://www.sans.org/white-papers/1516/ ※かなり古い

2.d.2 解析・対応のおおまかなステップ (2/4)



2.d.3 解析・対応のおおまかなステップ (3/4)

1. Preparation : 準備
    - インシデントレスポンスに向き合う上で最も理想的なスタート地点
    - システム的な準備だけではなく、体制的な準備も
2. Identification and Scoping : 識別と絞り込み
    - インシデント発生の疑いを受け状況の識別と調査対象の絞り込みを実施
    - 起点の例
        - セキュリティ機器のアラート
        - 法執行機関からの連絡
        - Abuse への通報
3. Containment / Intelligence Development : 隔離 / Intel 開発
    - 事態の悪化を防ぐため、攻撃者の能力を制限
        - マルウェア感染や侵入の疑いのあるホストのネットワーク隔離
        - 外部ホストとの通信速度制限
        - etc…

2.d.4 解析・対応のおおまかなステップ (4/4)

4. Eradication / Remediation : 根絶 / 修復
    - システムから攻撃者を完全に追放
    - ゴールとしては、攻撃者がシステムに入るためには新たな侵害を成功させなければいけない状態となること
    - やることは多く複雑
        - マルウェアの削除、永続化の削除、脆弱性の修正、etc…
5. Recovery : 復旧
    - インシデント発生前の運用に戻るためのすべてを実行
6. Lessons Learned / Threat Intel Consumption : 教訓 / 脅威 Intel の活用
    - 発生した事象を文書にまとめ、利害関係者間で共有・議論
    - その後、準備フェーズに戻っていく

#3
各製品の概要と
アラート対応事例

a. Network Security 製品全般
b. Identity and Access Management
c. Endpoint Detection and Response
d. その他（Email / 脅威インテリジェンス / ASM / SSPM / 改ざん検知 / etc..)

3.a.1 Network Security 製品例

以降、特に明示しない場合は TCP/IP モデルを基準とする

i. Intrusion Detection System / Intrusion Prevention System
   - 主にインターネット層以上のシグネチャ（パターン）検知

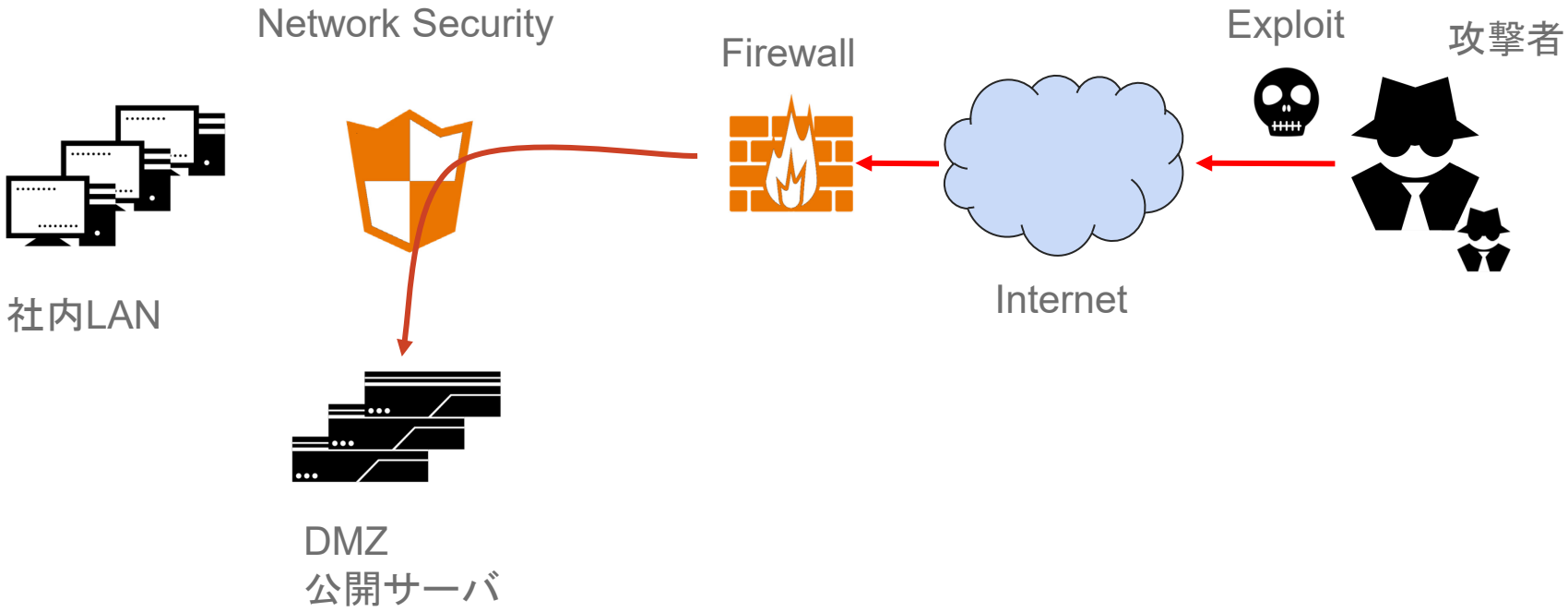
i. Secure Web Gateway / Proxy
   - アプリケーション層のシグネチャ（パターン）検知
   - レピュテーション検知

i. Network Detection and Response / Network logs with SIEM
   - レピュテーション検知
   - 統計的検知
   - 機械学習（分類）的検知

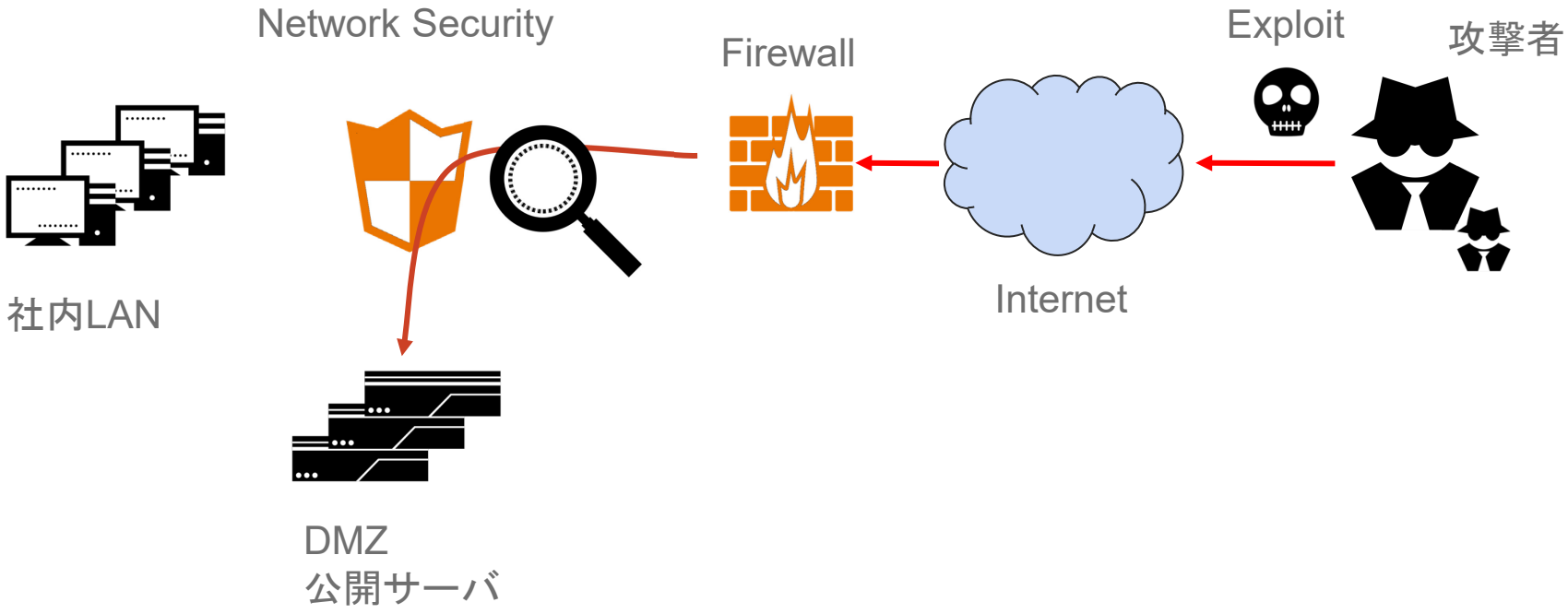
3.a.2 Network Security 製品 の基本的原則

- Network Security 製品のアラート解析において重要な点は次の3つ
  - どんなロジックによって検知が起きたか
  - 検知されたパケットが流れた場所
  - 検知されたパケットが流れる方向
- パケットが流れた場所と方向に基づいて次の3種類にアラートを分類することが可能
  - インバウンドアラート
  - アウトバウンドアラート
  - インターナルアラート

3.a.3 インバウンドアラートの概要図



3.a.3 インバウンドアラートの概要図



3.a.4 インバウンドアラートの概要

- インバウンドアラートは外部から内部システムに対するリクエストパケットに含まれるデータや通信頻度に基づいて検知されているケースが多い
- これらの検知はシグネチャの説明・種類によって次に分類可能
  - 何らかの脆弱性の悪用を意図した攻撃
  - 通常許容されない程に高頻度の通信
    - ブルートフォース攻撃などの不正な認証要求
    - DDoS
  - 発生が好ましくない特殊な用途の通信の発生
    - 外部からの RDP 接続

3.a.5 インバウンドアラートのケース別対応方法 (1/3)

1. 何らかの脆弱性の悪用を意図した攻撃
    a. Identification のため、以下のいずれか簡単に把握できるものを確認
        - 当該リクエスト・パケットがブロックされているか否か
        - 宛先ホストが当該脆弱性の影響を受けるか否か
        - 宛先ホストからのレスポンスが攻撃の成功を意味するか否か
    b. インシデントの発生が懸念される場合には次のような Containment / Eradication のためのアクションを実行
        - 送信元や攻撃パケットに含まれる攻撃者の IP / ドメインをブロック
        - （可能な場合は）一時的な宛先ホストのネットワーク隔離や影響を受ける可能性のあるサービスの停止
        - パッチ適用や IPS のシグネチャのモード切り替え (検知->防御)
    c. 誤検知である場合にはシグネチャと IP アドレスによる検知除外を検討

3.a.6 インバウンドアラートのケース別対応方法 (2/3)

2. 通常許容されない程に高頻度の通信
    a. Identification のため、以下を確認
        - その通信は1件たりとも発生が許されないものであるか否か
        - 事象発生時と属性（曜日、平日/祝日、時刻）が等価である時のデータと比較し統計的に発生が予期されるものか否か
        - 事象発生に関連しうるビジネス上のイベントがないか
            - 例えば重要度の高いプレスリリースの公開など
    b. インシデントの発生が懸念される場合、このケースでは根本的対処が難しいため次のような緩和策 (Mitigation) を検討
        - プロバイダーやCDN、各クラウドのDDoS対策サービスを検討
        - 通信経路上のボトルネックやシステムの用途に合わせた Rate Limit を設定
    c. 誤検知である場合には閾値の調整などを検討

3.a.7 余談: DDoS対策

- 仮に、本当にシステムが DDoS 攻撃を受けているならば、それを解決するためには次の点に注意
  - 攻撃を受けた時点で CDN を使用していないならば、CDN を使うと共にサーバの IP アドレスを変更しなければならない
    - 変更しない場合、攻撃者は CDN ではなく直接サーバにパケットを送ることが出来るため、CDNによる防御が機能しない
    - https://www.cloudflare.com/ja-jp/learning/cdn/glossary/origin-server/
- システムが AWS や Google Cloud 上に構築されているならば、それらのサービスで DDoS 対策ソリューションがあるので下記等を参照
  - https://cloud.google.com/armor/docs/advanced-network-ddos?hl=ja
  - https://docs.aws.amazon.com/waf/latest/developerguide/ddos-responding.html

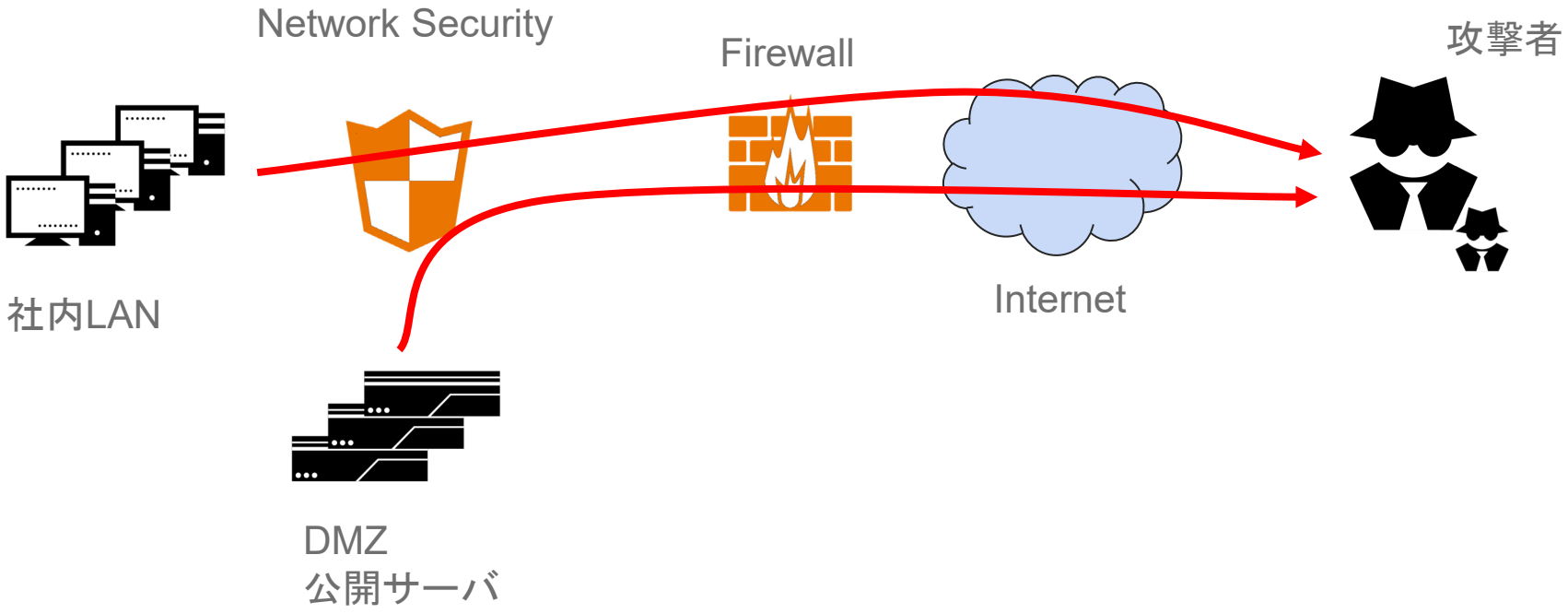
3.a.8 インバウンドアラートのケース別対応方法 (3/3)

3. 発生が好ましくない特殊な用途の通信の発生
    a. Identification のため、以下を確認
        - その通信は業務上必要か
        - 送信元アドレスやAS番号が関連する業務を担う組織のものか
        - 当該通信はいつもと同じ日時に発生したか
    b. インシデントの発生が懸念される場合には次のような Containment / Eradication のためのアクションを実行
        - 該当する通信セッションの強制切断
        - FWによる当該ポートの遮断
    c. 誤検知である場合には特定条件の検知のSOARによる自動クローズや検知除外を検討

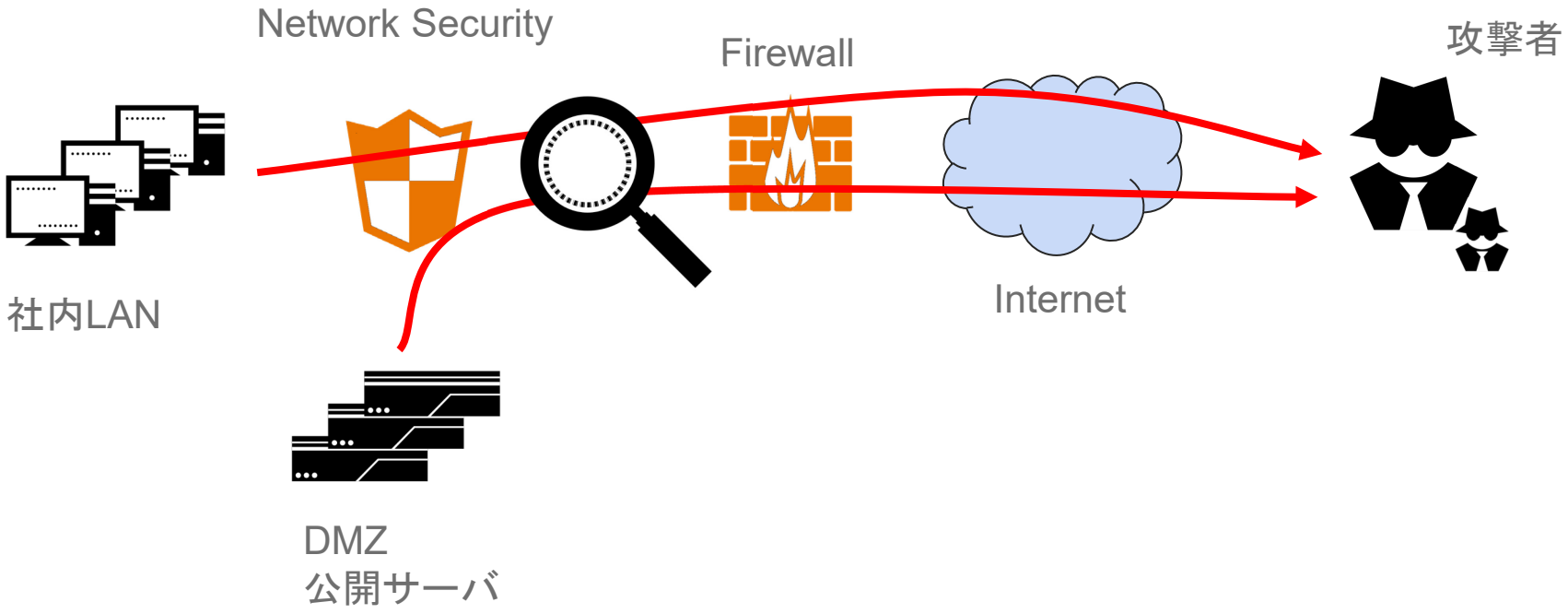
3.a.9 インバウンドアラートの検知・誤検知例

- HTTP プロトコルのクエリストリングやリクエストボディ上の特殊文字列などを検知するルール
  - 検知する攻撃例 :
    1. SQL インジェクション
    2. OS コマンドインジェクション
  - 誤検知例 :
    1. 一部の特殊文字列が含まれている場合
    2. SQL を用いた検索システムの途中経路に IPS が入っている

3.a.10 アウトバウンドアラートの概要図



3.a.11 アウトバウンドアラートの概要図



3.a.12 アウトバウンドアラートの概要

- アウトバウンドアラートは通信先レピュテーションや
  発生した通信と既知の悪性通信との類似性に基づいて検知している
  ことが多い
- これらの検知はシグネチャの説明・種類によって次に分類可能
  - 通信先が悪性
    - ただし悪性にも色んな定義が存在
    - フィッシング？C2サーバ？悪性広告？違法配布サイト？etc
  - パケットのヘッダ等が既知のマルウェアの通信手法と一致
  - 予期されない通信の発生
    - 瞬間的に大きなトラフィック
    - 異常に長いセッション時間
    - 外部への能動的な通信が意図されていないホストからの通信

3.a.13 アウトバウンドアラートのケース別対応方法 (1/3)

1. 通信先が悪性
    a. Identification のため、以下を確認
        - その通信先はなぜ悪性と判定されているのか？またその根拠は？
        - 通信元はなぜその通信を発生させたのか？何が通信をしているのか？
            - メールセキュリティや FW 製品が自身のブロック機能のために通信先の Verification を行っている場合がある
    b. インシデントの発生が懸念される場合には次のような Containment / Eradication のためのアクションを実行
        - 該当する通信セッションの強制切断
        - （マルウェア感染の場合は）EDR や FW で当該ホストの隔離
        - FWによる当該ポートの遮断
        - 業務外の用途に会社支給品を使わないようユーザ教育
    c. 誤検知である場合には送信元を基準とした検知除外や自動クローズを検討

3.a.14 アウトバウンドアラートのケース別対応方法 (2/3)

2. パケットのヘッダ等が既知のマルウェアの通信手法と一致
    a. Identification のため、以下を確認
        - （可能であれば）検知ロジックの詳細を確認
        - 実際に発生した通信のパケットキャプチャを確認
        - 宛先ホストのレピュテーション、送信元ホストの業務上の特性、通信を発生させたプロセスの情報を確認
    b. インシデントの発生が懸念される場合には次のような Containment / Eradication のためのアクションを実行
        - 前ページの対応に準ずる
    c. 結構誤検知が発生しうるが、このケースでは誤検知の見極めが難しく、検知ロジックを直接修正できないと検知抑止も出来ないケースが多い
        - このような場合は、思い切って検知ロジックを無効化する判断も重要

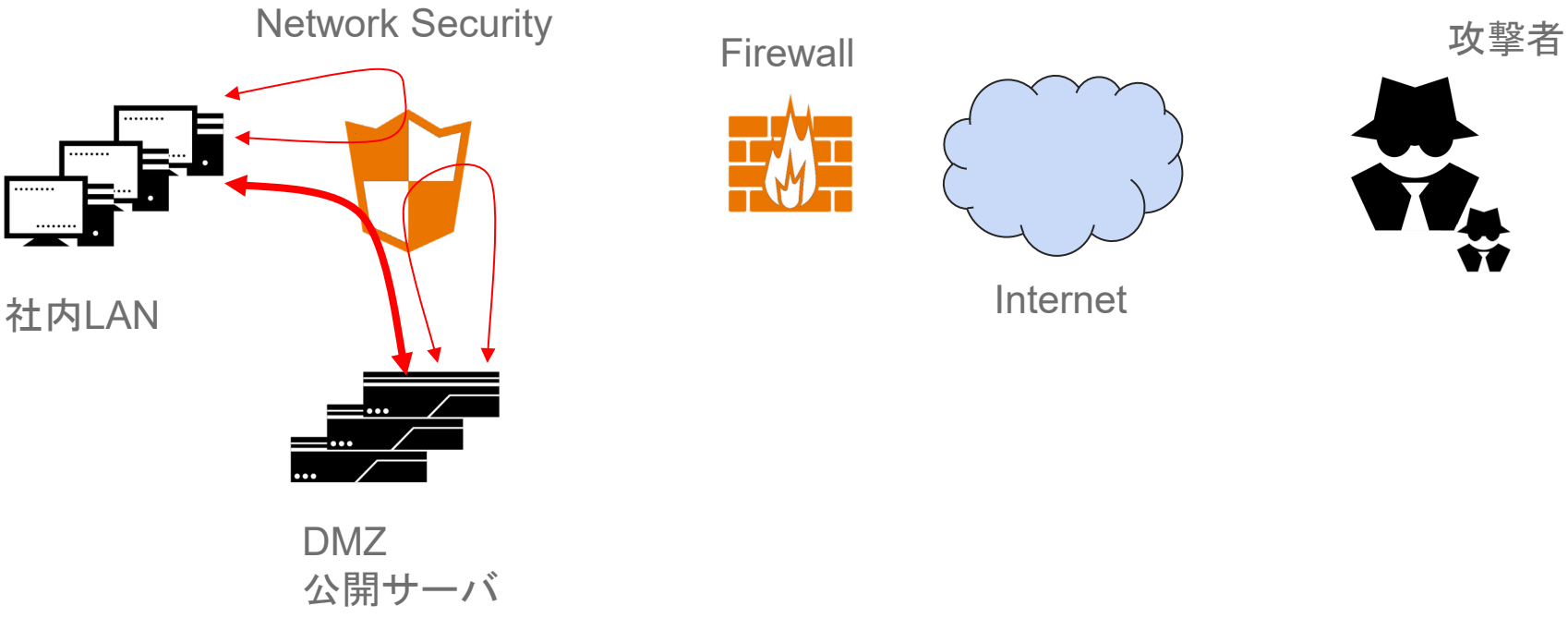
3.a.15 アウトバウンドアラートのケース別対応方法 (3/3)

3. 予期されない通信の発生
    a. Identification のため、以下を確認
        - 発生した通信に業務上の正当性・意図は存在するか
        - 発生した通信はプロトコルの仕様に準拠したものであるか
        - 通信を発生させたホスト・プロセスがその通信を発生させることに妥当性はあるか
            - 例 : Web 会議ソフトはセッション時間が長く、トラフィックが大きくなりがち
    b. インシデントの発生が懸念される場合には次のような Containment / Eradication のためのアクションを実行
        - 前ページの対応に準ずる
    c. 誤検知が多数発生しうるが、このケースの解析では自組織の様々な業務への理解を深め、それらを文章化することで1つずつ検知ルールの見直しを図っていくしかない

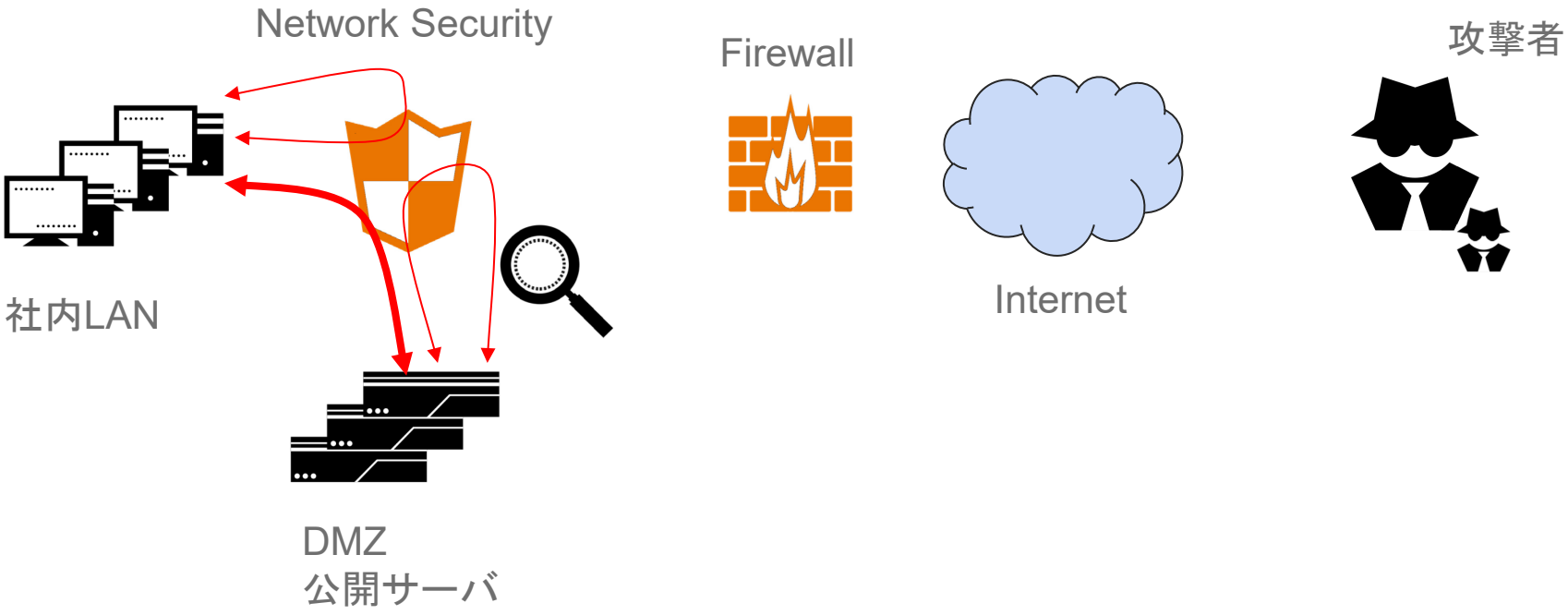
3.a.16 アウトバウンドアラートの検知・誤検知例

- マルウェア XXX の C2 サーバに対する通信
  - 検知シナリオ :
    - スクリプト実行等からの最初のマルウェアダウンロード
    - マルウェア感染後、追加のペイロード・マルウェアのダウンロード
    - マルウェア感染後の命令要求、ハートビート
  - 誤検知例 :
    - ーメールセキュリティの送信元アドレス検証の一環の通信
    - FQDN ACLなどのセキュリティ製品の検証の一環の通信

3.a.17 インターナルアラートの概要図



3.a.18 インターナルアラートの概要図



3.a.19 インターナルアラートの概要

- インターナルアラートは内部ホスト間の通信において異常な振る舞いを検出することを目的としていることが多い
- 性質上、アウトバウンドアラートと一部重複したルールがあり解析の観点でも近しいものがある
- これらの検知はシグネチャの説明・種類によって次に分類可能
  - 統計的に異常
  - パケットや通信の振る舞いのパターンが異常
    - 機械学習的検知も分類的にはここ
  - 通信の発生自体が異常

3.a.20 インターナルアラートのケース別対応方法 (1/3)

1. 統計的に異常
    a. Identification のため、以下を確認
        - その通信を発生させたホスト / アカウントについて、事象発生前後に通常利用時と異なるログイン等の振る舞いがないか
        - その統計データは学習初期段階ではないか
            - 学習初期段階の統計的検知は誤検知を多発させる
        - 送信元プロセス、通信プロトコルに関する既知のバグ報告等がないか
    b. インシデントの発生が懸念される場合には次のような Containment / Eradication のためのアクションを実行
        - EDR や FW で当該ホストの隔離、当該ホストの調査
        - 危険であると判断したならば認証情報の初期化を実行
    c. 誤検知である場合には検知感度の調整や学習期間を長く設けるなど
        - 学習期間の短いアラートは捨てる等

3.a.21 インターナルアラートのケース別対応方法 (2/3)

2. パケットや通信の振る舞いのパターンが異常
    a. Identification のため、以下を確認
        - 検知ロジックの詳細を確認し、何がどう異常なのかを理解
        - 実際に発生した通信のパケットキャプチャやログを確認し、検知されたと考えられるパケットパターンを特定
        - ポート番号等をもとに通信の妥当性を確認
    b. インシデントの発生が懸念される場合には次のような Containment / Eradication のためのアクションを実行
        - EDR や FW で当該ホストの隔離
    c. 誤検知が多数発生しうるが、これも誤検知の見極めは困難を極める
        - 業務への影響がないのであれば、通信をブロックしアラート通知は抑止するというのも一つの手
        - 通信をブロックするということをある程度社内に明示しておくのも重要

3.a.22 インターナルアラートのケース別対応方法 (3/3)

3. 通信の発生自体が異常
    a. Identification のため、以下を確認
        - 当該時刻に送信元ホストから宛先ホストに対するシステム管理者業務は予定されていないか
    b. インシデントの発生が懸念される場合には次のような Containment / Eradication のためのアクションを実行
        - EDR や FW で当該ホストの即時隔離
    c. 管理者業務が誤検知される場合があるが、これは業務の事前申告、および申告内容の監視チームとの連携が重要

3.a.23 インターナルアラートの検知・誤検知例

- マルウェア XXX による Lateral Movement (横展開)
  - 検知シナリオ :
    - あるホストのマルウェア感染後、内部スキャン
    - あるホストのマルウェア感染後、ファイル共有などを経由した感染拡大
    - あるホストのマルウェア感染後、ADサーバへのアクセスと認証情報の奪取
  - 誤検知例 :
    - 管理者業務全般
    - ペネトレーションテストなど（ある意味誤検知ではない）
    - 脆弱性のある古いプロトコルによる業務通信

3.a.24 製品分類再掲とアラートとのマッピング

i. Intrusion Detection System / Intrusion Prevention System
   - 主にインターネット層以上のシグネチャ（パターン）検知
   - 3種類のアラートを満遍なく発報
ii. Secure Web Gateway / Proxy
   - アプリケーション層のシグネチャ（パターン）検知
   - レピュテーション検知
   - アウトバウンドアラートを中心に発報
iii. Network Detection and Response / Network logs with SIEM
   - レピュテーション検知
   - 統計的検知
   - 機械学習（分類）的検知
   - 主にアウトバウンド・インターナルアラートを発報させるために使用
   - ブロックするには他製品との連携が必要

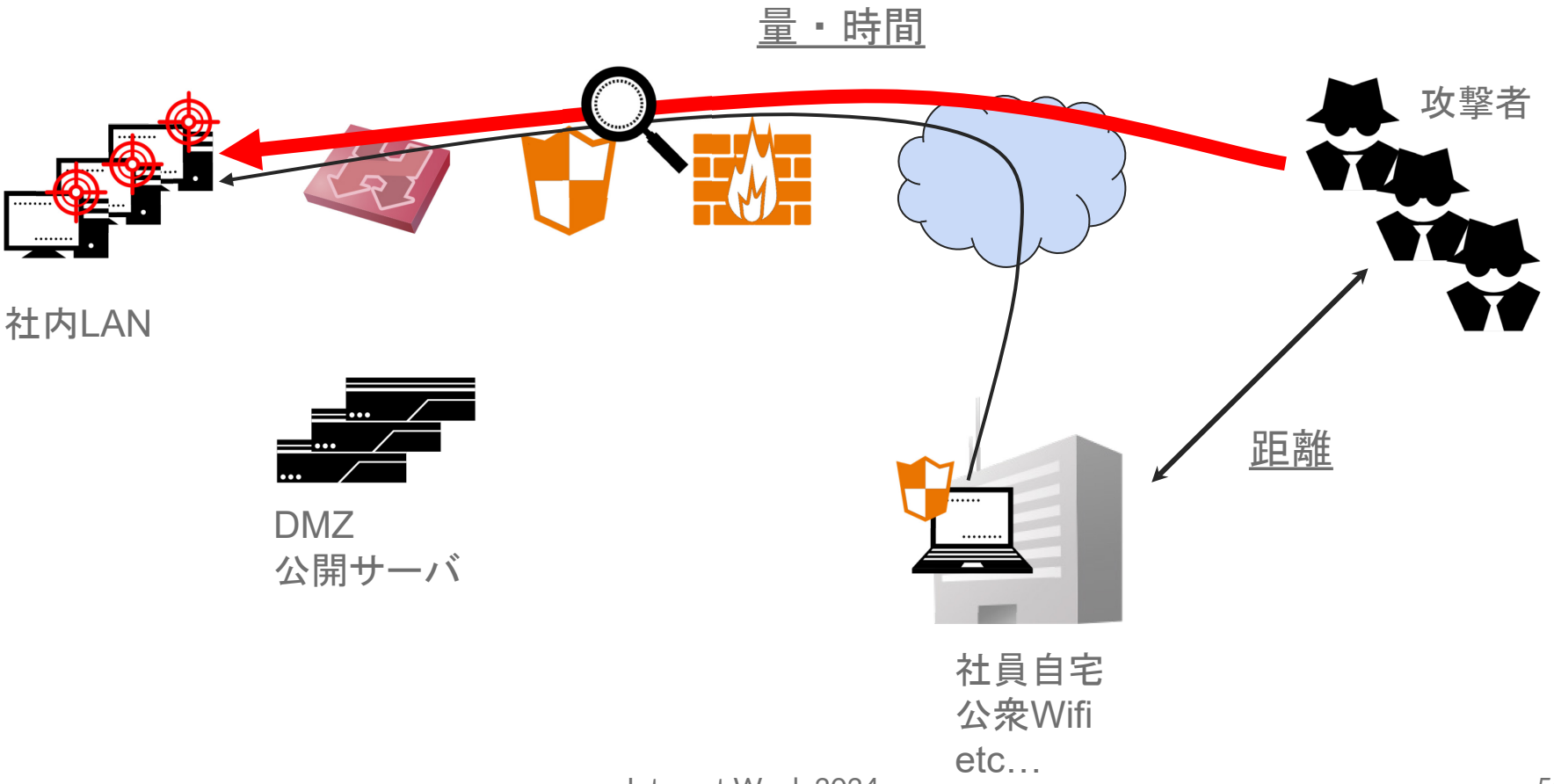
3.b.1 Identity and Access Management の基本的原則

- 解析の観点は今までの Network Security と同様ですが
  アイデンティティに着目していくつか補足できる部分を解説します
- IAMで想定されるアラートとしては次の通り
  - 不正ログイン
  - 不正なアカウント発行
  - 不必要な権限取得
- 重要な観点は
  - スケーラビリティ（アラートの数が増えても処理に時間がかからない）
  - 運用設計
  - 情報収集（業務理解、自社のシステムアーキテクチャの理解）

3.b.2 不正ログインの種類

- アイデンティティ周りで最も触れる機会の多いアラート
- 例えば次のようなものが不正ログインとしてアラート化される
  （アラート化することが望ましい）
  - 普段と異なる場所（Geo Location、あるいはIPアドレス）からログイン
  - 短期間のうちに明らかに遠く離れた2地点からのログイン
  - 普段と異なる端末からのログイン
  - 同一ホストから多数のアカウントへのログイン試行
  - 同一アカウントに対する多数のログイン試行

3.b.3 不正なログインのイメージ



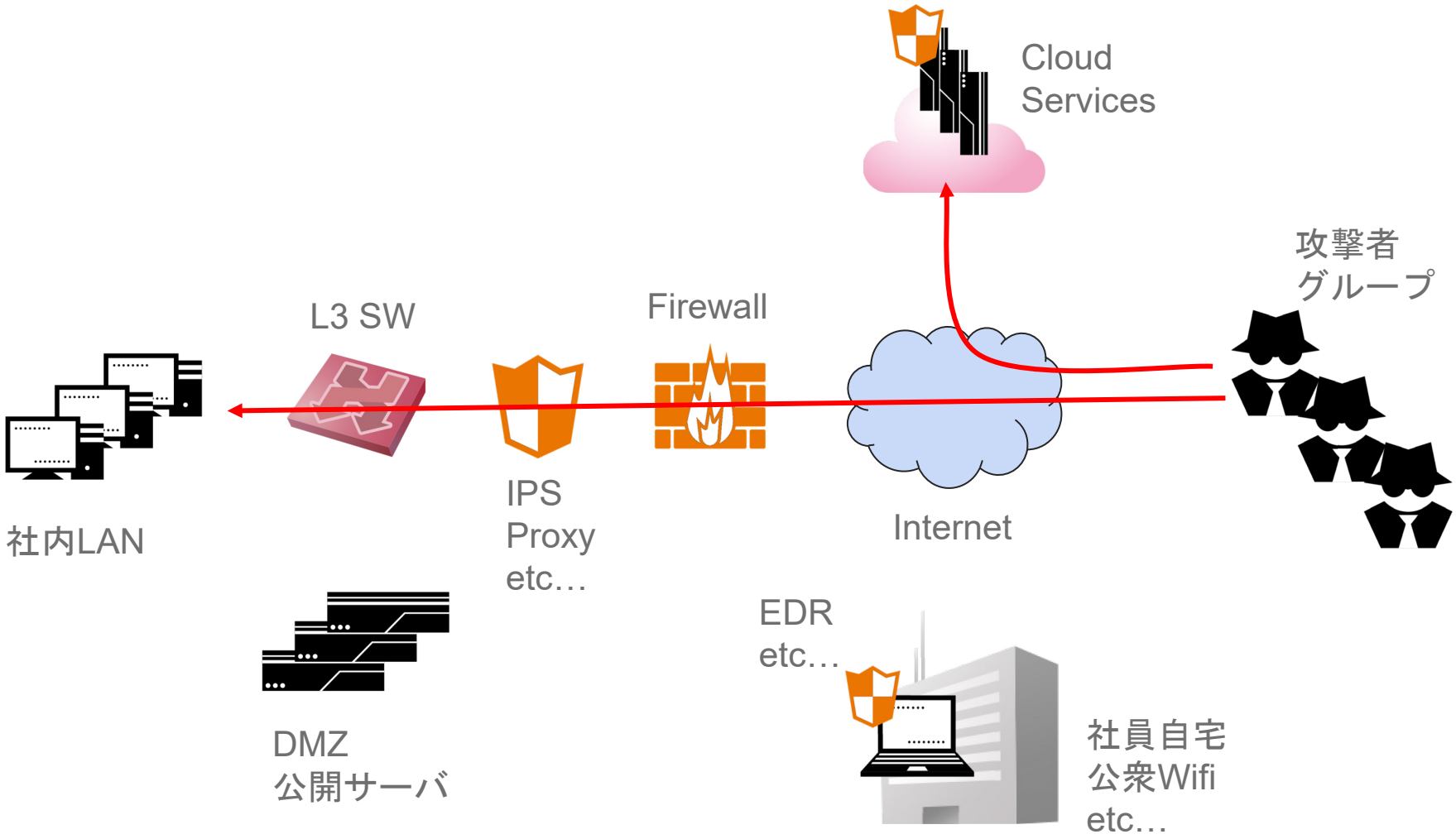
3.b.4 不正なログインに対する勘所

- ログイン失敗に対してはそこまで神経質になる必要はない
- 問題は前述のアラート内にてログイン成功が記録されている場合
- また、真に恐ろしいのはこのようなアラートの多発
- このようなケースに備えるため、自動化ソリューションを用いて不正ログインアラートと関連したログイン成功イベントだけを抽出するような仕組み作りをしておくことが重要
- また、効果的な制御を行うためにはリクエスト元を識別出来るようなログ設計も重要
    - 送信元IPアドレスやMACアドレス、ホスト名、その他端末情報など

3.b.5 不正なログインの誤検知例

- 社員の出張による想定外の国からのログイン
- Geo Location DB の情報が古い
- 認証情報リセット直後の認証失敗の増加
- 社員がやたらとパスワードうち間違えてる
  - 10回くらいは極稀に間違える人はいる
  - けれど、一回ロックアウトして電話とかで解除とかの対応にした方が無難

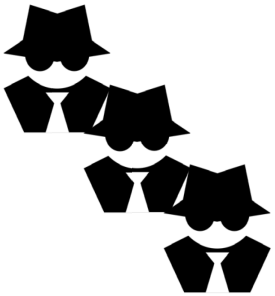
3.b.6 不正なアカウント発行のイメージ (1/2)



3.b.7 不正なアカウント発行のイメージ (2/2)

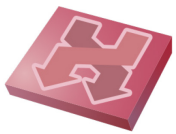
Cloud
Services

攻撃者
グループ



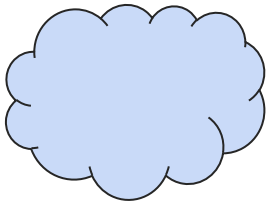
社内LAN

L3 SW



IPS
Proxy
etc…

Firewall



Internet

DMZ
公開サーバ

EDR
etc…

社員自宅
公衆Wifi
etc…

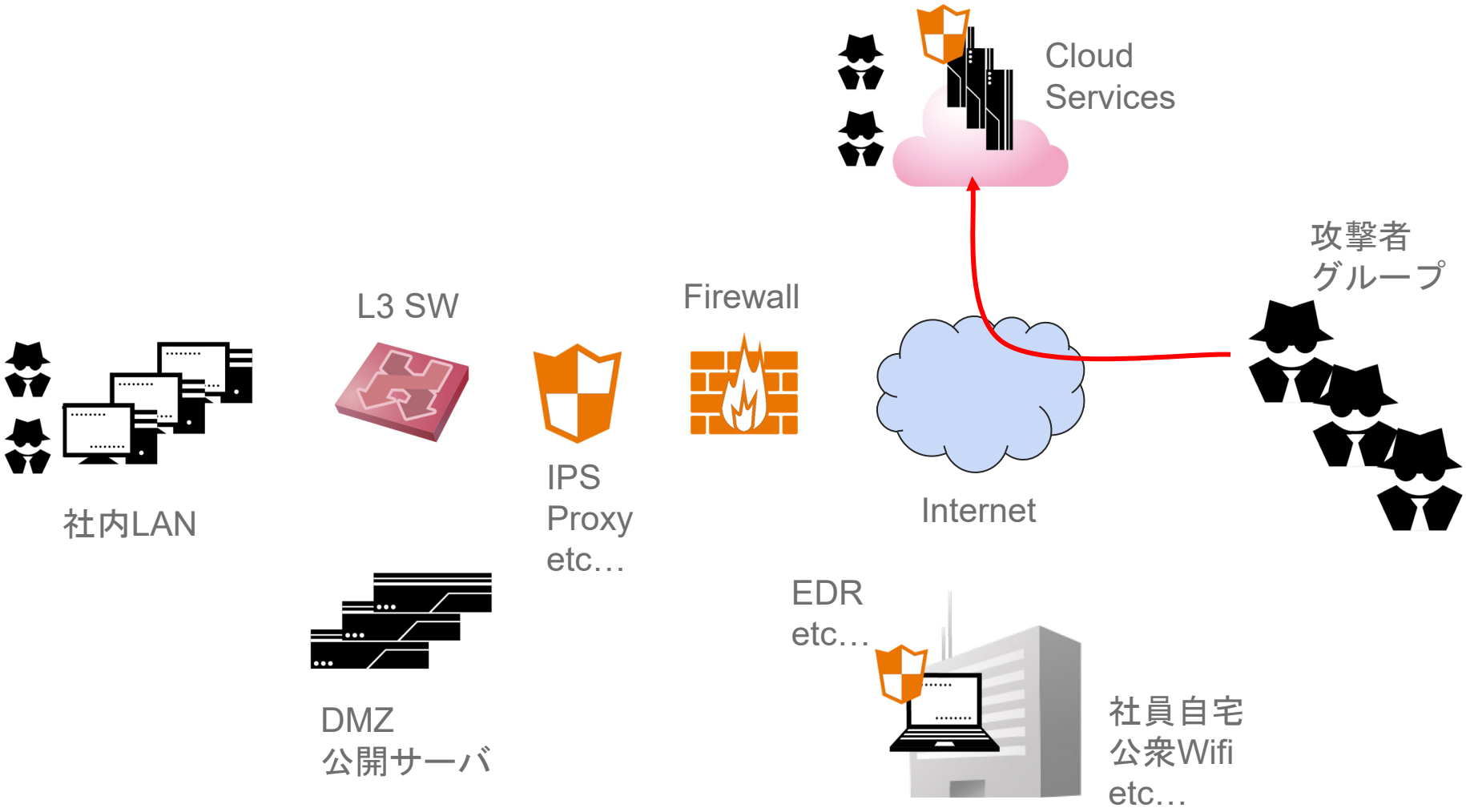
3.b.8 不正なアカウント発行に対する勘所

- もし不正なアカウント発行が事実であれば緊急事態である
- 基本的には侵入されているという前提で次の対応を
  - 発行されたアカウントの無効化
  - 発行したアカウントの無効化
  - 発行したアカウントのログイン端末が管理下のホストであれば当該ホストの隔離
    - その後マルウェア感染などを調査し侵入経路の特定
  - 管理下にないのであればIPアドレス等を用いて環境内からブロック

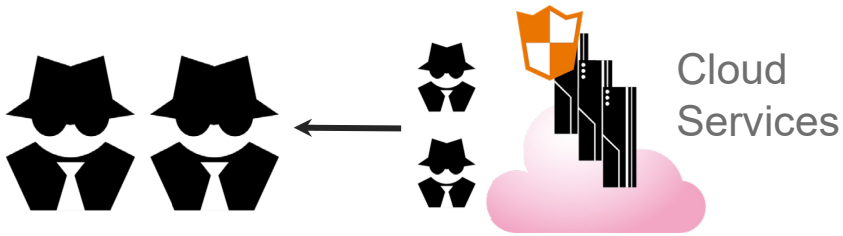
3.b.9 不正なアカウント発行の誤検知例

- 管理者によるアカウント追加
  - 社員の部署異動
  - 新規採用
    - 4月、9月は作業が増える
    - そういったときに誤検知にまぎれた本物を見逃さないように留意

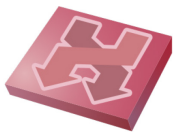
3.b.10 不必要な権限取得のイメージ (1/2)



3.b.11 不必要な権限取得のイメージ (2/2)



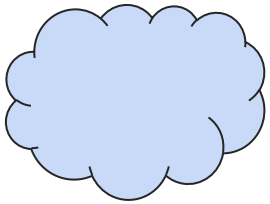
社内LAN



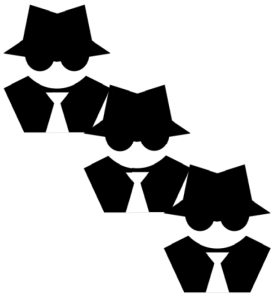
L3 SW

IPS
Proxy
etc…

Firewall



Internet



攻撃者
グループ

DMZ
公開サーバ

EDR
etc…

社員自宅
公衆Wifi
etc…

3.b.12 不必要な権限取得に対する勘所

- こちらは実運用しようとすると誤検知が多いかも
- 権限取得のケースとしてはアカウントの初期プロビジョニングや
  部署異動、新規サービス開発時などに多数の権限取得が必要か？
- それ以外で多数の権限取得が行われる場合をなんとか可視化したい
- 最小権限の原則とは言うが、それを実際に運用するためには
  権限申請用のポータルサイトや申請処理の自動化なども非常に重要

3.b.13 不正な権限取得の誤検知例

- 社員の昇進
  - 昇進に伴う権限付与およびその検知の除外を一連の手続きとして
    まとめて自動化できているとよい
- 社員への新規業務のアサイン
- 新事業開発、立ち上げ
  - ある程度柔軟で高速な物が必要
  - 自動化もいいが、ある程度他から隔離された環境をごっそり渡すとか
    出来るといいかも
    - 立ち上げ完了時にこういった環境はスクラップすべき

3.c.1 Endpoint Detection and Response の基本的原則

- EDRのアラート解析において重要な点は次の5つ
  - どのような検知ロジックで発報されたか
    - その検知ロジックに関連するログにはどのような値・文字列が記録されているか
  - そのプロセスの実行者は誰か
  - プロセスはどのように実行されたか
  - 実行されたプロセスは未知か既知か
  - 当該プロセス・ソフトウェアの業務上必要性の有無
- 実行の主体に着目すれば、発生した事象を次のように分類可能
  - ユーザの行動を切っ掛けにシステムが正常な処理を行った
  - ユーザの行動を切っ掛けにシステムが不正な処理を行った
  - ユーザが関わらずシステムが正常な処理を行った
  - ユーザが関わらずシステムが不正な処理を行った

3.c.2 EDR の検知ロジックについて

- 多くのセキュリティベンダーは、攻撃者への検知ロジック漏洩を恐れて検知ロジックを公開していない
- その代わりに、MITRE ATT&CK の Tactics (戦術) と Techniques (技術) は示してくれていることが多い
- これらのマッピングと検知説明を元に検知内容を読み解き、関連ログと照らし合わせてそのアラートが真に危険か否かを判断
- 関連ログの調べ方としては、製品にも依るが、検知されたプロセスのプロセスIDや時刻を起点とすることが多い

3.c.3 プロセスの実行者

- プロセスの実行者は誰かと言われて一番最初に確認するのは
  プロセスの実行ユーザ
  - ただし、その実行ユーザが”だれ”かまでは気にしていますか？
  - そのユーザはもしかしたら、ネットワーク経由で不正ログインに成功した攻撃者かもしれません
  - あるいはオフィスに侵入した産業スパイかも？
- 自分は実行ユーザの確認のために次の項目をよく確認します
  - ログインユーザ名
  - ログイン方法
    - https://learn.microsoft.com/ja-jp/windows-server/identity/securing-privileged-access/reference-tools-logon-types
  - 検知前後の実行プロセスや Read/Write したファイルの業務関連性

3.c.4 プロセスはどのように実行されたか

- プロセスの実行方法を判別する上で最も重要なのは親プロセスの情報である
  - 例えば、Windows 系 OS において Explorer.exe を親プロセスとしてプロセスが起動していればそのプロセスはユーザが Explorer からダブルクリックなどで実行したものと判断可能
  - 逆に outlook や Office ソフト系ソフトウェアを親プロセスとして実行プログラムが実行されるのは通常ではない
    - と、言いつつ、案外あるんですけどね…
    - Excel マクロに色々集めるのは EDR と相性が悪いよとだけ…
- 次スライドに著名な親プロセスを提示

3.c.5 著名な親プロセスとその子プロセスとの関係 - Windows 編

- explorer.exe の子プロセス : ユーザによる手動実行
- svchost.exe の子プロセス : タスクスケジューラに記録されたタスクの実行
- cmd.exe の子プロセス : コマンドプロンプトを経由して実行
- mshta.exe の子プロセス : hta ファイルのスクリプトによる実行を示すことが多い
- msiexec.exe の子プロセス : MSIインストーラーによるソフトウェアインストールの過程での実行
- powershell.exe の子プロセス : powershell スクリプト実行かあるいは powershell CLI 操作による実行
- etc…

3.c.6 著名な親プロセスとその子プロセスとの関係 - Linux 編

- cron.d の子プロセス : プログラムの自動実行
- sshd の子プロセス : 外部からの ssh セッションの元で実行されたコマンド等
- cron の子プロセス : 定期的なスケジュールタスク
- bash の子プロセス : CLI操作によって実行されたコマンド
- systemd の子プロセス : サービス等
  - ※サービス : bash などを経由せずシステムが直接起動するプロセスの総称
- python/perl の子プロセス : 該当するスクリプト言語の実行によって呼び出されたプロセス、親のコマンドラインにも注意
- etc…

3.c.7 著名な親プロセスとその子プロセスとの関係 - macOS 編

- launchd : Linux の systemd や cron 等の機能を包括するプロセス
- zsh (bash) の子プロセス : CLI操作によって実行されたコマンド
- osascript の子プロセス : AppleScript や JavaScript for Automation の実行によって呼び出されたプロセス
- python/perl の子プロセス : 該当するスクリプト言語の実行によって呼び出されたプロセス、親のコマンドラインにも注意
- etc…

3.c.8 実行されたプロセスは未知か既知か (1/2)

- 実行プロセスが未知か既知かで今後の解析の流れが変わる
- なお、自社システム内と Virus Total などの公衆環境とで意味合いが異なる
- 自社システム内における既知/未知：
  - 自社システム内で既知である（当該ハッシュなどが複数のホストに存在する）場合にはそれは業務上利用されている可能性を考慮すべき
  - 自社システム内において未知（単一ホスト上にしか存在しない）場合には業務上の利用であるとは想定しにくい

3.c.9 実行されたプロセスは未知か既知か (2/2)

- Virus Total などの公衆環境における既知/未知 :
  - 既知であればレピュテーション情報を更に確認するが、このとき検知されたソフトウェアの性質にも着目すべきである
    - 例えばCADソフトなど画像編集系ソフトは誤検知されやすい
    - Powershell など、それ単体は良性であっても、攻撃者によって悪意を持って使用されるケースも
      - コマンドライン文字列や、その他実行に際し社内ユーザなら普通やらないテクニック等に留意（検知ルールをよく読む）
        - 特殊記号を用いた難読化とか
      - よく悪用されるものリスト : https://lolbas-project.github.io/
  - 未知であれば内製ツールの可能性も考慮しつつ解析を続行

3.c.10 業務上必要性の有無

- 実行されたプロセスが業務上必要か否かは重要な観点である
- 極論を言えば、仮に不正な操作でなかったとしても、業務上必要性のないプロセスは止めてしまえばよい
  - もちろん、このような運用はユーザビリティの低下を招く
  - これを回避し、セキュリティとユーザビリティを担保するには、情報システム部門が実際の業務内容を深く理解し、社員のためにどのようなIT基盤が必要であるかを詳細に理解する必要がある
- 私がこの観点でよく調査していたのは次の通り
  - 検知ソフトウェア内と同一ディレクトリ上に存在する txt ファイル
  - 実行の結果起きうる現象
  - 各種 (読み/書き/実行) ファイル名と事業内容の整合性
  - ファイルサイズやアンチデバッグ機能など、同一ディレクトリ内の他ファイルを含め、不自然な点がないか

3.c.11 ユーザの行動を切っ掛けとした正常処理

- ユーザの操作とログがある程度一致するのがこの分類
  - explorer.exe からファイルを開く
  - ブラウザを起動する
  - etc…
- ただし、業務上どこまでの処理が正常として許容されるかは
  組織の定義・認識によっても異なりますので、自社の認識を言語化
  しておくことが重要
  例：
  - あなたの組織においてフリーソフトは利用が許可されていますか？
    - 許可されているとしたらどのライセンスがOKですか？基準は？
  - PUPは駆除対象ですか？容認しますか？

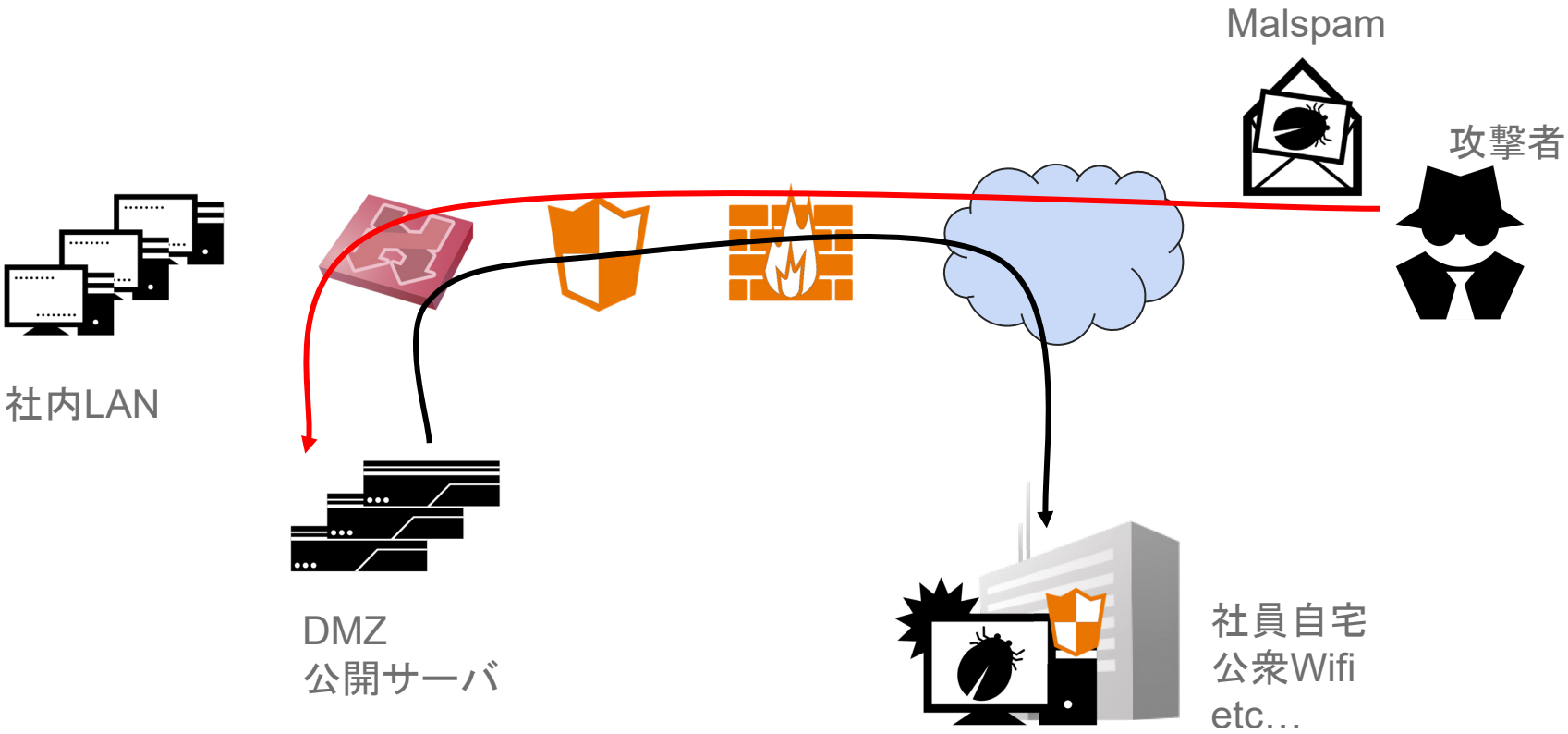
3.c.12 ユーザ行動の誤検知例

- パスワードを忘れて初期化
- 業務用Excel マクロの実行
- スクリプト言語を実行ファイルに変換したものの実行
- （ユーザの行動によって誘発されたプロセス実行に対する）セキュリティソフトによる干渉
  - 例えば Data Loss Prevention 系統

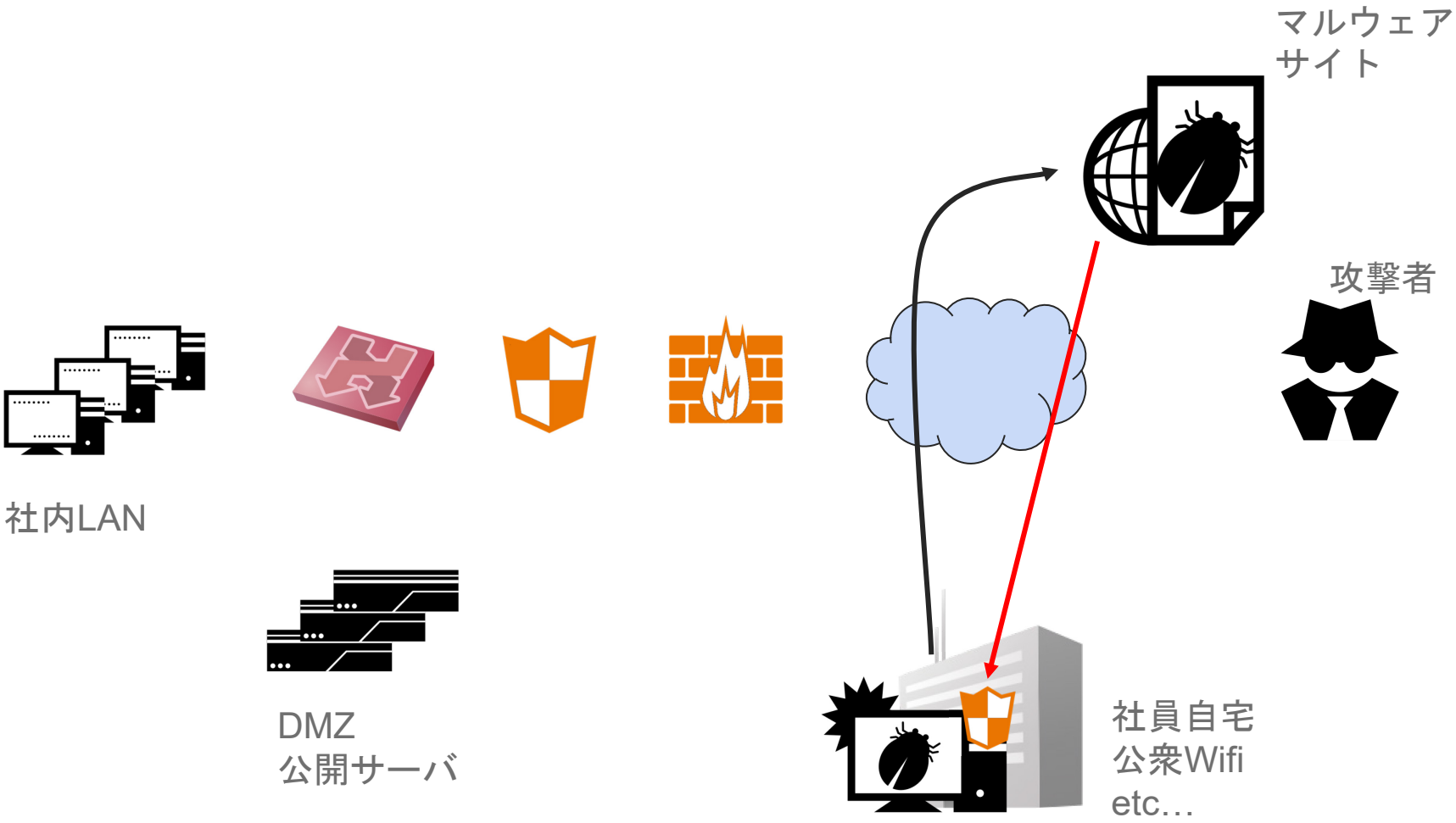
3.c.13 ユーザの行動を切っ掛けとした不正処理

- ユーザの行動が切っ掛けで不正な処理が行われる例が次の通り
  - マルウェアスパムの添付ファイルをクリックする
  - 不正スクリプトが仕掛けられたサイトにアクセスする
  - 無料ソフトウェアをダウンロードして実行する
  - ユーザが悪性広告等の指示に従い操作を行う
- このような場合、概ね親プロセスは次の通り
  - explorer.exe
  - オフィスソフトや PDF ソフト
  - ブラウザプロセス
- ユーザログインは基本的にインタラクティブ（対話的）
- このような場合はユーザへのヒアリングや、操作時刻以降の処理内容を追いかければ影響箇所の特定や修復は容易

3.c.14 ユーザ行動による不正処理イメージ (1/2)



3.c.15 ユーザ行動による不正処理イメージ (2/2)



3.c.16 ユーザの行動を切っ掛けとしない正常処理

- スタートアップやスケジュールされたタスクの実行によるプロセス
  - システムアップデート
  - スタートアップソフトウェアの実行
  - このような自動実行系の仕組みはユーザがコンピュータシステムをより便利に使うために重要であるが。。。
    - 攻撃者による Persistence (永続化)、つまり橋頭堡として悪用されうる
- その他
  - ネットワーク経由でのRDP等を用いたログイン
  - ホスト上で実行されているサービスが外部からリクエストを受けることで実行したプロセス
    - これらも、想定されたものであればよいのだが。。。

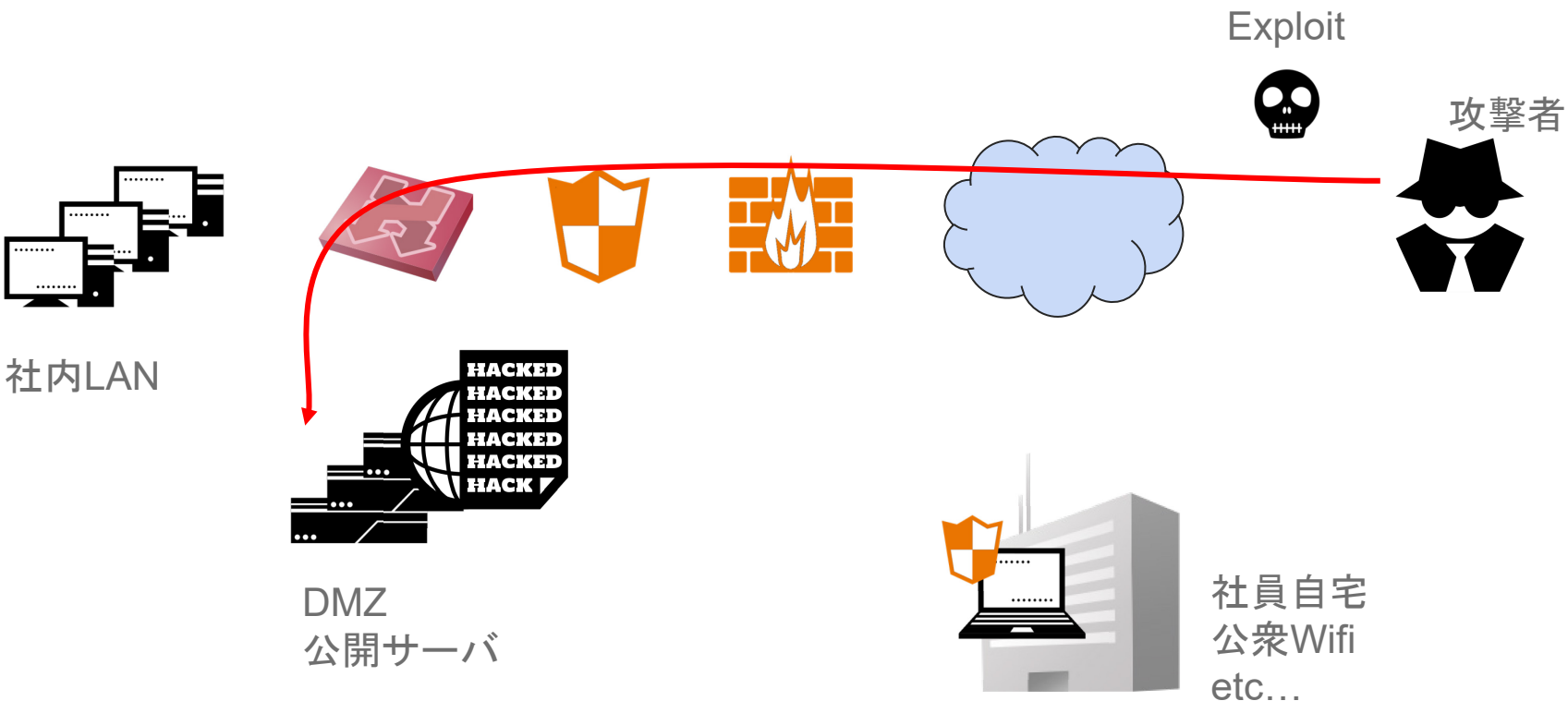
3.c.17 ユーザの行動を切っ掛けとしない誤検知例

- MDM製品の端末情報収集系スクリプト実行
- セキュリティ製品同士の干渉
  - 互いを消そうとする
  - 互いのAPIフックや dll インジェクションを異常と検知する
- システムファイル等の定期的なクリーンアップ

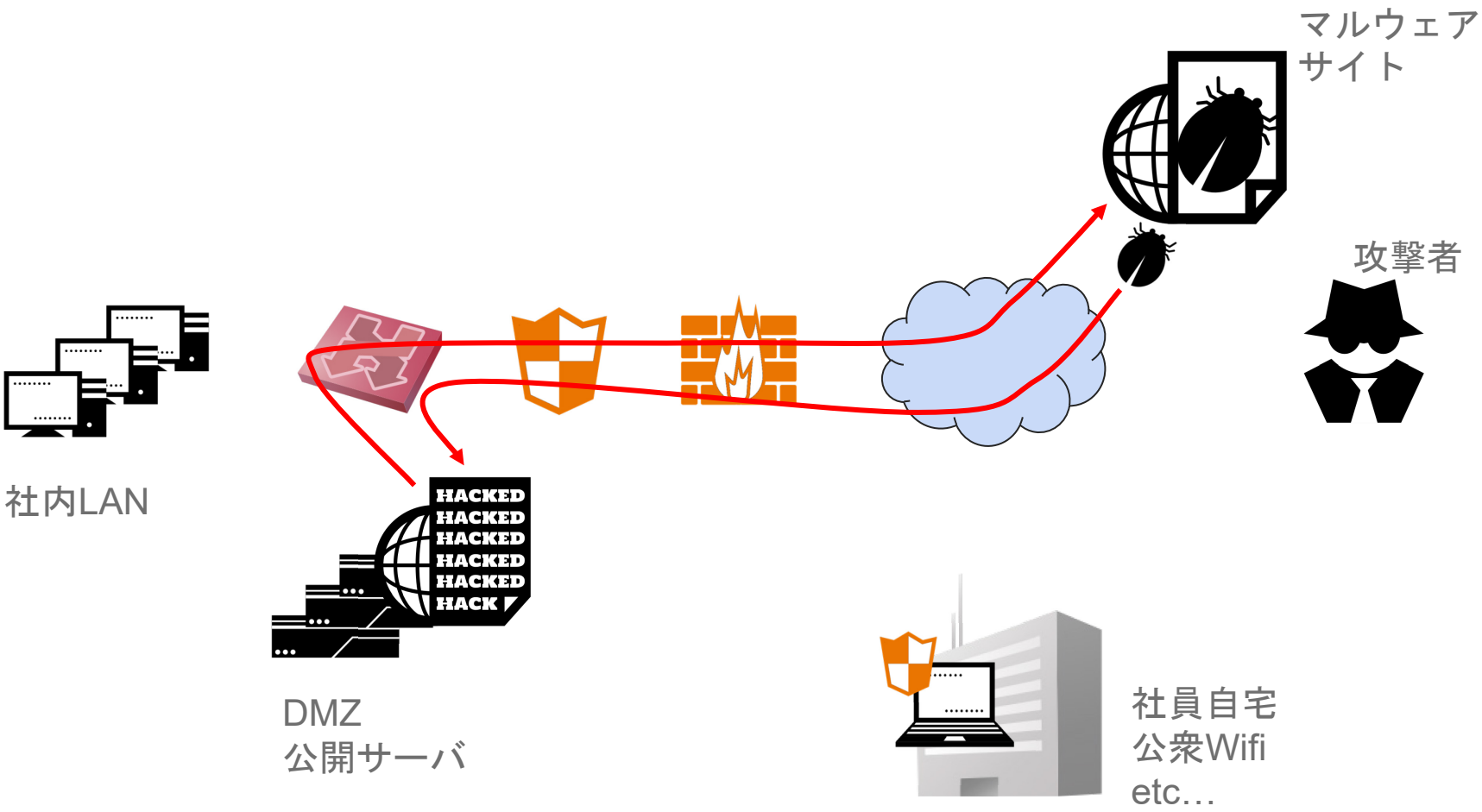
3.c.18 ユーザの行動を切っ掛けとしない不正処理

- マルウェアや攻撃者によって前述のものが悪用されるとかなり厄介
- 特にいつのまにか仕掛けられていた永続化は恐怖
  - 影響範囲や調査対象が莫大になってしまう
- 対策としては
  - EDRの導入
    - 直接入れられないホストがあっても、EDRが入っている端末がやられていないことを確認できれば調査範囲は限定可能
  - EDRログやサーバログを他のデータレイク（SIEMとか）に保存
    - 攻撃者に改ざんされていないと信頼できるログで影響の分析が出来ることが重要
  - EDR 以外の基本的な対策の徹底
    - 不要なポートのクローズやアップデート/パッチ/脆弱性の管理

3.c.19 ユーザ行動に因らない不正処理イメージ (1/2)



3.c.20 ユーザ行動に因らない不正処理イメージ (2/2)



3.c.21 EDR運用の難しいポイント

- セキュリティソリューションの中でも突出して誤検知が多い
  - 管理者業務やオフィスマクロを用いた業務自動化は間違いなく誤検知されると覚悟したほうが良い
  - スクリプトの実行ファイル変換とかもかなり。。。
- 攻撃に対する最後の壁であることが多く、EDRの解析を間違えると大惨事になりやすい
  - 多層防御をして、可能な限りEDR（エンドポイント）にたどり着かせないことが重要
  - 誤検知の自動処理や検知除外も重要（後述）

3.d.1 その他（Email / 脅威インテリジェンス / ASM / SSPM / 改ざん検知 / etc..）

Email セキュリティ

- 基本的にはインバウンドメールに対し次のような防御を提供
  - SPF/DKIM に基づく送信元メールアドレス詐称の検知
  - 添付ファイルのマルウェア・サンドボックススキャン
  - メール内URLのレピュテーションチェック
- 基本的にはあまり処理で悩むことはないが、アラートの数で圧倒されがちであるのと誤検知によるビジネスメール不達で少しトラブルになることも
- 運用にあたって重要なのは次の2点
  - 隔離などがエンドユーザにわかりやすくあること
  - 隔離開放リクエストや調査リクエストをある程度自動的に処理出来るようにしておくこと

3.d.2 その他（Email / 脅威インテリジェンス / ASM / SSPM / 改ざん検知 / etc..）

脅威インテリジェンス（戦術、IoCs）

- 日夜、SNSや各種業界団体、ベンダー等から共有
- 各種製品やSIEMにアドインするだけなら簡単だが、運用するのであればどのような状態が危険か理解・定義するのが必須
- そしてそのためには次の情報が必要
  - 悪用された期間
  - MITRE ATT&CKのTTPsとのマッピング情報
    - 何のために、何を用いて、どのように
- これら情報のことをコンテクスト (Context) と呼ぶこともある
- これら Context と実際の検知を照らし合わせて齟齬があるかどうか
  - 齟齬があるならチューニング
  - 齟齬がないならこれまで説明したシチュエーションに近いもので解析を

3.d.3 その他（Email / 脅威インテリジェンス / ASM / SSPM / 改ざん検知 / etc..）

Attack Surface Management

- システム内に存在する管理外ホストの存在や管理下ホストの脆弱性を発見・管理する製品・サービス
- 運用にあたっては自組織の優先順位を明確に定めることが必要
- 運用上の懸念は、取引先が自組織に無断でスキャンを放った時…
  - それって除外すべき？抗議すべき？遮断すべき？答えは…

SaaS Security Posture Management / 改ざん検知

- 正当な業務上の振る舞いが誤検知されないようにしましょう
- 次の連携が簡単に出来るように整備
  - 作業開始・終了や作業内容
  - 作業枠内のアラートへの情報添付や作業者への照会

#4
判断を誤らせる要素とその対策

a. システムアーキテクチャの問題
（ログ遅延 / 情報不足 / 頻発する誤検知 / etc…）
b. オペレータや組織の問題
（認知バイアス etc…）

4.a.1 システムアーキテクチャの問題 - ログ (1/2)

- いざ運用を始めてみたら想像もしないトラブルに見舞われるもの…
- ログにまつわるトラブル例
  - ログの欠損
    - UDP でログを配送していると、大きすぎるログは部分的に欠損
    - 例えばログ内に HTTP のヘッダの詳細が乗ってくるようなログはサイズ超過のリスクがある
  - ログの配送遅延
    - 配送経路上でパケ詰まりが起きると、数分の遅延は覚悟
    - あるいは、旧来のSIEMのようなログの集権アーキテクチャを変更
  - 両者 TCP にすれば解決するように見えて…負荷増加という別の問題が
  - 欠損が許されないログはTCPに、他をUDPに
    - 例えば製品アラートや発生が稀なログのみTCPとか

4.a.2 システムアーキテクチャの問題 - ログ (2/2)

- いざ運用を始めてみたら想像もしないトラブルに見舞われるもの…
- ログにまつわるトラブル例
  - ログのパーシング（解釈）不良
    - JSON以外のフォーマット (ベンダー独自、CEF) を正規表現などで正確に解釈するのは不可能
      - LLM ならやってくれる？
      - 某 SIEM だけはかなりうまくパーシングしているみたい
    - パーシングがうまくできないと、ログの検索や検知に深刻な影響が出る可能性が高い
      - ないと思ったログが実はあったり...
      - 検索や検知などの、運用の根幹となるものの信頼性を損なう
    - 可能な限り簡素なフォーマットにしよう…最悪はJSONで

4.a.3 システムアーキテクチャの問題 - 情報

- 情報（共有）にまつわるトラブル例
  - ネットワーク構成図がない
    - 通信の経路がわからないのでNATを使用しているネットワークだと攻撃の大元を特定できないケースがある
    - NATを使っている場合にはそのNATを行っている機器情報とNAT変換ログは連携必須
  - 検知ロジックがわからない
    - よくあるけどどうしようもない
    - せめて、各社、説明はしっかり書いてほしいと願っています
  - 脅威インテリジェンスのコンテクストが共有されてない
    - そのIoCはフィッシングサイトか？マルウェアのC2か？
    - 複数の脅威ソースで検索をかけてコンテクストを確認
    - 個別の会社・製品の紹介は差し控えます

4.a.4 システムアーキテクチャの問題 - 誤検知

- 誤検知が多い製品・ソリューションはミスの元
  - 誤検知でブロックが行われれば、業務影響も懸念
  - 誤検知によってシステムトラブルが起きたなら検知除外や防御モードの変更が必要だが。。。
    - 誤検知だけを正確に検知除外できますか？
  - 誤検知が続けば、それが人間の認知へも影響し始めます

4.b.5 オペレータや組織の問題 - 認知バイアス (1/4)

- 理想的には、誤検知が何件続こうが、あるいはインシデントが何件発生しようが常に客観的にアラートを識別すべき
- ただ、現実には、誤検知がずっと続けば今回の検知も誤検知だと思うもの
  - 正常性バイアスや確証バイアス、ホットハンドの誤謬など
  - セキュリティ運用でこれを防ぐには、処理のプロセスを重視する必要がある
- ただし、このような認知バイアスは、似たパターンのアラートの処理をより迅速に処理できる可能性も秘めている
  - 実際、運用をしていればわかるが、このような認知バイアスに基づく処理でもかなりの高精度でアラート処理は出来る
  - 問題は使い分けの意識と危険エリアの可視化、繰り返し処理の自動化

4.b.6 オペレータや組織の問題 - 認知バイアス (2/4)

- 認知バイアスがセキュリティオペレーションに悪影響を及ぼすもう1つの例が後知恵バイアス
  - 過去の事象を全て予測可能であったかのように見えるバイアス
  - 例: インシデント発覚後、過去に発生した関連するアラートが明らかに危険性を訴えるものであったかのように見える
  - このバイアスによって、セキュリティ運用の現場では、判断ミスに対し過剰な罰則を自他に強いている例が存在
  - ストレスや緊張感の下での業務を強いられ、またこれまで説明した様々な問題が潜在しているセキュリティ運用の現場では”判断ミス”を起きてはならないことだとするのはご法度
  - むしろ必ず起きるものとして、1つの判断ミスでは問題が起きない制度・システム設計に挑戦すべき

4.b.7 オペレータや組織の問題 - 認知バイアス (3/4)

- その他留意すべき認知バイアス
  - アンカリング バイアス
    - 最初に受け取った情報に過度に依存する（信じる）
  - 根本的な帰属の誤り
    - 他人の振る舞いはその人のパーソナリティに問題があるとし自分の振る舞いには環境的な因子が問題であるとする
- その他の説明と認知バイアスを克服する方法の説明資料：
  - https://www.sans.org/presentations/identifying-and-countering-cognitive-bias/
  - https://www.betterup.com/blog/cognitive-bias

4.b.8 オペレータや組織の問題 - 認知バイアス (4/4)

注意事項:

- 基本的に、他人の認知バイアスを正すことは難しい
  - 人の考えを変えるには、その人自身の課題の認知が必要
  - 大抵、問題を生む認知バイアスを抱えている人は、自分の認知バイアスへの問題意識が欠如
- 誰もが認知バイアスを持っている
  - 私も、皆様も
  - これらの話を聞いて、ギクッと思うくらいでちょうどいい
- 重要なのは、そのような認知バイアスを自分が持っていることを自覚し、必要な時にはそれを制御し、客観的に物事を捉えること
- 他人を正すのではなく、自分を制していきましょう

4.b.9 オペレータや組織の問題 - その他

- （人的）リソース
  - 人はいない、育たない（というリスクを覚悟しましょう）
  - 勉強するものを絞り込みましょう
    - OSの基本的な機能（便利機能含む）は価値が高い
    - セキュリティのよくわからん横文字新技術はベンダーとかコンサルに任せましょう
      - 大抵、昔ながらの概念を今風に言い換えているだけ
    - 基本は、業務・ユーザ・顧客、そしてアーキテクチャの理解
- 予算
  - 予算には限りがあるので出来ることからやりましょう
    - セキュリティ運用組織の教科書やISMS認証へ
    - 同一・類似業界のリーディングカンパニーを真似るのも有効
      - 成長・成熟してきたら独自に企画してみても◯

#5
まとめと質疑応答

5.1 本資料/発表の要点

- 監視環境の変遷
  - ネットワーク +エンドポイント + アイデンティティ
- 監視の観点
  - システム構造
  - アラート構造
  - 業務理解の重要性
- 組織のミスの原因と対策
  - システムアーキテクチャの問題
  - マネジメントの問題

Special Thanks

レビュー、執筆支援等

※五十音順に記載

飯沼翼さん
- ゲヒルン株式会社

石川章史さん
- シスコシステムズ合同会社

木内智之さん

平澤友里さん
- シスコシステムズ合同会社

利用アイコン画像 :
- https://www.security-design.jp/security-icons
- https://github.com/interop-tokyo-shownet/shownet-icons

To be continued…?

補足スライド

Intrusion Detection System (IDS) / Intrusion Prevention System (IPS)

- IDS / IPS は古くからネットワークレイヤーの境界にて
  攻撃の検知・防御を担ってきたセキュリティソリューション
- Firewall と L3 ルータの間に対する物理的設置、あるいは L3 ルータの
  パケットミラーリングを受ける形で導入されるケースが多い
  - 要件によってはシステムの最前面 (Firewall の手前) に設置されるケースも
  - ホストにインストールされる Host型 IDS という製品ジャンルもあるが
    今回は割愛
- 動作するネットワークレイヤーは TCP/IP におけるインターネット層と
  トランスポート層
  - アプリケーション層において動作すると謳う製品もあるが通信を復号
    していない場合には効果を期待するのは禁物

IDS / IPS の動作原理

- IDS / IPS として知られるものの多くはその機器・ソフトウェアを通過して流れるネットワークパケットに対し特定の条件を満たすものを検知
- この条件のことを一般にシグネチャと呼び、これは基本的に文字列や正規表現の形で定義される
  - 通信の回数や方向、文字列が含まれるヘッダーの種類等の条件が組み合わされることも