

# Windowsログ分析の基礎 ～基本編～

– ADへの攻撃を理解するために –

一般社団法人JPCERTコーディネーションセンター



## 本コンテンツについて

- 本コンテンツは、社内ネットワーク（主にWindowsネットワーク）におけるログ分析の基本的な知識を学ぶための資料です
- 学習目的で自由にお使いください
- 編集・再配布などをご希望の場合は、  
JPCERT/CC 広報（[pr@jpcert.or.jp](mailto:pr@jpcert.or.jp)）までご連絡ください

本コンテンツは、攻撃者のネットワーク侵入手法を学び、インシデント発生時に必要となるログ調査の中で主にWindowsのイベントログの調査を中心に学習するものです。インシデント対応では、検知 → 初動調査 → 一時対処 → 本格調査 → 報告 → 恒久対策 という流れで行われることが多く、本コンテンツは初動調査に特化しています。

## Agenda

1 社内ネットワーク基礎

2 社内ネットワークへの攻撃手順

3 Windowsイベントログ

4 Windowsイベントログの分析

5 ハンズオン

1章から4章までは、座学としてネットワーク侵入時の攻撃手法について学びます。

5章および～実践編～資料がハンズオンになっています。

すでに基本的な攻撃手法を把握している場合はハンズオンに進んでください。

## 1 社内ネットワーク基礎

## 2 社内ネットワークへの攻撃手順

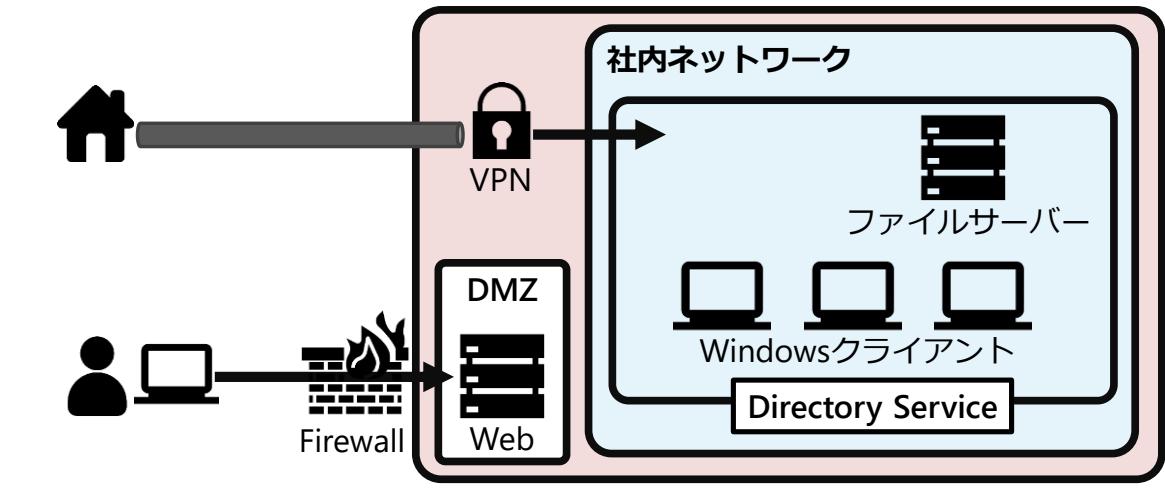
## 3 Windowsイベントログ

## 4 Windowsイベントログの分析

## 5 ハンズオン

まず、攻撃から守るべきネットワークの基本となるWindowsネットワークについて説明します。

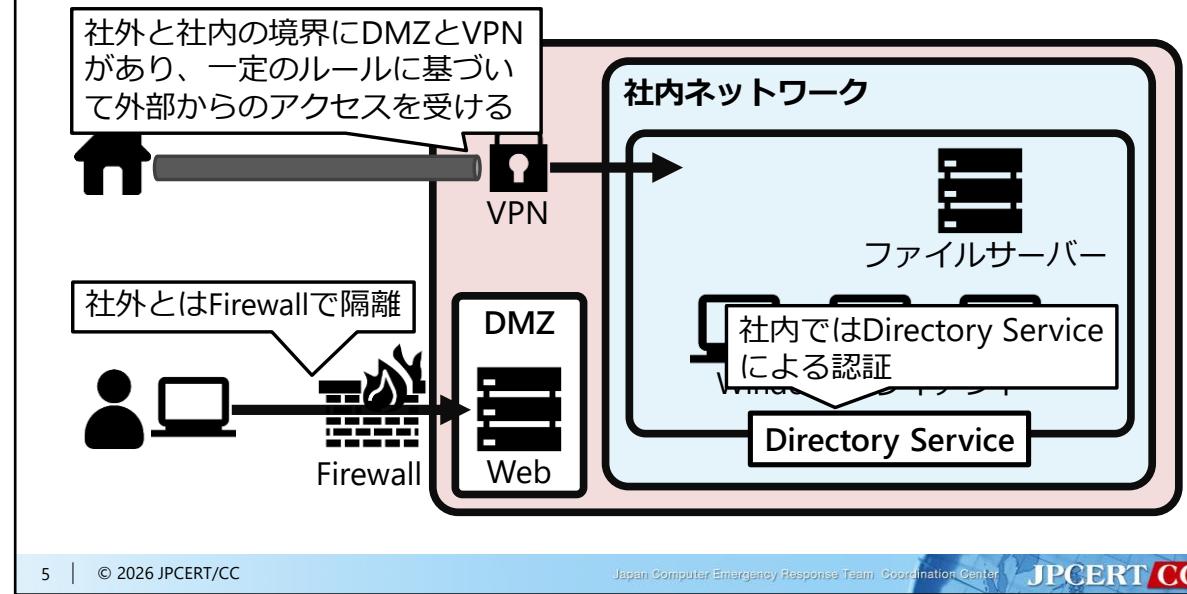
## 一般的な社内ネットワークの構成



この図は、一般的な社内ネットワークを主要な要素に絞り簡略化したものです。

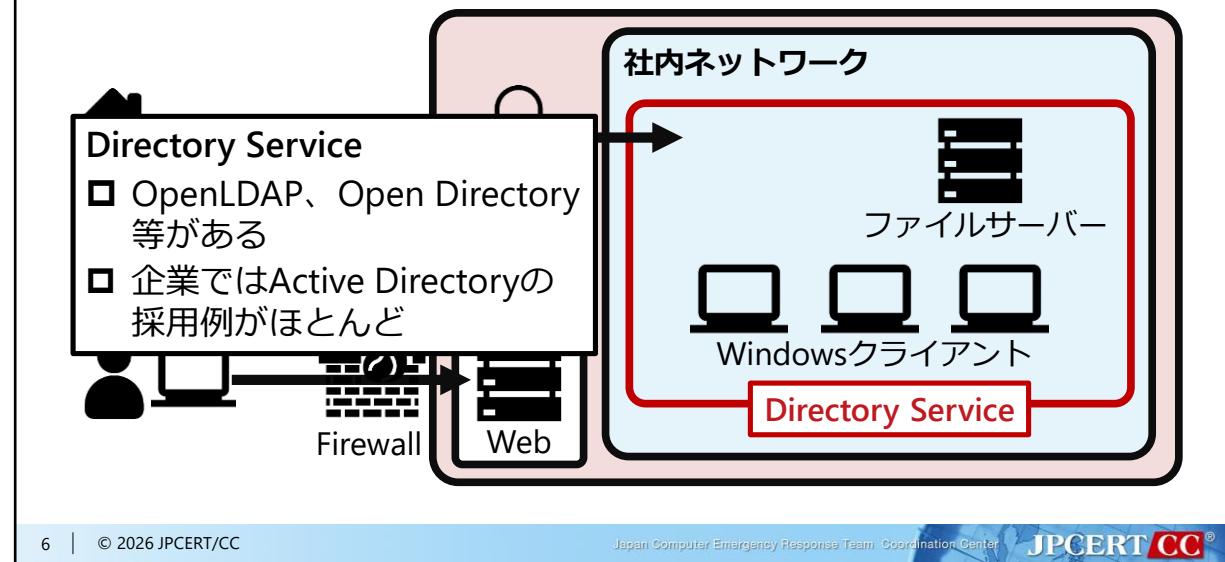
昨今、攻撃者のターゲットとなっているのは、VPN、Webサーバー、ドメインコントローラー、Windowsクライアント、ファイルサーバー（管理者権限で動作するサーバー群など）です。

## 一般的な社内ネットワークの構成



一般的な社内ネットワークでは、社外と社内の境界にDMZとVPNがあり、一定のルールに基づいて外部からのアクセスを受け付けます。そして、社内ネットワークではDirectory Serviceによって社内システムの認証が行われています。

## 一般的な社内ネットワークの構成



本コンテンツでは、基本的に社内ネットワークをWindowsネットワークで構成した組織を想定して説明を進めます。Windowsネットワークを理解する上で、AD、DS、ドメインコントローラーという単語がよく使われます。以下に、それらの言葉を整理します。

- Directory Service
  - ネットワーク内に接続されたリソースを管理するためのサービスで、OpenLDAP、Open Directory等がある
- Active Directory
  - WindowsサーバーにあるDirectory Service
- ドメインコントローラ
  - Active Directoryサービスを提供するサーバー

以降では、ネットワーク内のリソース、アカウント情報などを管理するサーバーをドメインコントローラーとして説明します。

## Active Directory Domain Service (AD DS)

### AD DSとは

- Windows Serverの**機能**で、Windowsネットワーク内の端末管理や認証・認可を行う

例えば、



- このサービス使える？
- ファイル見て良い？
- USBは使用禁止
- 5分でスクリーンロック

### ドメインコントローラーとは

- AD DSがインストールされた、認証・認可を行う**サーバー**
  - 管理する範囲は**ドメイン**という
- 複数台で運用することも可能（子会社／海外支局など）

Active Directoryには、複数の機能があります。

その中でも、Domain Serviceを提供する機能をAD DSと呼び、この機能はWindowsネットワーク内の端末管理や認証・認可を行います。

この、AD DSがインストールされたWindowsサーバーがドメインコントローラーとなります。

# ドメインとフォレスト

## ドメイン

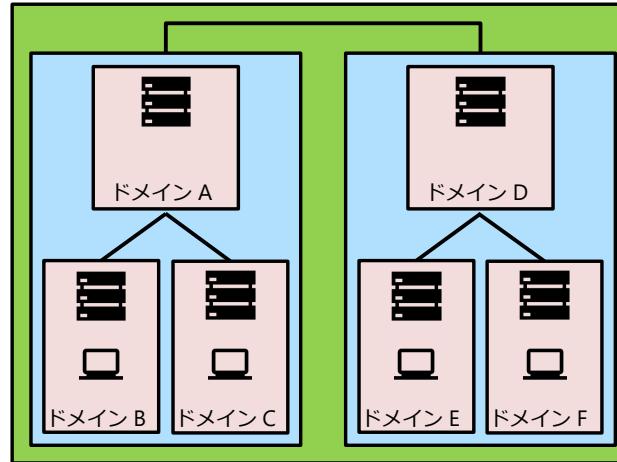
- あるドメインコントローラーが管理する範囲
- 複数台で管理することもある
- 親子：サブドメイン継承
- 信頼：相互に認証し合う

## ツリー

- ドメインの親子関係のみで形成された構造
- 子は親のサブドメインを用いる
- 事業部制・独立部隊がいる・子会社etc...

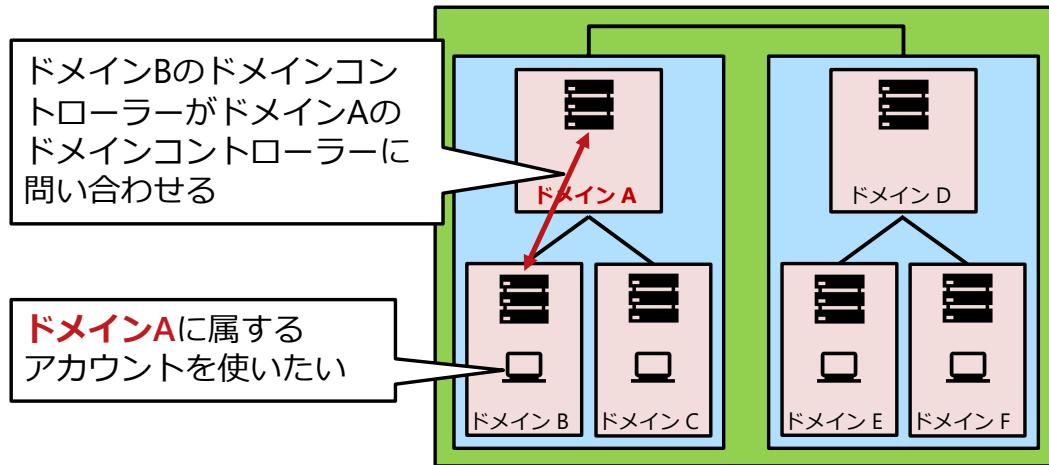
## フォレスト

- ツリー同士の信頼で形成された構造
- 信頼関係にあるドメインに命名規則はない
- 企業合併・子会社etc...



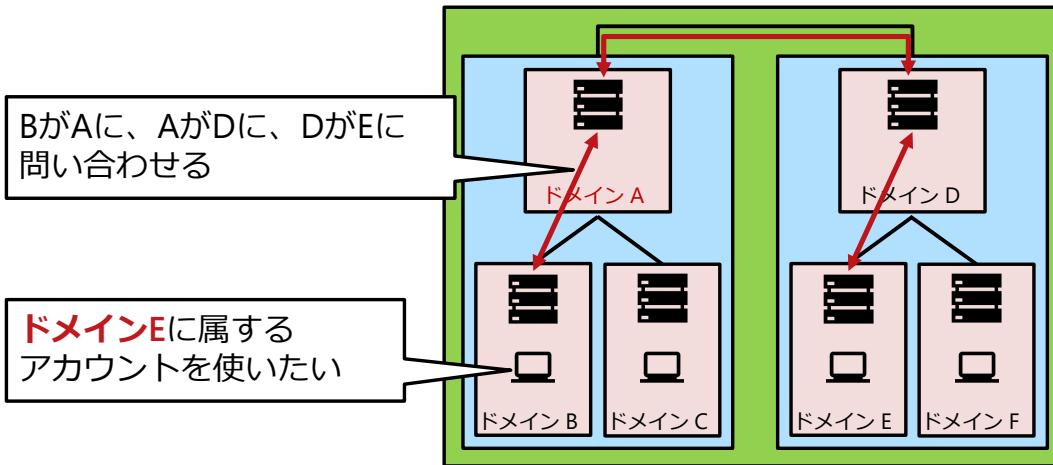
ドメインコントローラーが管理する一つの範囲をドメインと呼びます。このドメインは複数台のドメインコントローラーで管理することも可能です。さらに子会社やグループ会社をドメインで分け、これらの別組織のドメイン同士（サブドメイン）を結び付け、信頼関係を結ぶことで、異なるリソースへのアクセス可能にすることもでき、この信頼関係の構成をドメインツリーと呼びます。そして、複数のドメインツリーで構成されたものをフォレストと呼びます。フォレスト内のツリー間では全く異なるドメイン名として運用が可能になります。

## ドメインとフォレスト



ドメインBのクライアントがドメインAのサーバーにアクセスしたい場合、この図のような認証確認が行われます。

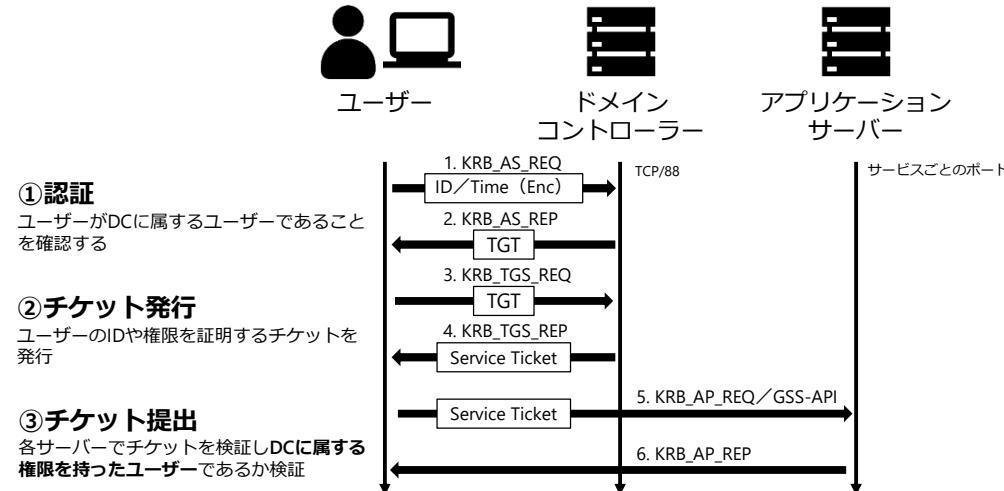
## ドメインとフォレスト



ドメインBのクライアントがドメインEのサーバーにアクセスしたい場合、この図のような認証確認が行われます。

## ADにおける認証／認可の仕組み

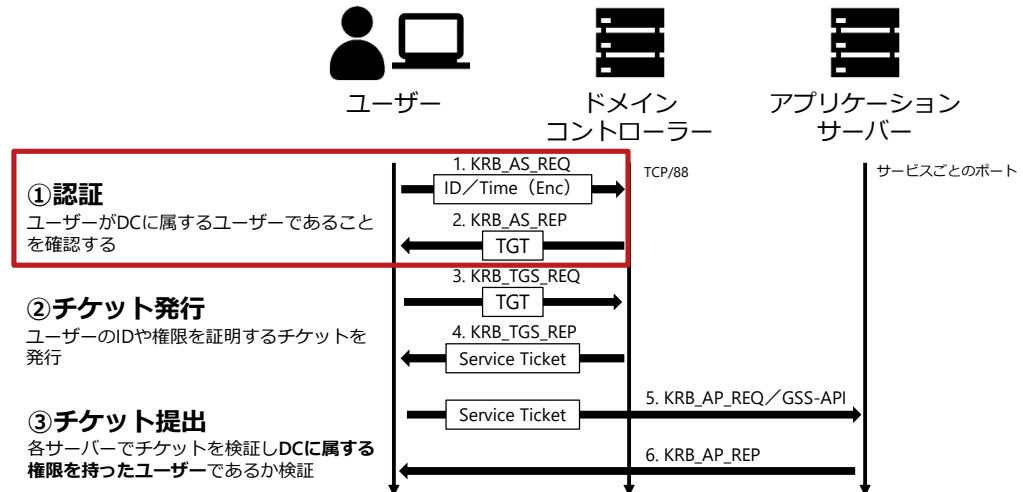
RFC 4120 - The Kerberos Network Authentication Service (V5)



ADによる認証はKerberos認証によって実現されています。  
以降では、Kerberos認証の流れを図で説明しています。

## ADにおける認証／認可の仕組み

RFC 4120 - The Kerberos Network Authentication Service (V5)



## ①認証 : KRB\_AS\_REQ/REP

### 1. KRB\_AS\_REQ

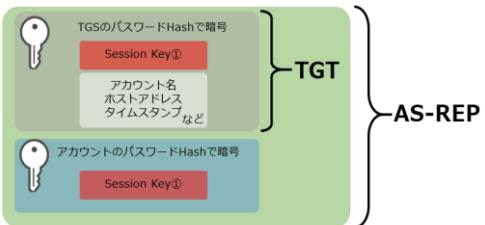
- ▣ TGTをもらうための認証情報を含んだリクエスト
  - ✓ アカウント名
  - ✓ 端末のIPアドレス
  - ✓ 時刻を自身のパスワードハッシュ（NTLM）で暗号化した値
  - ✓ nonce（リプレイ攻撃防止のための乱数）

### 2. KRB\_AS REP

- ▣ 時刻を復号し一致していればTGTとSession Key<sup>①</sup>（チケット用）を発行
- ▣ Authentication Server（≒DC）が発行する

### TGT (Ticket Granting Ticket) とは

サーバー利用時のチケット発行に用いるための、最初の認証で発行されるチケット（Service Ticketを要求するTicket）



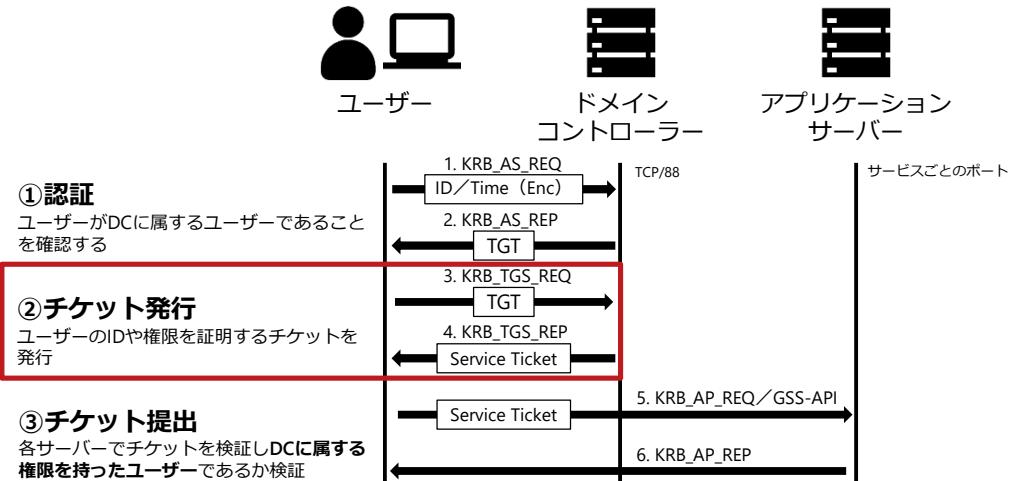
※1 ADはユーザーのPWを知っているので時刻を暗号化／復号できる

※2 ユーザーはkrbtgtのPWを知らないのでTGTを復号できない

出典：三井物産セキュアティレクション Security Knowledge 「パスワードってどこにある？その1」  
<https://www.mbsd.jp/research/20190514/password1/>

## ADにおける認証／認可の仕組み

RFC 4120 - The Kerberos Network Authentication Service (V5)



## ②チケット発行：KRB\_TGS\_REQ/REP

### 3. KRB\_TGS\_REQ

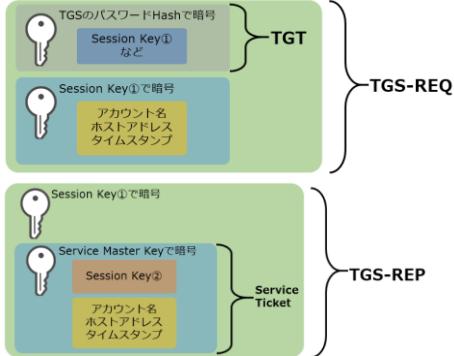
- Session Key①でアカウント名等を暗号化して送信
- サービスプリンシパル名（SPN）でサービスを指定（Session Keyで暗号化）
- SPNはアプリケーションサーバ（内で動くサービス）を特定するためのIDで、DC内で一意

### 4. KRB\_TGS\_Rep

- TGSの暗号鍵（krbtgtのPW=TGS既知の文字列）によってTGTを復号して検証できる
- 検証結果が正しければ、Service Ticketをユーザーに送信
- Session Key②（サービス用）を発行

### Service Ticketとは

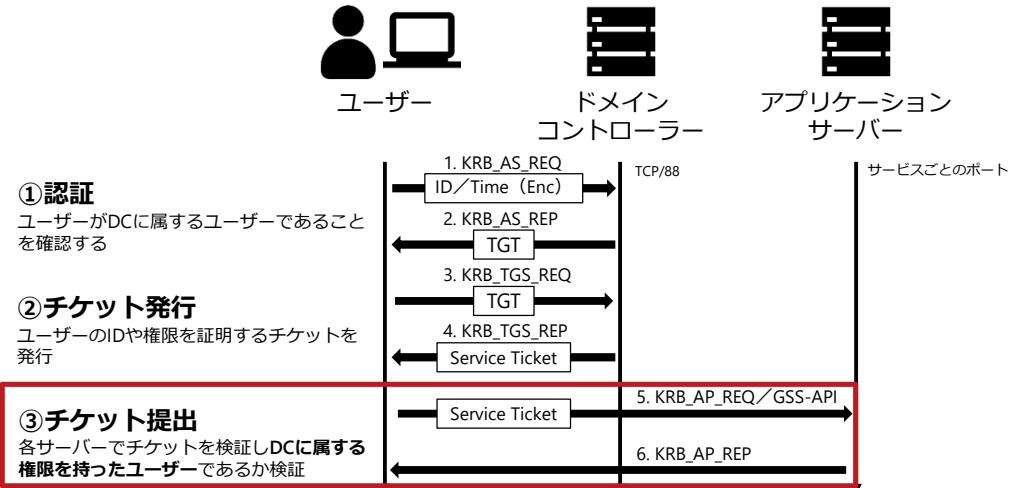
対象サービスで使えるアカウントを認証するためのチケット



出典：三井物産セキュアディレクション Security Knowledge 「パスワードってどこにあるの？その1」  
<https://www.mbsd.jp/research/20190514/password1/>

## ADにおける認証／認可の仕組み

RFC 4120 - The Kerberos Network Authentication Service (V5)



### ③チケット提出：KRB\_AP\_REQ／REP

#### 5. KRB\_AP\_REQ

- Service Ticketを送信して認可を求める
- Session Key②（サービス用）は別途記録

#### 6. KRB\_AP REP

- 認証結果を返す
- Session Key②で暗号化
- subkey等後続通信用の鍵を返す

#### 通信に使用するポート／プロトコル

- アプリケーションごとに異なる  
ポート／プロトコルを利用
- ✓ HTTP／SMB／LDAP etc...
  - ✓ GSS-APIという規格をKerberos  
で実装したものが使われる

# 確認演習：正常系におけるKerberos認証

演習

演習

## 1. Wiresharkをインストールして、パケットキャプチャしたデータを閲覧

- 演習ディレクトリ内のSMB.pcapngを見る
- Session Setup Requestでチケットが送信されていることを確認
- ✓ SMB2->Security Blob->GSS-API->Simple Protected Negotiation->negTokenInit->krb5\_blob->Kerberos->ap-req->ticket
- ✓ enc-part以上は掘り下げて閲覧できない (cipherパラメーターを展開できない)

## 2. configを設定して再度閲覧

- Preference->Protocols>KRBS
- Try to decrypt Kerberos blobsをチェック
- Kerberos keytab fileにtest.keytabファイル指定
- test.keytabはアカウントのNTLMハッシュの詰め合わせ

問題

## なぜ1ではパケットの中身が見られず、2で見られたか？

- ① このパケットはKRB\_(AS|TGS|AP)\_(\_REQ|\_RES)のどれに当たるか
- ② 何の情報が何の鍵で暗号化されていたか
- ③ Wiresharkはどういった処理を行っているのか

Kerberos認証の仕組みを、実際の通信パケットから理解する演習です。  
Wiresharkを使ってSMB通信パケットを確認してください。

## 確認演習：config設定前

演習

The screenshot shows a Wireshark capture of SMB traffic. The timeline pane shows 8 captured frames. The details and bytes panes show the content of each frame. Frame 8 is highlighted with a red box, showing a Kerberos authentication message (Ticket Granting Ticket) with a long hex dump.

Frame 8 details:

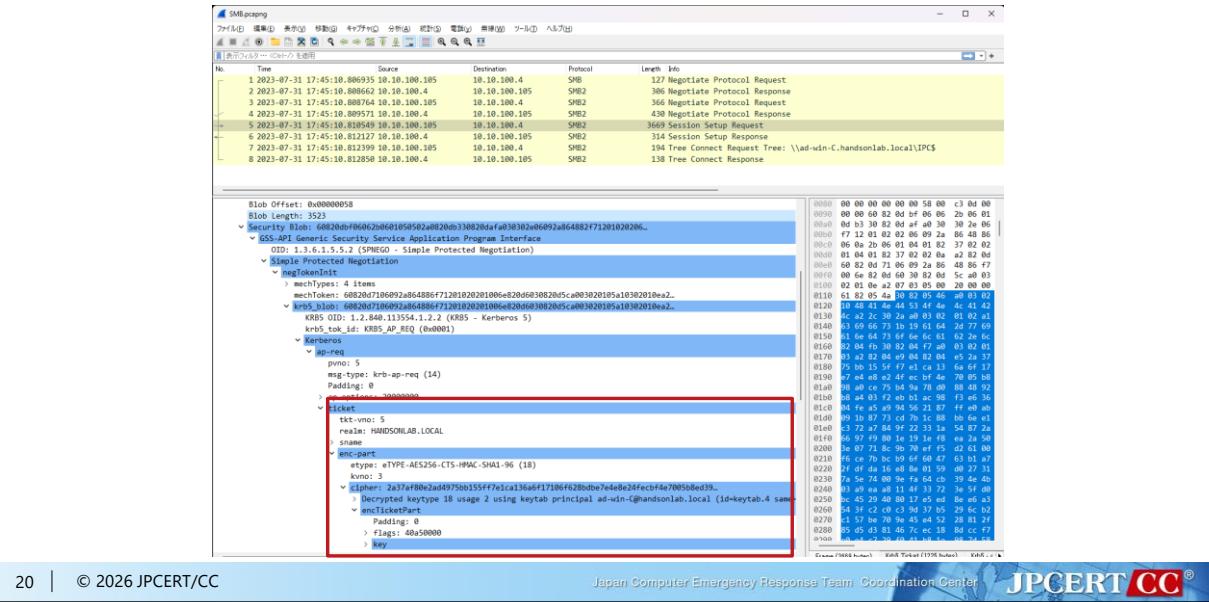
- Source: 10.10.100.4
- Destination: 10.10.100.105
- Protocol: SMB
- Length: 138
- Info: Tree Connect Response

Frame 8 bytes:

```
0170 03 a2 82 04 e9 04 82 04 e5 fa 17
0180 73 b6 15 5f f7 e1 ca 13 ea 6f 32
0190 4e 9c 0d 01 00 00 00 00 00 00 00 00
01a0 48 40 01 73 bd 9c 7d 09 08 48 32
01b0 08 ad 09 f2 eb h1 ac 98 f3 e6 36
01c0 00 00 00 00 00 00 00 00 00 00 00 00
01d0 00 00 00 00 00 00 00 00 00 00 00 00
01e0 c3 72 a7 8d 9f 22 31 1a 54 87 2a
01f0 66 97 f0 89 1e 19 1e ff ea za 58
0200 00 00 00 00 00 00 00 00 00 00 00 00
0210 76 ce 7b bc 39 6f 6a 42 13 11 47
0220 2f d6 d6 15 e3 8e 01 59 00 27 31
0230 7a 5c 74 00 9a fa 64 cb 39 49 4b
0240 00 00 00 00 00 00 00 00 00 00 00 00
0250 c5 45 29 40 00 17 e1 ed 8e 6f a3
0260 54 3f c9 c9 c3 9d 37 b5 29 6c 12
0270 00 00 00 00 00 00 00 00 00 00 00 00
0280 25 45 00 00 46 7c e1 58 6d cc 77
0290 00 4d c7 29 f0 41 bd 1e 08 7d 58
02a0 00 18 60 00 46 1f ff ff fd 54 58
02b0 00 00 00 00 00 00 00 00 00 00 00 00
02c0 75 4c 79 a4 04 48 a9 14 12 95 48
02d0 3b 74 a0 75 13 e1 54 c3 05 59 09
02e0 2b 49 d6 1a 01 74 d2 22 00 46 03
02f0 00 00 00 00 00 00 00 00 00 00 00 00
0300 aa fa b5 a1 06 0a 71 04 48 9f 5c
0310 48 34 c9 49 61 fc 50 3e 2a fe ff
0320 02 cb 29 00 00 00 00 00 00 00 00 00
0330 00 00 00 00 00 00 00 00 00 00 00 00
0340 44 9c c4 42 77 e8 43 cc 1b 7c 03
0350 00 87 05 e5 5c ee 00 87 11 e3 9c
0360 00 00 00 00 00 00 00 00 00 00 00 00
0370 fe 98 ee e7 c1 55 35 58 04 27 ff
0380 15 1c d9 29 30 44 68 99 4b fc 00
0390 74 4c 43 53 13 01 0e 0e 0e 0e 0e 0e
```

## 確認演習：config設定後

演習



## 確認演習：回答

演習

問題

なぜ 1 ではパケットの中身が見られず、2 で見られたか？

- ① このパケットはKRB\_(AS|TGS|AP)\_(REQ|RES)のどれに当たるか
- ② 何の情報が何の鍵で暗号化されていたか
- ③ Wiresharkはどういった処理を行っているのか

回答

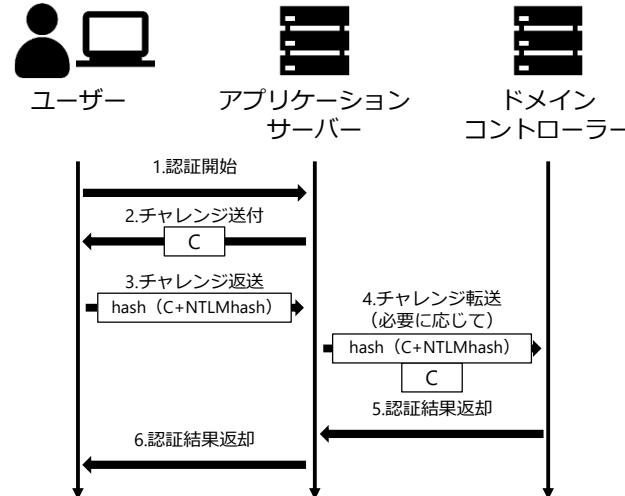
1では鍵が分からないのでService Ticketを復号できないが、2ではkeytabに鍵が保存されているので復号できた

- ① このパケットはKRB\_(AS|TGS|AP)\_(REQ|RES)のどれに当たるか
  - ✓ KRB\_AP\_REQ (Service Ticketの提出リクエスト)
- ② 何の情報が何の鍵で暗号化されていたか
  - ✓ Service TicketがSPNにひも付く鍵で暗号化されている
- ③ Wiresharkはどういった処理を行っているのか
  - ✓ keytabに記録されたSPNにひも付く鍵で復号

このパケットは、認証結果を返すKRB\_AP\_REQであり、Session keyで暗号化されているので、そのままでは内容を確認することはできません。Wiresharkにキーを与えれば、その中身を確認することができるようになります。

## その他の認証方式

### NTLM認証



Kerberos認証以外にもNTLM認証があります。この認証方式で運用された認証を回避する複数の攻撃手法が確立しており、またマイクロソフトが本認証方式を廃止することがアナウンスされています。

<https://jpwinsup.github.io/blog/2024/08/22/ActiveDirectory/Authentication/retireNTLM/>

# NTLM認証とは

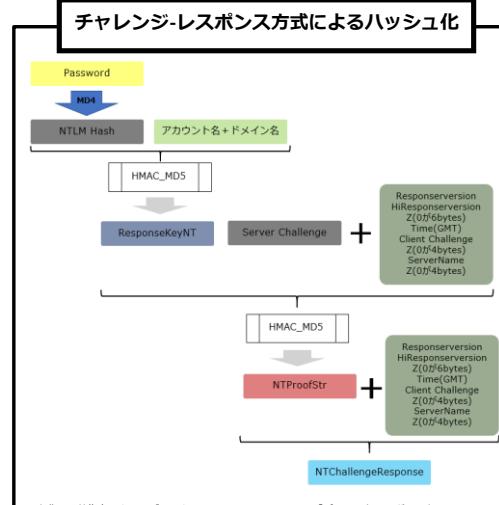
## チャレンジ-レスポンス方式

- ランダムな文字列 + パスワードをハッシュ化
- サーバー側で同様の処理を行いパスワードと一致するか検証

## NTLM認証における実装

- パスワードではなくNTLM Hashを用いる
- アカウント名とドメイン名も付与する
- HMAC-MD5を用いる

## チャレンジ-レスポンス方式によるハッシュ化



出典：三井物産セキュアティレクション Security Knowledge 「パスワードってどこにあるの？その1」  
<https://www.mbsd.jp/research/20190514/password1/>

NTLM認証は、チャレンジ・レスポンス方式の認証でパスワードをハッシュ化・暗号化して送受信します。そのため、パスワードを知らなくても、ハッシュ化されたパスワード情報を入手できれば悪用できるという問題点があります。

## リモート認証とローカル認証

### リモート認証

- ドメインコントローラによる認証
- ドメインコントローラに認証結果が記録される
- ドメインユーザー
  - ✓ [AD ドメイン名]¥[アカウント名]
- 管理者権限 : Domain Administrator

### ローカル認証

- ローカル端末で認証
- ローカル端末に認証結果が記録される
- ローカルユーザー
  - ✓ .¥[アカウント名]
- 管理者権限 : ローカルAdministrator

### 共通点

- Windowsのログインユーザーとして使用可能
- 各マシンのイベントログに記録が残る
- RDP接続時のユーザーとして使用可能

ここまで説明してきた認証方式は、ドメインコントローラによるリモート認証です。

一方で、Windows OSではローカル認証も存在します。これは、端末上で作成されたユーザーにログオンする仕組みで、端末自体にはアクセス可能になりますが、他のWindowsネットワーク上のリソースにはアクセスできません。

# ドメインアカウントの権限管理の重要性

## 業務要件と最小権限の原則

### □ 必要以上の権限を与えない（不用意に管理者権限を与えない）

- ✓ 営業部は顧客先や契約資料にアクセス可能
- ✓ 情シス担当者は各端末を管理可能
- ✓ 各部署は個別に設置されたプリンターのみ使用可能

## なぜ、権限管理が重要なのか？

### □ 管理者権限は攻撃者のターゲットになりやすい

- ✓ GPOによって全端末に影響を与えることができる
- ✓ 任意のユーザー や権限を作成できる
- ✓ ADのデータダンプ等を実行できる

➡ 攻撃者のやりたいことが  
すべてできる

ドメイン内で使用するアカウントの権限は重要です。

もしAdministratorsやDomain Adminsグループの所属するユーザーが使用している端末が乗っ取られたら、管理者権限でそのWindowsネットワーク内のあらゆる操作が可能になってしまいます。

そのため、攻撃者は管理者権限を持つユーザーにアクセスするためにさまざまなシステムに侵入を繰り返します。

通常の端末やサーバーで使用するユーザーの権限は最小限にして、その端末が乗っ取られたとしてもWindowsネットワーク全体への影響が及ばないようにする必要があります。

# グループポリシーオブジェクト (GPO)

ドメインに参加している端末の設定を  
ドメインコントローラー側で管理する仕組み

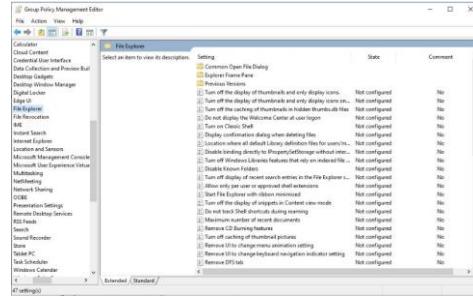
□ ルールを定めたGPOをDCの各グループにリンク

- ✓ 端末ごとのローカルGPOとDCの持つGPOがある
- ✓ ローカルGPOが通常優先されるが、DC側から強制することも可能

□ GPO保存場所

- ✓ ¥¥[ドメインコントローラー]¥SysVol¥[ドメイン名]¥Policies¥
- ✓ XML形式で保存

□ 通常は専用エディターを使って操作

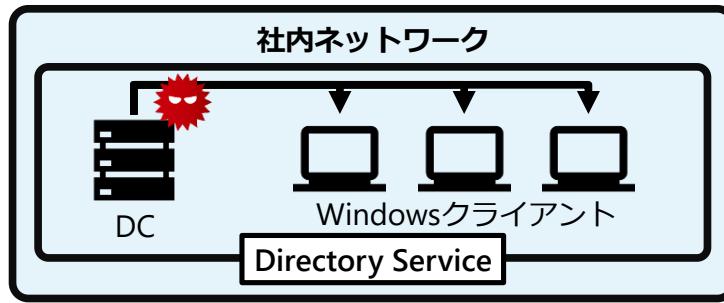


Windowsには、ドメインに参加している端末の設定をドメインコントローラー側で管理するグループポリシーオブジェクトという仕組みがあります。これを使用すると、各端末をリモートから設定したり、アプリケーションのインストールなどができます。攻撃者がこの機能を悪用してマルウェアを各端末に配信することもあります。

## SYSVOLを悪用したマルウェア拡散

### SYSVOLとは

- ドメインコントローラに存在する共有フォルダー
- クライアントなどに配布するスクリプトなどが保存される
  - ✓ クライアントへのマルウェア拡散に悪用される場合がある
- スクリプト内にパスワードなどの記載がある場合は注意



グループポリシーは、通常SYSVOLフォルダー（C:\Windows\SYSVOL\domain）から共有されます。

このフォルダーから、各種設定やスクリプト、アプリケーションなども配布されるのですが、このフォルダーに保存されるスクリプトには重要なシステムのアカウントやパスワードなどが含まれていることも多く、攻撃者はこのフォルダーからアカウント情報を収集する可能性があります。

1 社内ネットワーク基礎

2 社内ネットワークへの攻撃手順

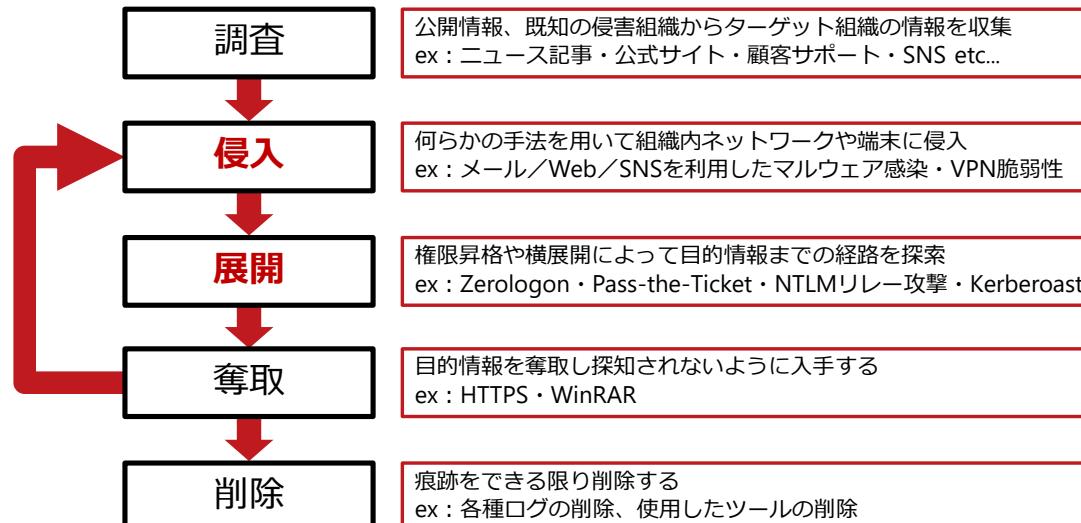
3 Windowsイベントログ

4 Windowsイベントログの分析

5 ハンズオン

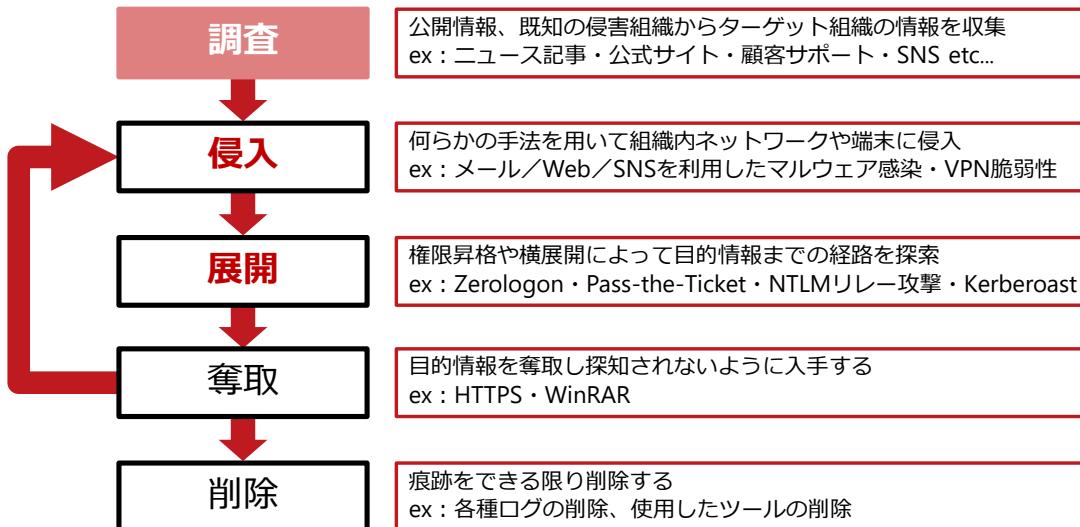
次に、社内ネットワークへの侵入手法について説明します。

## 攻撃者のネットワーク侵入の流れ



攻撃者のネットワーク侵入から目的達成までの活動の流れをフロー化したのがこの図です。  
以降では、各パートでよくみられる攻撃手法について解説します。

## 攻撃者のネットワーク侵入の流れ



## 調査

### 公開情報から相手を調査するフェーズ

- 外部公開資産があるか
  - ✓ Firewall、VPN、ルーター、Webサーバーなど
  - ✓ RDP、SSHなどアクセスポイントが公開されていないか
- メールアドレス、SNSなど従業員への>Contact手段
- すでに侵害された組織からの情報をもとに調査

### 検知は現実的ではないが**対策は可能**

- EASM (External Attack Surface Management)
- 公開メールアドレスの調査
- SNS利用時の注意点を周知

調査フェーズでは、攻撃者はネットワークへの侵入経路を探査します。

最近では、ご存じのとおりVPNやFirewallなど外部公開資産が攻撃のターゲットになることが多く、攻撃者はそのような外部からのアクセス先がないかを探査していると考えられます。

# EASM

## EASMとは

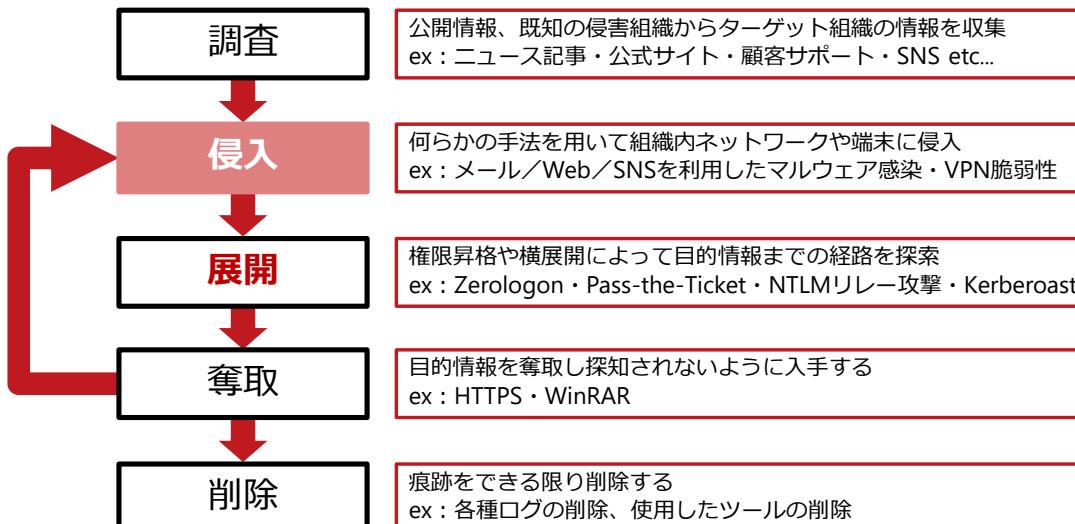
- 外部から攻撃される恐れのある公開資産を把握し、脆弱性対処やセキュリティリスクの低減を目的に管理を行う
  - ✓ Firewallやルーター、ファイルサーバー、Webサーバー、VPNなど
- 特にVPNが攻撃のターゲットになる場合が多いため管理が重要

## VPN管理のポイント

- 未管理のアクセスポイントを増やさせない（部署単位の判断などで未管理のデバイスを設置させない）
- すべての製品で攻撃を受ける可能性があることを理解して製品選定を行う
  - ✓ 導入する製品がどのようなログを取得できるのか、被害発生時にどのような調査が可能かを事前に把握する
- パッチ未公開の脆弱性が公表されることを前提に、公表された際にどのような対応（VPNの停止など）をするかを事前に検討しておく

外部から攻撃される恐れのある公開資産を把握し管理することは、昨今のセキュリティ対策の中でも重要なことです。昨今のセキュリティ対策の中でも重要なことです。  
攻撃者が公開直後の脆弱性を悪用してVPNに不正ログインしたり、ゼロデイ脆弱性を使って侵入したりするケースを多く確認しています。  
VPNが増えれば増えるほど攻撃されるリスクを増やすことにつながることを理解し、非管理のアクセスポイントを増加させないようにしてください。  
また、どの製品を使えば攻撃されるリスクを抑えられるということではなく、すべての製品が攻撃のターゲットになる可能性があります。そのため、管理している製品に脆弱性が出来ることを想定して対応フローを整備しておくことも重要です。

## 攻撃者のネットワーク侵入の流れ



## 侵入

実際に攻撃を行って権限を奪取するフェーズ

- 外部公開資産の脆弱性を悪用
- メール経由でマルウェアを実行させる
- SNS経由で従業員にコンタクトし、マルウェアを実行させる

侵入への対策

- 外部公開資産の洗い出しと脆弱性管理
- セキュリティ製品の導入
- セキュリティ教育
- ログ分析



100%侵入を防ぐことは難しいが、  
対策を取ることでリスクの低減につながる

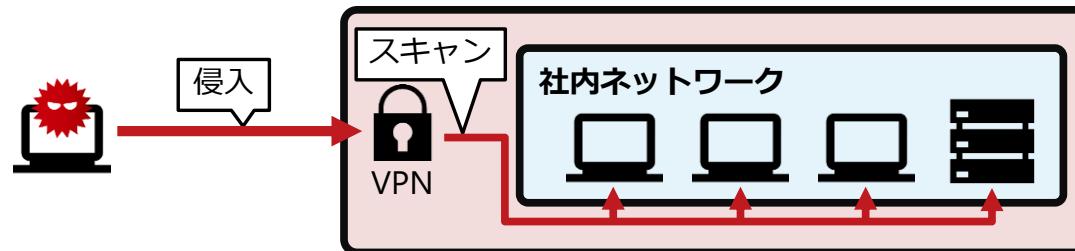
侵入には多数のパターンがあります。

調査フェーズで説明した、外部公開資産の脆弱性を悪用する以外にも、メールでの攻撃やSNS経由の攻撃があります。

# 外部公開資産の脆弱性を悪用

## VPNへの攻撃

- 狙われることが多いVPN製品
  - FortiGate製品、Ivanti製品、SonicWall製品、Palo Alto製品、Array Networks製品
- VPN機器に侵入した攻撃者は、VPNから社内ネットワークに対してネットワークスキャンを行い、侵入できるターゲットを見つける



外部公開資産経由の攻撃の場合、攻撃者はVPNなどに侵入が成功するとその機器経由で内部ネットワークにスキャンを行い、侵入するターゲットを探します。そのため、VPNから直接アクセスできる範囲を制限したり、VPNから不要な通信が出ていないか監視をするなどの対策が考えられます。

## RDPやSSHを外部公開しない

### ブルートフォース攻撃

アカウント名・パスワードを総当たりでログイン可能か調査する攻撃

- パスワード辞書やHydraなどが用いられる

公開サーバーは必ずターゲットになる

- RDPとSSHは常に攻撃を受けている
- 公開しているつもりはなくとも、モバイルWifi接続時にグローバルIPアドレスが知らないうちに適用されて、攻撃を受けている場合もある

パスワード認証の前段で防御

- 管理画面を公開しない
- 接続元IPを制限する

RDPやSSHサーバーを外部公開しないのはセキュリティ対策の基本です。

これらのサーバーは、公開したら必ずパスワード総当たり攻撃をうけます。脆弱なパスワードを使っていれば侵入されてしまいます。

また、公開していないつもりでも、管理不十分で意図せず公開されてしまっているテストサーバーや、モバイルWifi接続時にグローバルIPアドレスが知らないうちに適用されて公開されてしまっているRDP端末などに侵入されるケースもあります。

このようなことにならないように、普段から強固なパスワードを使う・SSHサーバーは認証キーを使用する・アクセス元IPアドレスを制限するなどの対策が必要です。

## SNS経由の攻撃

### LinkedIn経由の標的型攻撃

- 攻撃者が、従業員に対してSNS経由でマルウェアを送信してくる
- 業務端末でSNSを使用している場合は、注意が必要



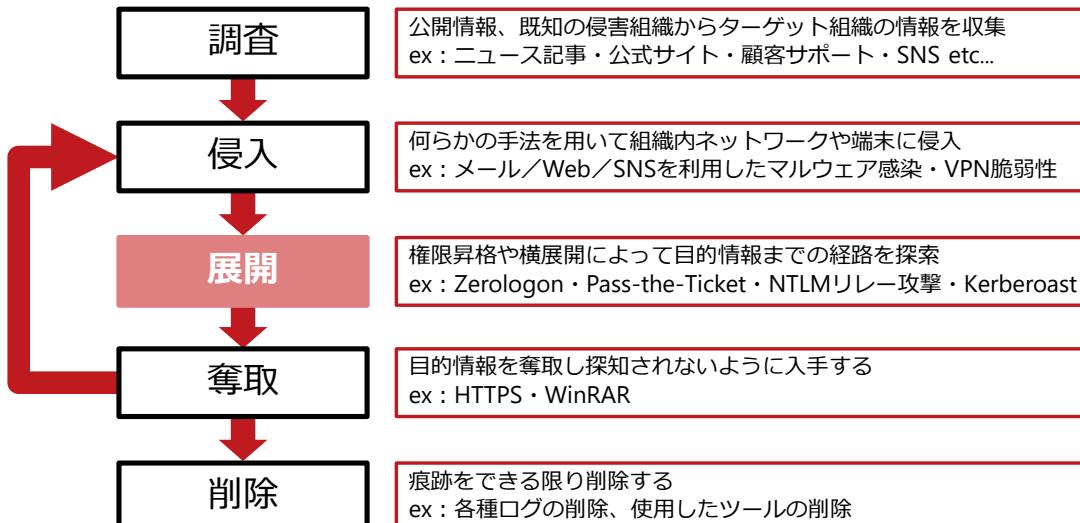
出典 : JPCERT/CC Eyes 「あなたではなく組織の財産を狙うLinkedIn経由のコンタクトにご用心」  
[https://blogs.jpcert.or.jp/ja/2025/01/initial\\_attack\\_vector.html](https://blogs.jpcert.or.jp/ja/2025/01/initial_attack_vector.html)

SNS経由の攻撃の場合、従業員の私用アカウントが攻撃のターゲットになります。

そのため、インシデント発生時の侵入フェーズは社内ネットワークではなく、従業員の自宅など外部で発生する可能性があります。マルウェア感染した端末をVPNで社内ネットワークに接続することで、展開フェーズに移行し社内ネットワークへの侵入が発生します。

未監視領域で発生するインシデントであるため初動検知が難しく、展開フェーズ以降で対処する必要があります。

## 攻撃者のネットワーク侵入の流れ



## 展開

### 管理者権限の奪取・他のシステムへの侵入

- 別のシステムに侵入するために管理者権限を奪取
  - ✓ 見つけたクレデンシャルを別システムでも使う
- 踏み台となっている端末から重要なサーバーへ侵入

### 展開への対策

- 設計段階からセキュリティを意識
  - ✓ 最小権限の原則：余計な権限を持たせない
  - ✓ 公開サーバーと重要な資産を持つサーバーはネットワークで分離しておく
  - ✓ パスワードを使い回さない
  - ✓ 一般端末でDomain Administrator権限（または管理者権限）を持つメインユーザーを使用しない

展開フェーズでは、社内ネットワークでの横展開（Lateral Movement）が行われます。  
このフェーズにもさまざまな攻撃手法があり、以降でいくつか取り上げて説明します。

## 展開

### 攻撃者がネットワーク内を探索するために使用する手法

#### ネットワークスキャン

- NmapやPingコマンドが有名
- その他にも、GitHub上で公開されているさまざまなツールをネットワーク内に持ち込んでスキャンを行う

#### Netコマンド

- ドメイン内のユーザー情報や端末情報を取得できるWindows標準コマンド
- 探索以外にも別システムへの接続など、さまざまな攻撃フェーズで使用される

展開フェーズで最初に実施されるのは、侵入可能な他の端末を探索することです。

これには、Windowsにデフォルトで使用可能なPingコマンドやNmap、それ以外にもGitHub上で公開されているさまざまなツールが使用されます。

そして、Windowsネットワークの探索にはWindowsにデフォルトで使用可能なNetコマンドが多用されます。このコマンドを使用すれば、ドメイン内のアカウントや端末情報を一覧することができます。

さらに、Netコマンドを使えば他の端末へのアクセスも可能になるので、攻撃者はあらゆる場面でNetコマンドを多用しています。

攻撃者が悪用するWindowsコマンド: <https://blogs.jpcert.or.jp/ja/2015/12/wincommand.html>

## 展開に使用される攻撃手法

### Windows／AD環境における攻撃手法

- システム内のパスワードを記載したファイルの奪取・共通アカウント
- Pass-The-Hash／Ticket
- Kerberoast
- 脆弱性（Zerologonなど）
- NTDSダンプ

### Linux環境における攻撃手法

- Kernel Exploit
- ブルートフォース攻撃（SSHへのログイン）

ターゲットとなる端末が見つかったら、次はその端末への侵入が試みられます。

このフェーズで最も多いのが、共通アカウントによる侵害です。

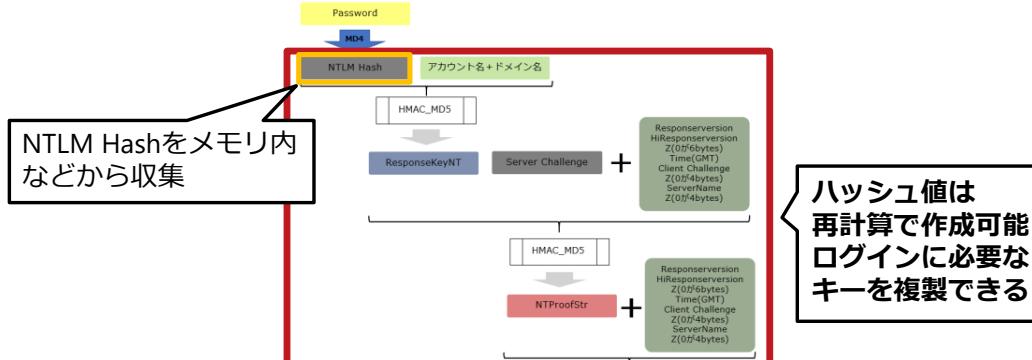
キッティングのためにすべての端末で同じドメインアドミンアカウントでログインしていた場合、その端末に残っているアカウントを使って別の端末にもログインできてしまいます。

また、システム内にパスワードを記載したスクリプトなどが残っていたことでパスワードが漏えいし、侵入されることもあります。

## Pass-the-Hash

### Pass-the-Hashとは

- ハッシュ化されたパスワード情報を盗み、それを使用して同じネットワーク上に新しいユーザーセッションを作成し、ログインする攻撃



出典：三井物産セキュアディレクション Security Knowledge 「パスワードってどこにあるの？その1」  
<https://www.mbsd.jp/research/20190514/password1/>

© 2026 JPCERT/CC Japan Computer Emergency Response Team Coordination Center JPCERT/CC®

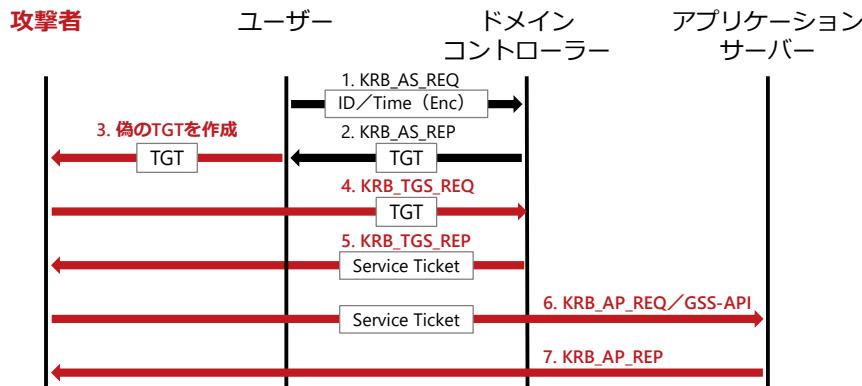
Pass-the-Hashは、NTLMが有効な環境で使用可能な攻撃手法です。

メモリ内などから、ハッシュ化されたパスワード情報を盗み、それを使用して同じネットワーク上に新しいユーザーセッションを作成し、ログインする攻撃です。NTLMを無効化することが対策となります。

## Pass-the-Ticket

### Pass-the-Ticketとは

- Kerberos認証で使用される認証チケットを偽装し、それを使用して同じネットワーク上に新しいユーザーセッションを作成し、ログインする攻撃



43 | © 2026 JPCERT/CC

Japan Computer Emergency Response Team Coordination Center

JPCERT CC®

Pass-the-Ticketは、Kerberos認証で使用される正規の認証チケットをメモリなどから窃取し、正規のユーザーになりすまして攻撃を行います。

この偽の認証チケットを作成する際に使用するチケットの種類によって、以下の2種類の偽チケットが存在します。

- Golden Tickets : TGTを不正に作成する（ドメイン管理者権限でなりすましが可能）
- Silver Ticket : Service Ticketを不正に作成する（各サーバー管理者の権限でなりすましが可能）

どちらのチケットの場合も有効期間が10年と長いため、一度発行されてしまうと強制的にリセットするまで継続的に悪用されてしまいます。なお、パスワードのリセットを2回行うことで、チケットが無効になります。

<https://learn.microsoft.com/ja-jp/windows-server/identity/ad-ds/manage/forest-recovery-guide/ad-forest-recovery-reset-the-krbtgt-password>

# NTDSダンプ<sup>®</sup>

## NTDSダンプとは

- 認証情報が含まれるNTDS.ditデータベースファイルをダンプ・解析することで、認証情報を窃取する
- NTDS.ditなどは、ボリュームシャドウコピー（VSS）からコピー可能

```
# NTDS.ditとレジストリハイブをVSSからコピー
```

```
vssadmin create shadow /for=C:  
copy ¥¥?¥GLOBALROOT¥Device¥HarddiskVolumeShadowCopy2¥Windows¥NTDS¥NTDS.dit C:¥temp¥  
copy ¥¥?¥GLOBALROOT¥Device¥HarddiskVolumeShadowCopy2¥Windows¥System32¥config¥SYSTEM  
C:¥temp¥
```

```
# NTDS.ditからパスワードハッシュを抽出
```

```
secretsdump.py -system SYSTEM -security SECURITY -ntds ntds.dit LOCAL
```

NTDSは、ドメインコントローラに保存されている認証情報などが保存されたデータベースファイルです。このファイルを持ち出して解析することで、認証情報を入手することができます。通常、NTDSにはアクセスできないため、攻撃者はVSSからファイルを抽出しようとします。

# Kerberoast (Kerberoasting攻撃)

## Kerberoastとは

- サービスチケットからパスワードを解析する攻撃
  - ✓ サービスチケットの暗号化キーはService Master Key
  - ✓ Service Master KeyはNTLMハッシュ値をもとに計算
  - ✓ オフラインで総当たり解析
  - ✓ サービスとひも付いたアカウントのパスワードが脆弱だと乗っ取られる



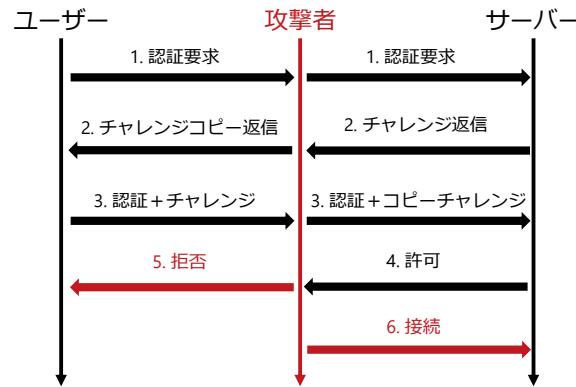
出典：三井物産セキュアディレクション Security Knowledge 「パスワードってどこにある？その2」  
<https://www.mbsd.jp/research/20190520/password2/>

→ ネットワーク上は正常なやり取りなので検知不可能  
(ローカルではLSASSメモリからダンプするためイベントログが残る)

# NTLMリレー攻撃

## NTLMリレー攻撃とは

- サーバーとクライアント間のチャレンジレスポンスを窃取し、本来のクライアントに代わって認証を取得する中間者攻撃



NTLMリレー攻撃は、NTLM認証が有効な環境で使用可能な攻撃で、認証時のやり取りを中継してなりますものです。

## 認証情報取得ツール : pwdump

### パスワードハッシュを取得するツール

- ローカル管理者からドメインユーザーへの横展開
- NTLMハッシュ値を取得し、Pass-the-Hash攻撃につなげる

### 検知

- 顕著なログは残らない
- プロセスの生成と終了から追跡するしかない
- 参考
  - [https://jpcertcc.github.io/ToolAnalysisResultSheet\\_jp/details/PwDump7.htm](https://jpcertcc.github.io/ToolAnalysisResultSheet_jp/details/PwDump7.htm)

ここまでに説明した攻撃手法を実際に実施するツールがいくつか公開されています。

Pwdumpは、端末のNTLMハッシュ値を取得するツールです。

## 認証情報取得ツール : Mimikatz

### パスワードやチケットの取得に使われるツール

- ローカル管理者からドメインユーザーへの横展開
- NTLMハッシュの取得
- ゴールデン／シルバーチケットの作成
- 個人証明書のダンプ
- SAM／SYSTEMの解析

### 検知

- それぞれの動作によって検知されるイベントが異なる
- 参考
  - [https://jpcertcc.github.io/ToolAnalysisResultSheet\\_jp/details/Mimikatz\\_lsadump-sam.htm](https://jpcertcc.github.io/ToolAnalysisResultSheet_jp/details/Mimikatz_lsadump-sam.htm)
  - [https://jpcertcc.github.io/ToolAnalysisResultSheet\\_jp/details/Mimikatz\\_sekurlsa-logonpasswords.htm](https://jpcertcc.github.io/ToolAnalysisResultSheet_jp/details/Mimikatz_sekurlsa-logonpasswords.htm)

Mimikatzは、Pass-the-HashやPass-the-Ticketが可能なツールです。

## 認証情報取得ツール : Rubeus

### アカウント検索やPW総当たりのツール

- 偽装チケットの作成
- Kerberoast可能なアカウントの検索
- 各アカウントへのパスワードブルートフォース

### 検知

- それぞれの動作によって検知されるイベントが異なる
  - ✓ ログイン試行や成功 : イベントID 4624
  - ✓ TGT要求 (ただし正常系と区別できない) : イベントID 4768
  - ✓ ST要求 : イベントID 4769

Rubeusは、Kerberoastが可能なツールです。また、偽のチケットの作成も可能です。

これらの各ツールの実行時には、イベントログが記録されます。ただし、これらの攻撃手法は通常のWindowの認証の仕組みを使用しているので、通常のログイン認証との差が分かりづらいという特徴があります。

## 認証情報はどこに保存されているのか？

### NTDS

- C:\Windows\NTDS\ntds.dit
- ドメインコントローラーのデータベース
- ドメイン中のすべてのユーザーの認証情報が保管されている

### SAM

- ファイル : %SystemRoot%\system32\config\SAM
- レジストリ : HKLM\SAM
- 各端末の認証情報を保管するファイルおよびレジストリ

### メモリ内

- LSASS.exe (認証をつかさどるアプリケーション) のメモリデータ
- Mimikatz、Procdumpなどさまざまなツールで取得可能

認証情報はWindows上のさまざまなものに記録されています。

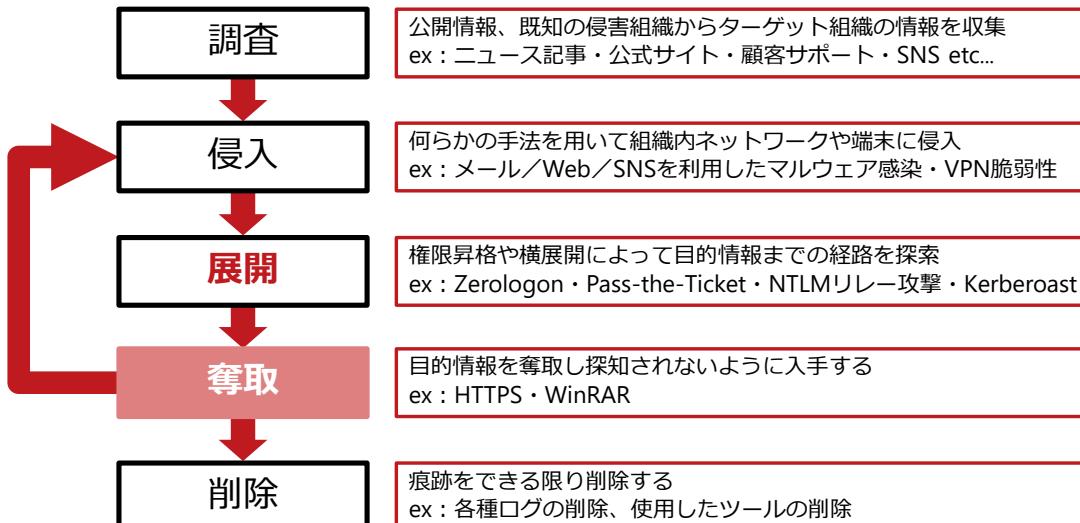
NTDSファイル以外にもレジストリやプロセスのメモリ内に存在します。

攻撃者は、先ほど紹介したツールを使用しなくても、レジストリにアクセスしたりメモリダンプを取得することでこれらの認証情報を得る可能性があります。

メモリダンプはWindowsが提供しているProcdumpというツールで取得でき、また、レジストリもWindowsの標準コマンドであるRegコマンドで取得できます。

このような、攻撃用のツールを使用しない方法は検知しづらいという特徴があります。

## 攻撃者のネットワーク侵入の流れ



## 奪取

侵入のための情報や資産価値のある情報を奪取

□ ドキュメント類

- ✓ 顧客情報
- ✓ 製品の開発情報

□ クラウドサービスへのアクセス情報

□ データベースのデータ

- ✓ 口座や決済の情報

奪取への対策

□ 攻撃者情報持ち出しを検知するのは不可能

- ✓ 一次的な外部への通信量の増加などを検知することができればいいが、通常の運用では困難

□ 重要情報の隔離、ネットワーク分離が重要

横展開の結果、目的の端末に侵入できたら、攻撃者は情報窃取を試みます。

## WinRAR

### なぜ攻撃者はWinRARを使用するのか

- 攻撃者は、WinRAR（正規ファイル）を侵入した端末にダウンロードして、RARファイルに圧縮を行う
- WinRAR自体は異なるファイル名に変更されているので、ファイル名だけでは特定が難しい
- 攻撃者が、WinRARを使うのには以下の理由が考えられる
  - ✓ 内部のドキュメント群を一斉に持ち出すためには、ファイル一式を圧縮して一つのファイルにする方が良い
  - ✓ なるべく圧縮率が高い手法を用いた方が外部への転送量が抑えられる

→ 転送後のRARファイルは、削除されることが多い  
事後のフォレンジック調査で、RARファイルの存在が判明することが多い

攻撃者が外部に情報を持ち出す際によく利用するツールにWinRARがあります。

攻撃者は、このツールを使用してドキュメントなどをRAR形式に圧縮して外部に送信します。

RAR形式に圧縮されたファイルは、多くの場合、攻撃者によって削除されるので、どのファイルが持ち出されたかを特定することは困難です。

攻撃者が、WinRARを実行する際のコマンドラインを何らかの手段で記録していれば、そこから攻撃者の持ち出したファイルが推測することができます。

## データの外部持ち出し方法

### マルウェアの機能を使用

- 多くのマルウェアはファイルを転送する機能を持っており、その機能を使って外部にファイル転送する

### SSH、HTTPS、SOCKS5のトンネリング通信

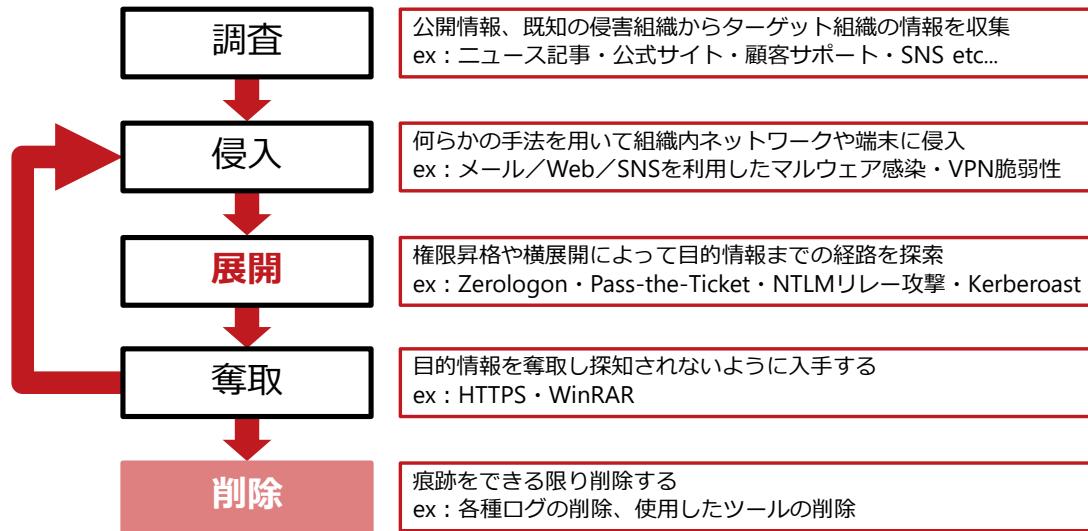
- トンネリングツールを利用して、ネットワーク上許可された経路でファイルを外部に転送する

### クラウドサービスへの送信

- 正規のクラウドサービスにファイルを転送する

データを外部に転送する際は、マルウェアの機能を使用する場合、トンネリングツールを使用する場合、クラウドサービス直接アップロードする場合などがあります。攻撃者は、プロキシ経由で通信できるように準備をしているため、奪取フェーズまで到達してしまうとデータの持ち出しを防ぐことは困難です。

## 攻撃者のネットワーク侵入の流れ



## 削除

### 侵入した痕跡を削除する

- 侵入し続ける
- 発覚を遅らせる
- 発覚後の検査を遅延させる

### 削除への対策

- ファイルの削除は通常操作と変わりないので、検知不可能
  - Windows標準コマンドdelなどが使用される
- ログの削除は通常操作では行わないため、検知できる可能性がある
  - イベントログの削除は、**イベントID：1102**で記録される

攻撃者は、攻撃終了または攻撃の途中に侵入の痕跡を削除します。

削除するものは、使用したツール・ログ・作成したファイルなどです。

削除する際は、マルウェアの機能や、デフォルトのコマンドで使用可能なコマンドが使用されます。

また、イベントログの削除にはWindowsの標準コマンドであるWevtutilが使用されます。その際は、イベントID：1102で記録されます。

## 参考：コマンド実行のログ

攻撃者はコマンドラインを使いこなす

- 調査／侵入／展開／奪取／削除すべてのフェーズで使用
- Windowsにおいては、PowerShellまたはCMDが多用される
- 攻撃者の挙動はコマンドプロンプト／シェルのログを見れば把握可能

Windows (PowerShell) のログ確認 ※ コマンドプロンプトのログは残らない

```
> type (Get-PSReadlineOption).HistorySavePath -Tail 20  
> (Get-PSReadlineOption).HistorySavePath  
> C:\Users\[UserName]\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt
```

Linuxのログ確認

```
# history  
# cat $HISTFILE
```

## 参考：コマンド実行のログ（Bash）

実行したコマンドが毎行記録される

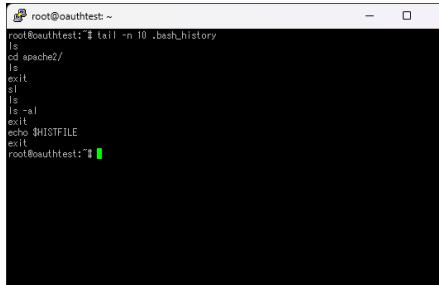
- 日時や出力は記録されない

攻撃者のアクティビティを把握する  
手掛かりになるが、注意が必要

- 同一権限で削除可能
- 記録されない利用法あり
- 偽装も容易

攻撃者自身にログを閲覧される危険性

- サーバーの役割
- 機密ファイルのパス
- コマンドで渡したパスワード

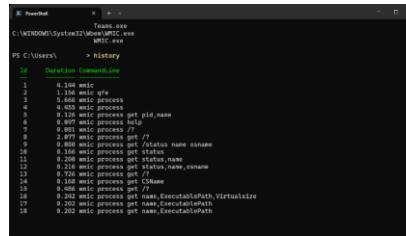


```
root@oauthtest:~$ tail -n 10 .bash_history
ls
cd apache2/
ls
exit
st
ls
ls -al
exit
echo $HISTFILE
exit
root@oauthtest:~$
```

## 参考：コマンド実行のログ（PowerShell）

Historyコマンドとファイルの双方に  
記録が残る

- 日時等の情報は無い
- プロセス終了後も残るのはファイルのみ
- 一般ユーザーは使わないので攻撃者のログが  
大半になる可能性が高い
- デフォルトで4,096行分記録される



```
PS C:\Users\> history
Get-History -Count 30
1 0.246 msic qf
2 1.156 msic qf
3 0.206 msic process
4 0.455 msic process
5 0.206 msic process get_id_name
6 0.897 msic process help
7 0.206 msic process get /
8 0.877 msic process get /
9 0.206 msic process get /?msic_name osname
10 0.166 msic process get /status
11 0.206 msic process get /status_name
12 0.206 msic process get /status_name,osname
13 0.726 msic process get /
14 0.206 msic process get /?
15 0.486 msic process get /
16 0.206 msic process get /?msic_name,ExecutionPath,VirtualAlloc
17 0.202 msic process get /?msic_name,ExecutionPath,VirtualAlloc
18 0.202 msic process get /?msic_name,ExecutionPath,VirtualAlloc
19 0.202 msic process get /?msic_name,ExecutionPath,VirtualAlloc
20 0.202 msic process get /?msic_name,ExecutionPath,VirtualAlloc
21 0.202 msic process get /?msic_name,ExecutionPath,VirtualAlloc
22 0.202 msic process get /?msic_name,ExecutionPath,VirtualAlloc
23 0.202 msic process get /?msic_name,ExecutionPath,VirtualAlloc
24 0.202 msic process get /?msic_name,ExecutionPath,VirtualAlloc
25 0.202 msic process get /?msic_name,ExecutionPath,VirtualAlloc
26 0.202 msic process get /?msic_name,ExecutionPath,VirtualAlloc
27 0.202 msic process get /?msic_name,ExecutionPath,VirtualAlloc
28 0.202 msic process get /?msic_name,ExecutionPath,VirtualAlloc
29 0.202 msic process get /?msic_name,ExecutionPath,VirtualAlloc
30 0.202 msic process get /?msic_name,ExecutionPath,VirtualAlloc
```

Windowsイベントログにも残る

- デフォルトでは不完全
- エラー等からコマンドの全容を探る



1 社内ネットワーク基礎

2 社内ネットワークへの攻撃手順

3 Windowsイベントログ

4 Windowsイベントログの分析

5 ハンズオン

## Windowsイベントログ

### Windowsにおけるログ

- OS内のいろいろな動作を記録
  - ✓ メインは、Security・Application、System
- AD／端末で同一形式

### 保存フォルダー

- %SystemRoot%¥System32¥winevt¥Logs¥

### 拡張子

- EVTX

### ログ確認方法

- イベントビューアーを使用するのがもっと簡単な方法

イベントログは、Windowsのログ機能です。

## イベントビューアー

### Windowsデフォルトのイベントログビューアー

- Windows Client／Server双方に標準搭載されている
- 動作が重い
- 最低限の検索機能がある

### 簡単な調査に限定して利用する

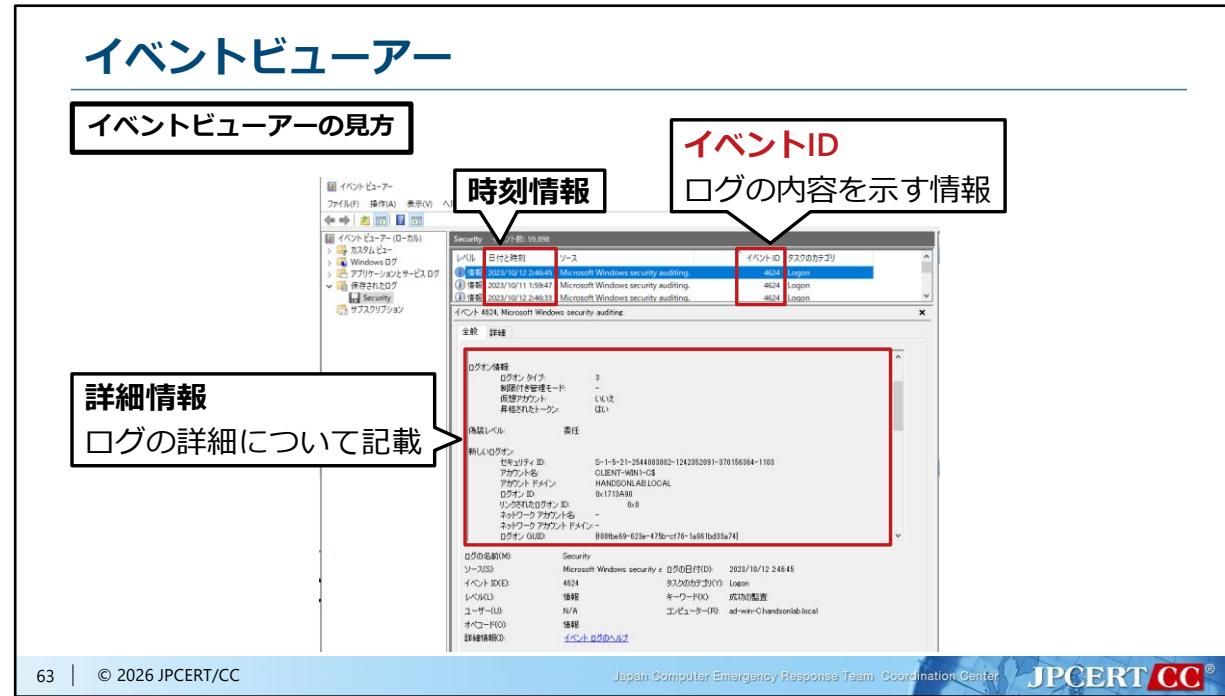
- 時刻やイベント（攻撃者の挙動）が明確な状態からの追跡に使う
- 大量のログを分析する用途には向かない**（専用の分析ツールを使用する）

イベントログを確認するツールがイベントビューアーです。

イベントビューアーには検索などの機能がありますが、動作が重いため大量のログを分析するという用途には向きません。

## イベントビューアー

### イベントビューアーの見方



イベントログには、各ログにイベントIDがつきます。このIDによってログの内容を識別できます。  
また、イベントログの詳細情報は、各イベントIDによってフォーマットが異なります。

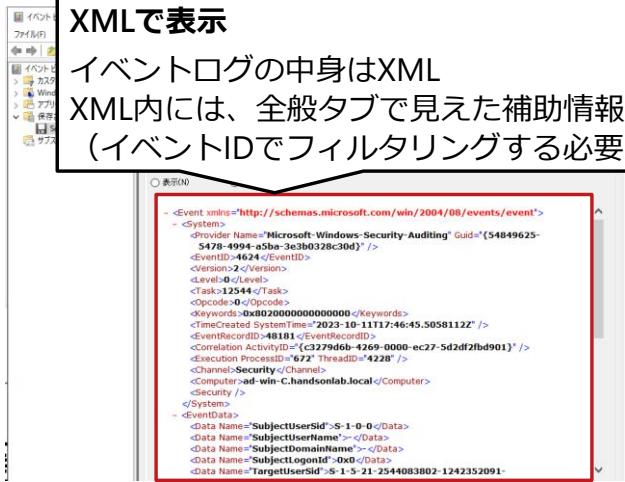
# イベントビューアー

## イベントビューアーの見方

### XMLで表示

イベントログの中身はXML

XML内には、全般タブで見えた補助情報がなくなる  
(イベントIDでフィルタリングする必要がある)



```
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-a5ba-3e3bd328c30d}" />
    <EventID>4624</EventID>
    <Version>2</Version>
    <Level>1</Level>
    <Task>134</Task>
    <Opcode>0</Opcode>
    <Keywords>0x0200000000000000</Keywords>
    <TimeCreated SystemTime="2023-10-11T17:46:45.505811Z" />
    <EventRecordID>4031</EventRecordID>
    <Correlation ActivityID="C2794db-4269-0000-ec27-5d2df2fb901" />
    <Execution ProcessID="672" ThreadID="4228" />
    <Channel>Security</Channel>
    <Computer>ad-win.C.hands-on-lab.local</Computer>
    <Security />
  </System>
  <EventData>
    <Data Name="SubjectUserSid">S-1-0-0</Data>
    <Data Name="SubjectUserName"></Data>
    <Data Name="SubjectDomainName"></Data>
    <Data Name="SubjectLogonId">0x0</Data>
    <Data Name="TargetUserSid">S-1-5-21-2544083002-1242352091</Data>
  </EventData>

```

イベントログの中身はXMLで表示およびエクスポートすることが可能です。ただし、XML内には全般タブで見えた日本語の補助情報がないため、どのイベントIDにどのような意味があるのかを把握していかなければなりません。

## 重要なWindowsイベントログ①

### アカウント関連 (Security)

- イベントID 4624 : アカウントが正常にログオンしました
- イベントID 4625 : アカウントがログオンに失敗しました
- イベントID 4768 : Kerberos 認証チケット (TGT) が要求されました
- イベントID 4769 : Kerberos サービス チケットが要求されました
- イベントID 4776 : コンピューターがアカウントの資格情報を検証しようとした
- イベントID 4672 : 新しいログオンに割り当てられた特別な特権

### 確認ポイント

- 大量のログオン失敗
- 意図しないアカウント作成・管理者アカウントの作成
- 意図しない特権アカウントでのログイン
- 意図しないリモートログイン（普段は行わないRDPからのログインなど）

イベントログを分析する際に重要なイベントをいくつか紹介します。

まずは、Securityイベントのアカウント関連のイベントログです。

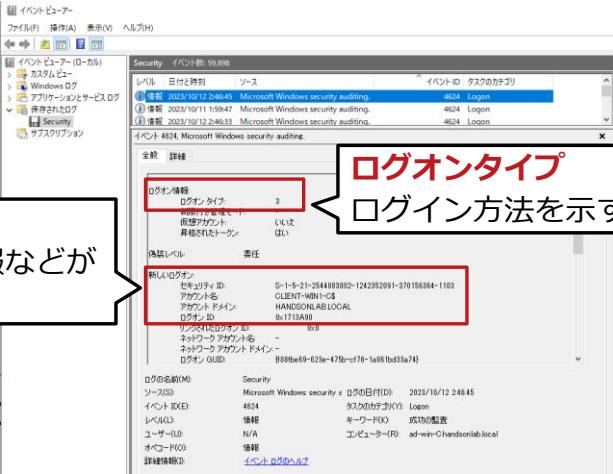
これらのイベントログは通常のログイン時にも発生しますが、攻撃者のログイン時にも記録されます。

## 重要なWindowsイベントログ

### アカウント関連 (Security)

**詳細情報**  
アカウント情報などが記載される

**ログオンタイプ**  
ログイン方法を示す情報



ログオンタイプを確認することで、どの手段でログインしたかを知ることができます。

## ログオン種別

ログオンの種類	ログオン タイトル	説明
0	System	システムの起動時など、システム アカウントでのみ使用されます。
2	Interactive	ユーザーがこのコンピューターにログオンしました。
<b>3</b>	<b>Network</b>	ネットワークからこのコンピューターにログオンしたユーザーまたはコンピューター。
4	Batch	バッチ ログオンの種類はバッチ サーバーによって使用され、そこではプロセスが直接介入せずにユーザーの代わりに実行される可能性があります。
5	Service	サービス コントロール マネージャーによってサービスが開始されました。
7	Unlock	このワークステーションのロックが解除されました。
8	NetworkCleartext	ユーザーがネットワークからこのコンピューターにログオンしました。ユーザーのパスワードは、非ハッシュ化形式で認証パッケージに渡されました。組み込みの認証では、ネットワーク経由で送信する前に、すべてのハッシュ資格情報がパッケージ化されます。資格情報は、ブレーンテキスト（クリア テキストとも呼ばれます）でネットワークを通過しません。
9	NewCredentials	送信元が現在のトークンを複製し、送信接続用に新しい資格情報を指定しました。新しいログオン セッションのローカルIDは同じですが、他のネットワーク接続には異なる資格情報を使用します。
<b>10</b>	<b>RemoteInteractive</b>	ターミナル サービスまたはリモート デスクトップを使用してリモートでこのコンピューターにログオンしたユーザー。
11	CachedInteractive	コンピューターにローカルに保存されたネットワーク資格情報を使用してこのコンピューターにログオンしたユーザー。資格情報を確認するために、ドメイン コントローラーに接続できませんでした。
12	CachedRemoteInteractive	RemoteInteractiveと同じです。これは、内部監査に使用されます。
13	CachedUnlock	ワークステーション ログオン。

出典 : Microsoft 「4624(S): An account was successfully logged on.」をJPCERT/CCが翻訳  
<https://learn.microsoft.com/ja-JP/previous-versions/windows/it-pro/windows-10/security/threat-protection/auditing/event-4624>

攻撃者のログインの場合、通常はリモートからのログインとなるため、ログオンタイプは3または10になることが多いです。

## 重要なWindowsイベントログ②

### プロセス関連 (Security)

- イベントID 4688 : 新しいプロセスが作成されました
- イベントID 4689 : プロセスが終了しました
- イベントID 5154 : Windows フィルタリング プラットフォームで、アプリケーション  
またはサービスによるポートでの着信接続のリッスンが許可されました
- イベントID 5156 : フィルタリング プラットフォームによる接続の許可

### 確認ポイント

- 通常と異なるプロセスの生成／終了を検知する
  - ✓ 深夜にPowerShellが立ち上がっている
  - ✓ 知らないドメインに通信している

→ これらのログは、デフォルト設定では記録されない  
監査ポリシーの設定をする必要がある

次に、プロセス関連のイベントログです。

イベントログでは、プロセスの実行や終了を記録することができます。

ただし、この機能は監査ポリシーと呼ばれる機能で、デフォルトでは無効化されています。

## 重要なWindowsイベントログ③

### Windows Defender (Microsoft-Windows-Windows Defender Operational)

- イベントID 1013 : マルウェアやその他の望ましくない可能性のあるソフトウェアの履歴を削除しました
- イベントID 1150 : エンドポイント保護クライアントは正常に稼働しています
- イベントID 1151 : エンドポイント保護 クライアントの正常性レポート
- イベントID 5001 : リアルタイム保護が無効になっています
- イベントID 5007 : Microsoft Defender ウイルス対策 の構成が変更されました

#### 確認ポイント

- Windows Defenderをオフにしたり、マルウェアを検知した際に残る
- ノイズが少なく攻撃の全容もつかめる
- 使用しているウイルス対策ソフトで検知できなかったファイルを検知している可能性がある

Windows Defenderのログもイベントログとして記録されます。

# 監査ポリシー

## 監査ポリシーとは

- Windowsに標準で搭載されているログオン・ログオフやファイルアクセスなどの詳細なログを取得するための設定
- アカウント関連の監査ログは有効にしておくことを推奨

## 監査ポリシー使用の注意点

- 監査ポリシーを有効にすることで、**ログが増加する**
  - ✓ ログのローテーションが早くなり古いログが残りにくくなる
- 監査ポリシーを有効化する場合は、**イベントログの最大サイズの変更**もあわせて検討
  - ✓ イベントビューアーやwevtutilコマンドで変更可能

監査ポリシーは、Windowsに標準で搭載されているログオン・ログオフやファイルアクセス、プロセスの実行・終了などの詳細なログを取得するための設定で、デフォルトでは無効化されています。

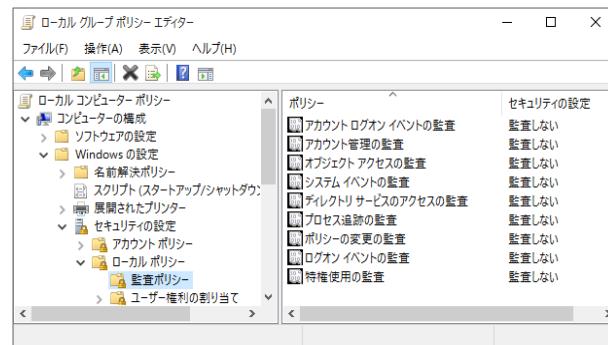
監査ポリシーを有効化すると、ログの量が増加しますが、インシデント発生時に有用なログが記録されることになるため、有効化することを推奨します。

## 参考：監査ポリシーの有効化方法

### 設定方法 ①

#### □ ローカルグループ ポリシーの編集

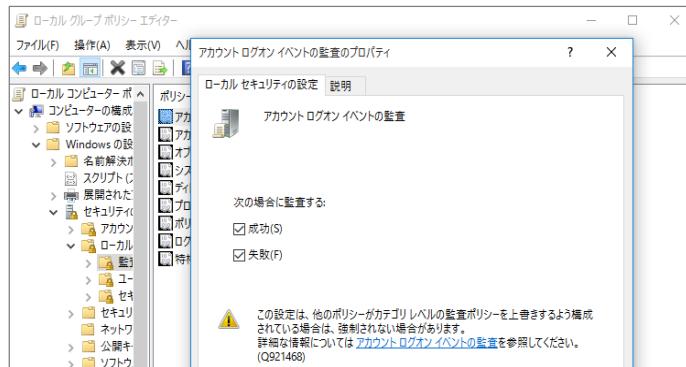
- ✓ [コンピューターの構成] → [Windowsの設定] → [セキュリティの設定] → [ローカル ポリシー]→ [監査ポリシー]



## 参考：監査ポリシーの有効化方法

### 設定方法 ②

- 各ポリシーの「成功」「失敗」を有効



## Sysmon

### Sysmonとは

- 監査ログと同じく、デフォルトでは取得できないWindows上のアクティビティをログとして保存することができるマイクロソフトの提供するツール
- 以下のアクティビティをログに記録できる
  - ✓ ファイル作成・削除
  - ✓ プロセス起動・終了・インジェクション関連
  - ✓ レジストリ操作
  - ✓ DNS通信
  - ✓ ネットワーク通信
  - ✓ WMIイベント
  - ✓ ドライバー読み込み
- <https://learn.microsoft.com/ja-jp/sysinternals/downloads/sysmon>

Sysmonも、監査ポリシーと同じくWindows上の詳細なログを残すために有用なツールです。  
このツールはデフォルトではインストールされていないため、インストールする必要があります。  
また、監査ポリシーと同じくログ量が多いため、不要なログをチューニングすることが重要です。

Sysmonの設定情報に関しては、以下をご参照ください。  
<https://github.com/SwiftOnSecurity/sysmon-config>

## イベントログ分析のポイント

### ポイント

- イベントログは、見る必要がない大量のログが記録されているので、ある程度絞り込みを行う必要がある

### ログ絞り込みのポイント

- 見るイベントIDを特定する
  - ✓ 攻撃時にどのようなイベントIDが記録されるのかを理解する
- インシデント発生時刻前後に絞り込む
- ログ分析ツールを使用する
  - ✓ Splunk
  - ✓ Microsoft Sentinel
  - ✓ LogonTracer (OSS)
  - ✓ Hayabusa (OSS)

イベントログには、見る必要がない大量の正常なログが記録されているので、ある程度絞り込みを行う必要があります。

そのためには、①どのイベントIDを見る必要があるのかというのを把握しておくことが重要です。

さらに、イベントビューアーで分析が困難な場合（ログ量が多いとき）は②別のツールを使って分析することを検討する必要があります。

## 攻撃時にどのようなイベントIDが記録されるのか？

### ポイント

- ブログやレポートなどで、攻撃時に記録されたイベントログの情報を知る

### 参考

- 侵入型ランサムウェア攻撃発生時に残るWindowsイベントログの調査  
<https://blogs.jpcert.or.jp/ja/2024/09/windows.html>
- ツール分析結果シート  
[https://jpcertcc.github.io/ToolAnalysisResultSheet\\_jp/](https://jpcertcc.github.io/ToolAnalysisResultSheet_jp/)
- Operation Blotless攻撃キャンペーンに関する注意喚起  
<https://www.jpcert.or.jp/at/2024/at240013.html>
- 攻撃グループMirrorFaceの攻撃活動  
<https://blogs.jpcert.or.jp/ja/2024/07/mirrorface.html>

①どのイベントIDを見る必要があるのかについては、さまざまなドキュメントが公開されていますのでご覧ください。  
これ以外にも、セキュリティベンダーなどからさまざまなイベントIDに関する文書が出ていますので、そちらもご確認ください。

## 参考：攻撃グループMirrorFaceの攻撃活動

### (4) ファイアウォールのルール追加

- Windows Firewallの除外リストに、NOOPDOORで使用する特定ポート宛ての通信を許可する設定を追加
- イベントログ Firewall With Advanced Security/Firewall : イベントID 2004で記録される

### (5) 登録したサービスの隠蔽

- 登録したサービスが表示されないように、アクセス制御を設定

### (6) Windowsイベントログの消去

- システムログの削除
- 各イベントログ : イベントID 1102で記録される

### (7) Windows Defenderの停止

- イベントログ Windows Defender/Operational : イベントID 5001で記録される

出典：JPCERT/CC Eyes「攻撃グループMirrorFaceの攻撃活動」  
<https://blogs.jpcert.or.jp/ja/2024/07/mirrorface.html>

## LogonTracer

### イベントログの分析をサポートするツール

- ▣ イベントログを可視化
- ▣ アカウントのログイン情報を一画面に表示可能
- ▣ 重要性の高いアカウントおよびホストの抽出
- ▣ イベントログのタイムライン表示

### 膨大なログから着眼すべきログを 教えてくれる

- ▣ 怪しいアカウントやホストの“当たりを付ける”  
ためのツール
- ▣ 意図しないアカウントとホストの結び付きを  
見るだけでも良い



出典：JPCERT/CC 「LogonTracer」 <https://github.com/JPCERTCC/LogonTracer/wiki>

②別のツールを使って分析するについては、いくつかのイベントログを分析するツールがあります。

LogonTracerはJPCERT/CCが公開するツールで、ドメインコントローラーのSecurityログを可視化して分析することができます。

# Hayabusa

## イベントログのタイムライン分析を サポートするツール

- イベントログをCSVとして整形
  - イベントビューアーでExportするよりも  
分析しやすいフォーマットで出力可能
  - 分析結果をサマリーとして取得可能

## ルールによる不審なログの検知

- SIGMAルールを使って、怪しいイベントログ検知
  - 見るべきポイントを絞ることができるので分析の効率化に有効

出典 : Yamato Security 「hayabusa」 <https://github.com/Yamato-Security/hayabusa>

1 社内ネットワーク基礎

2 社内ネットワークへの攻撃手順

3 Windowsイベントログ

4 Windowsイベントログの分析

5 ハンズオン

## イベントビューアーでログ分析

### 1. 画面左からソースとなる情報源を選択する

- Securityログの場合：イベントビューアー（ローカル） ➔ Windowsログ ➔ セキュリティ
- その他のログ：イベントビューアー（ローカル） ➔ アプリケーションとサービス ログ ➔ Microsoft ➔ Windows

### 2. 画面右の“検索”や“フィルター”等で絞り込んで目的のログを探す

- 検索：軽いが検索しながらログを見ることができない。一度検索窓を閉じる必要がある
- フィルター：重いがリアルタイムにフィルタリングされたログが表示される
- イベントのIDが定まっている場合はフィルター、ホスト名等で探す場合は検索

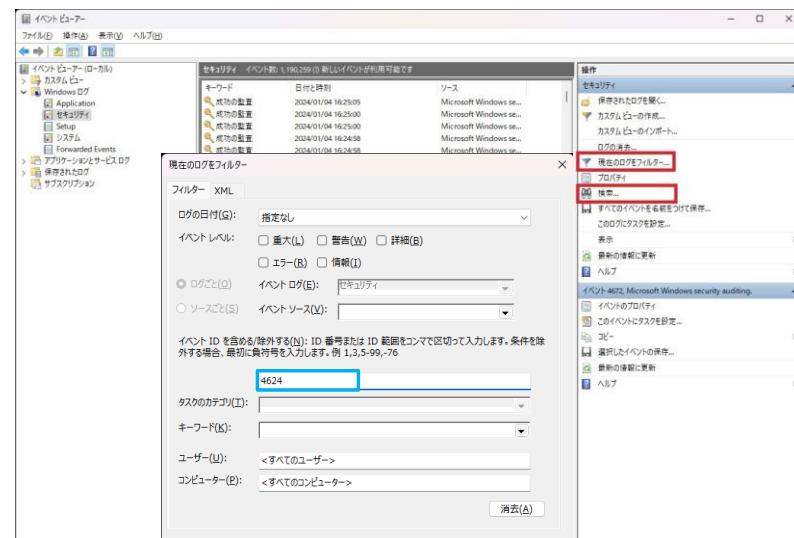
### 3. 複雑な検索等を行う場合はCSV出力してから別ソフトで行う

- 対象のログを右クリック ➔ すべての（or フィルターされた）イベントに名前を付けて  
保存 ➔ ファイルの種類をTXTまたはCSV形式に変更して保存

イベントビューアーでイベントログを分析する流れを示します。

基本的には、イベントビューアーですべてのログを見ることは難しいため、フィルタリングした上でCSV形式などにExportして分析する方法が良いと考えます。

## イベントビューアーのフィルタリング



## よく使うフィルター条件

### 日時検索

- 大量のログを分析するには適さないので、日時である程度絞り込む

### イベントIDの範囲で検索

- 「8000-8010」 8000～8010のイベントを検索
- 「8000,8010」 8000と8010のイベントを検索

### 日時・イベントID以外はXMLタブからXPathによるフィルター

- 分析手順は以下のとおり

  - ① GUI上のフィルターを適用
  - ② XPathによるフィルターを適用
  - ③ テキストとして出力後、ツールやPython等を用いて加工

イベントビューアーを使ったフィルタリング条件で最もシンプルなものは、イベントIDでのフィルタリングです。また、日時で絞り込むのも有効です。これらのフィルタリングはGUIから操作可能なのですが、より複雑なフィルタリングを実施したい場合は、Xpathを使用します。

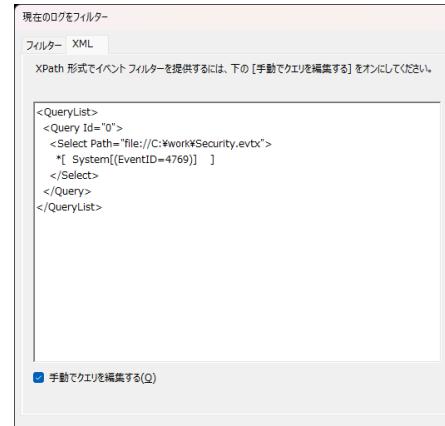
## XPathによるフィルター

イベントビューアーでは、XPath1.0を利用可能

ロイイベントログはXMLの集合体

XPathの一部のみサポート

- contains等の関数は使えない
- 基本的なクエリのみ利用可能
- 演算子も利用可能
- XPath 1.0 の制限事項
  - ✓<https://learn.microsoft.com/ja-jp/windows/win32/wes/consuming-events>



イベントビューアーでは、XPath1.0を利用可能です。XPath1.0については、ドキュメントなどをご覧ください。

イベントIDなどのフィルター条件を指定して、図のようにXMLタブに移動すると、現在しているフィルターのXpathが確認できます。

まずはフィルタータブである程度条件を設定してからXMLタブに移動し、「手動でクエリを編集する」をクリックし、Xpathを編集していくことをお勧めします。

## 参考：XPathの基礎知識

### フィルターパターン1：\*[タグ名[子タグ名]]

- ・フィルター例：\*[exampleA[exampleB[exampleC]]]
- ・フィルター結果：exampleCタグがどちらもヒット

### フィルターパターン2：\*[タグ名[@属性=値]]

- ・フィルター例：\*[exampleC[@id=1]]
- ・フィルター結果：id=1のexampleCタグがヒット

### フィルターパターン3：\*[タグ名=値]

- ・フィルター例：\*[exampleC = "aaa"]
- ・フィルター結果：id=1のexampleCタグがヒット

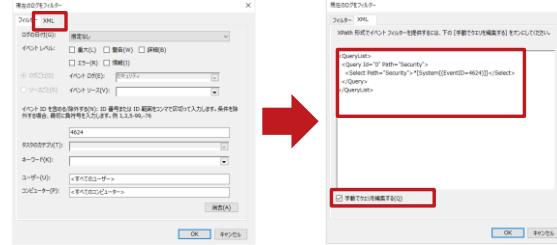
<!-- XML例 -->

```
<exampleA>
  <exampleB>
    <exampleC id=1>aaa
    </exampleC >
    <exampleC id=2>bbb
    </exampleC >
  </exampleB>
</exampleA>
```

## 参考：XPathのフィルタールールの組み立て方

### 1. フィルタータブでログのフィルターを実施

- 自動で検索用XMLに反映される
- 必要な一部だけ書き替える



### 2. イベントログのXMLと見比べる

- 全般／詳細タブを確認
- 何の値を対象にしたいか等

## 参考：XPathのフィルタールールの組み立て方

### フィルター例1

- 任意の要素 \*[] の中から<System>の子の<EventID>の値が4769のもの

```
<QueryList>
  <Query Id="0">
    <Select Path="file:///C:/work/sample\Security.evtx">
      *[ System[EventID=4769] ]
    </Select>
  </Query>
</QueryList>
```

## 参考：XPathのフィルタールールの組み立て方

### フィルター例2

- 任意の要素 \*[] の中から<System>の子の<EventID>の値が4769のもの
- <EventData>の子の<Data>のName属性がTargetUserNameであって、値がdomuser@LAB.LOCALのもの

```
<QueryList>
  <Query Id="0">
    <Select Path="file:///C:/work/Security.evtx">
      *[

        System[EventID=4769] and
        EventData[
          Data[@Name="TargetUserName"] = "domuser@LAB.LOCAL"
        ]
      ]
    </Select>
  </Query>
</QueryList>
```

## 参考：XPathのフィルタールールの組み立て方

### フィルター例3

- 任意の要素 \*[] の中から<System>の子の<EventID>の値が4769のもの
- <TimeCreated>のSystemTime属性値が2024年1月18日～2024年1月31日まで

```
<QueryList>
  <Query Id="0"
    <Select Path="file:///C:/work/Security.evtx">
      *[System[
        (EventID=4611) and
        TimeCreated[
          @SystemTime>='2024-01-18T00:00:00.000Z' and
          @SystemTime<='2024-01-31T00:00:00.000Z'
        ]
      ]]
    </Select>
  </Query>
</QueryList>
```

## PowerShellを使ったイベントログ分析

### Get-WinEventを使ったイベントログ分析

- PowerShellのコマンドGet-WinEventを使うことで、イベントビューアーと同様の分析を行うことが可能
- ただし、コマンドラインで大量のログを分析するのは困難なため、CSVなどにExportして分析する
- イベントビューアーと同じく、XPathによるフィルタリングが可能

イベントビューアー以外にも、PowerShellを使うことで、イベントログのフィルタリングやExportができます。この場合も、イベントビューアーと同様にXPathを使ったフィルタリングが可能です。

## PowerShellを使ったイベントログ分析

### Get-WinEvent例

```
Get-WinEvent -Path C:\test\Security.evtx -FilterXPath
'*[System[(EventID=4624)]' | Select-Object
RecordID,TimeCreated,Id,Message | Sort-Object RecordId |
Export-Csv -Path test.csv -NoTypeInformation -Encoding UTF8
```

Get-WinEventを使ってイベントログを読み込み、XPathでフィルタリングするコマンド例を示しています。  
イベントビューアーとできることはほぼ変わらないため、使いやすい方法をご利用ください。

1 社内ネットワーク基礎

2 社内ネットワークへの攻撃手順

3 Windowsイベントログ

4 Windowsイベントログの分析

5 ハンズオン

ここからハンズオンに入ります。

最初は、4章で説明したフィルタリングの使い方を中心にハンズオンでふれていきます。

## ハンズオン 1

演習

### 問題

- イベントビューアーを使って、sample1.evtxログファイルから、  
ID : 8000-8999までのログをまとめたCSVファイルを作成してください。

# ハンズオン 1

演習

回答

現在のログをフィルター

フィルター XML

ログの日付(G): 指定なし

イベントレベル:  重大(L)  警告(W)  詳細(B)  
 エラー(R)  情報(I)

ログごと(O) イベントログ(E):

ソースごと(S) イベントソース(V):

イベント ID を含める/除外する(N): ID 番号または ID 範囲をコンマで区切って入力します。条件を除外する場合、最初に負符合を入力します。例 1,3,5-99,-76

タスクのカテゴリ(T):

キーワード(K):

ユーザー(U): <すべてのユーザー>

コンピューター(P): <すべてのコンピューター>

## ハンズオン 2

演習

### 問題

- sample1.evtxログファイルから、Microsoft Edgeが起動した時刻を抽出してください。

## ハンズオン 2

演習

### 問題

- sample1.evtxログファイルから、Microsoft Edgeが起動した時刻を抽出してください。

### ヒント

プロセス起動のイベントID

- 4688

Microsoft Edgeのプロセス名

- C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe

プロセス名のフィールドは、`NewProcessName`です。

## ハンズオン 2

演習

### 回答 ※イベントビューアーの場合

```
<QueryList>
<Query Id="0" Path="file:///sample1.evtx">
  <Select Path="file:///sample1.evtx">
    *[

      System[(EventID=4688)] and
      EventData[Data[@Name="NewProcessName"]="C:\Program Files
(x86)\Microsoft\Edge\Application\msedge.exe"]

    ]
  </Select>
</Query>
</QueryList>
```

## ハンズオン 2

演習

回答

The screenshot shows the Windows Event Viewer interface. On the left, there's a navigation pane with options like 'イベントビューアー (ローカル)', 'カスタムビュー', 'Windows ログ', 'アプリケーションとサービス ログ', '保存されたログ' (which is expanded to show 'sample1' and 'サブスクリプション'), and 'サブスクリプション'. The main pane displays a table of events from 'sample1' with 42,628 entries. The first five rows are shown in detail:

レベル	日付と時刻	ソース	イベント ID	タスクのカテゴリ
情報	2023/07/21 18:45:37	Microsoft Windows security auditing.	4688	Process Creation
情報	2023/07/21 18:45:37	Microsoft Windows security auditing.	4688	Process Creation
情報	2023/07/21 18:45:36	Microsoft Windows security auditing.	4688	Process Creation
情報	2023/07/21 18:45:34	Microsoft Windows security auditing.	4688	Process Creation

Below the table, a tooltip for event 4688, 'Microsoft Windows security auditing.', is displayed, showing the XML structure:

```
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
```

## ハンズオン 2

演習

回答 ※PowerShellの場合

```
Get-WinEvent -Path C:\sample1.evtx -FilterXPath
'*[System[(EventID=4688)] and
EventData[Data[@Name="NewProcessName"]='C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe']]' |
Select-Object RecordID,TimeCreated,Id,Message | Sort-
Object RecordId | Export-Csv -Path test.csv -
NoTypeInformation -Encoding UTF8
```

## ハンズオン 3

演習

### 問題

- sample1.evtxログファイルから、アカウント : jpcertuserでRDP経由でログインしたログをフィルターしてCSVファイルを作成してください。

## ハンズオン 3

演習

### 問題

- sample1.evtxログファイルから、アカウント名“jpcertuser”でRDP経由でログインしたログをフィルターしてCSVファイルを作成してください。

### ヒント

ログオン成功のイベントID

- イベントID : 4624

RDP接続のLogonType

- 10

アカウント名のフィールドは`TargetUserName`  
LogonTypeのフィールドは`LogonType`

## ハンズオン 3

演習

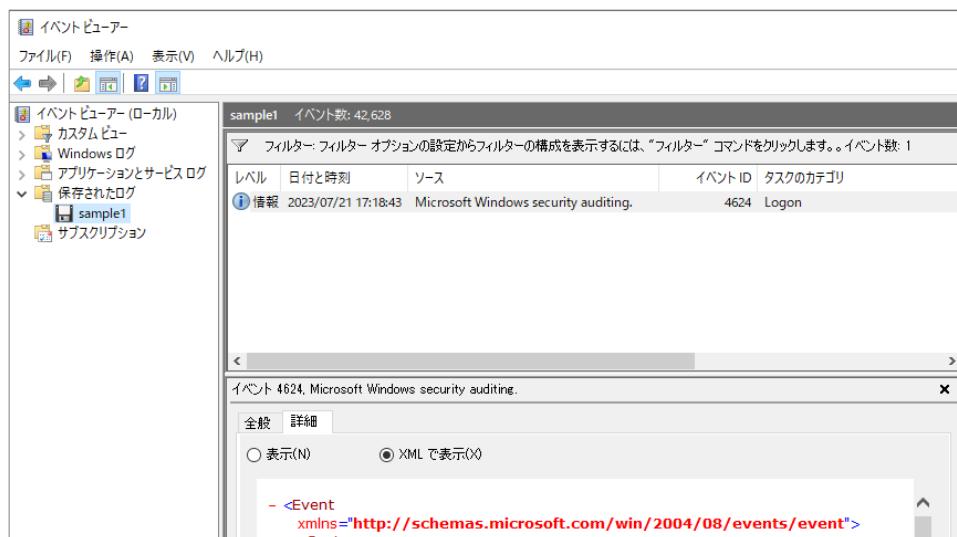
### 回答

```
<QueryList>
  <Query Id="0" Path="file:///sample1.evtx">
    <Select Path="file:///sample1.evtx">
      *[[
        System[(EventID=4624)] and
        EventData[Data[@Name="LogonType"]="10" and
          Data[@Name="TargetUserName"]="jpcertadmin"
        ]
      ]]
    </Select>
  </Query>
</QueryList>
```

## ハンズオン 3

演習

回答



## ハンズオン 4

演習

### 問題

sample2.evtxはバックドアツールがインストールされていた端末のイベントログです。通常このツールはWindows Defenderに検知されるはずでしたが、この時は検知されませんでした。誰がDefenderを切ったのか、特定してください。

## ハンズオン 4

演習

### 問題

sample2.evtxはバックドアツールがインストールされていた端末のイベントログです。通常このツールはWindows Defenderに検知されるはずでしたが、この時は検知されませんでした。誰がDefenderを切ったのか、特定してください。

### ヒント

Windows Defenderの停止時間を特定

イベントID : 5001

Windows Defenderの停止時間周辺でログインしているアカウントを特定

Windows Defenderを停止させたアカウント情報は記録されないので、Windows Defenderの停止時間周辺でログインしているアカウントがその操作をした可能性があると推測できます。

## ハンズオン 4

演習

回答

