

# 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

# 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

## 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

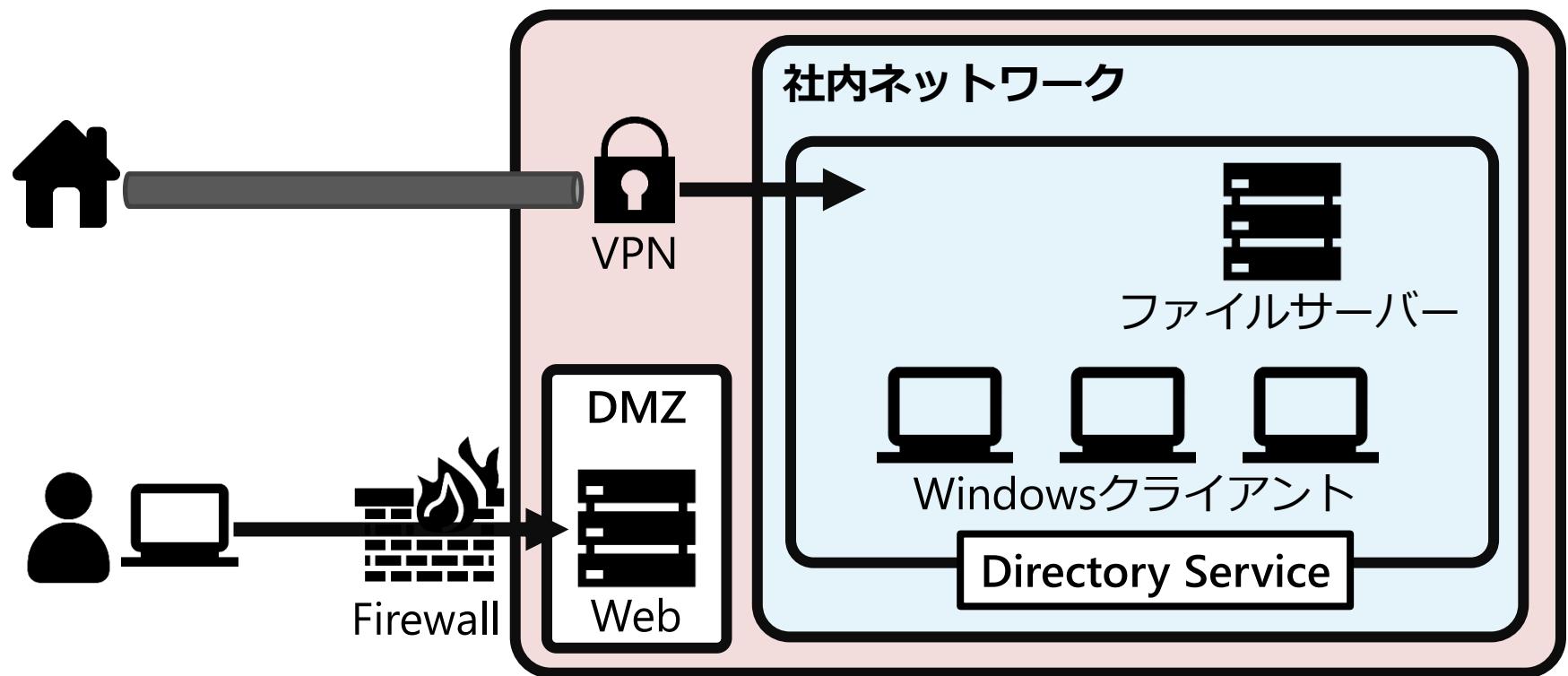
4

## Windowsイベントログの分析

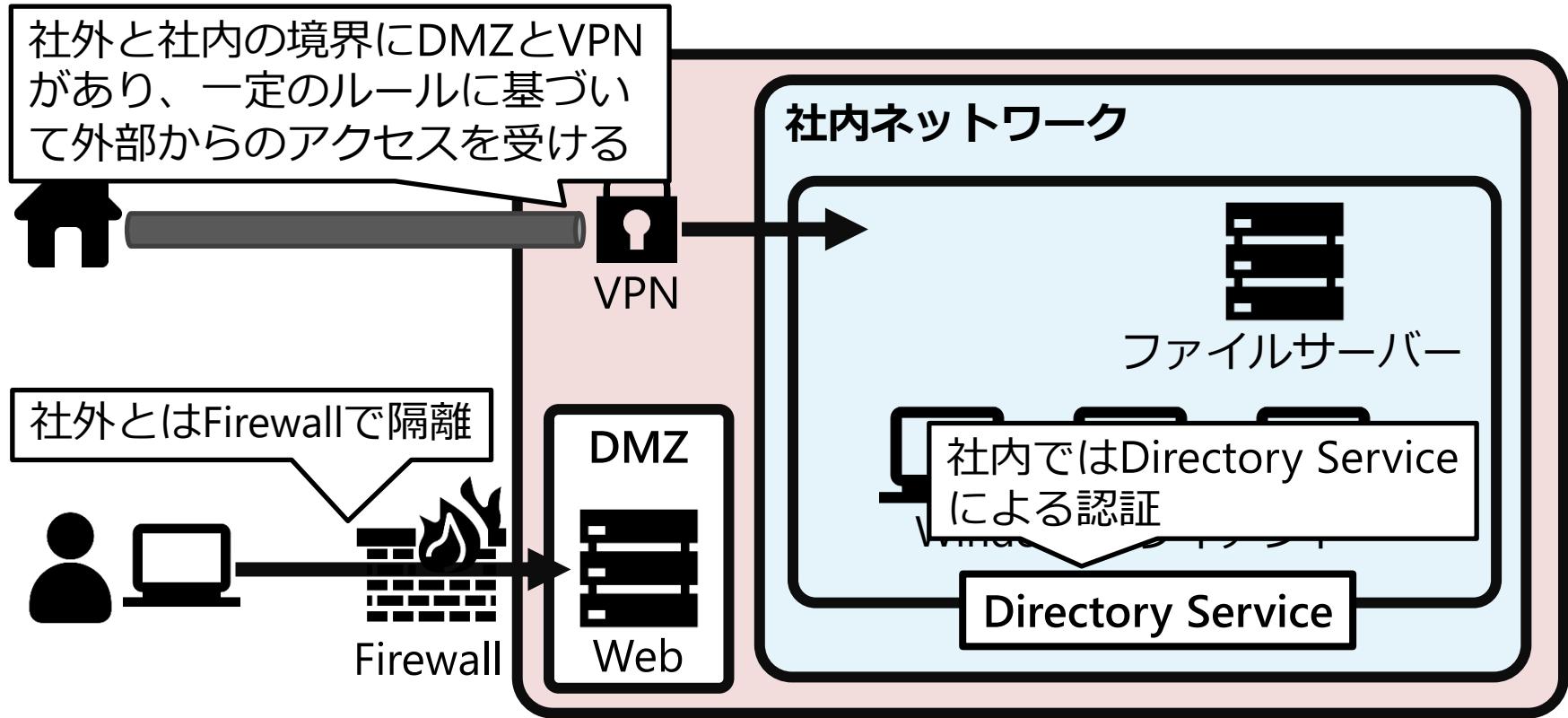
5

## ハンズオン

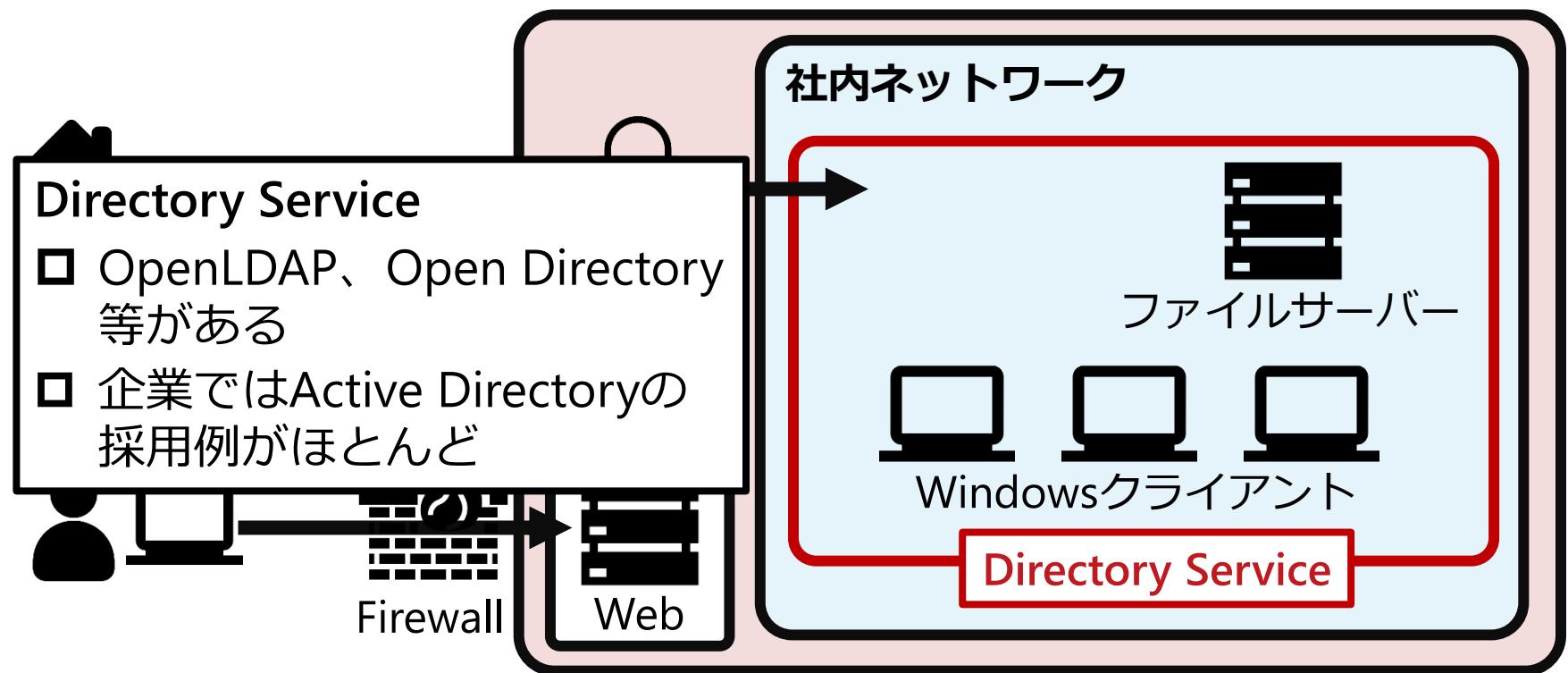
# 一般的な社内ネットワークの構成



# 一般的な社内ネットワークの構成



# 一般的な社内ネットワークの構成



# 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）

# ドメインとフォレスト

## ドメイン

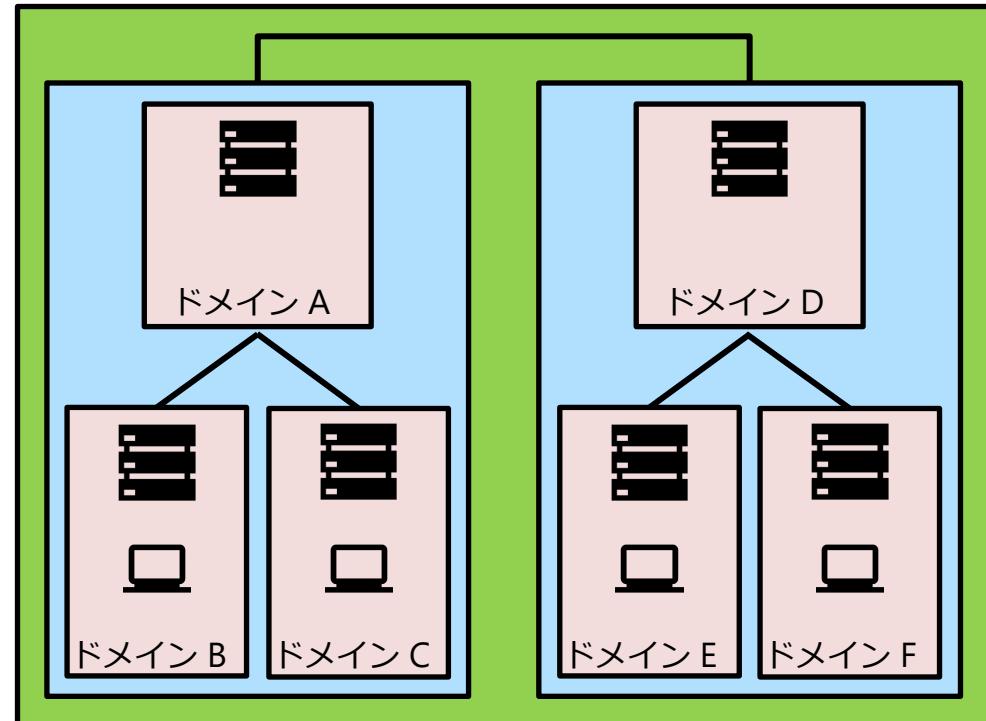
- あるドメインコントローラーが管理する範囲
- 複数台で管理することもある
- 親子：サブドメイン継承
- 信頼：相互に認証し合う

## ツリー

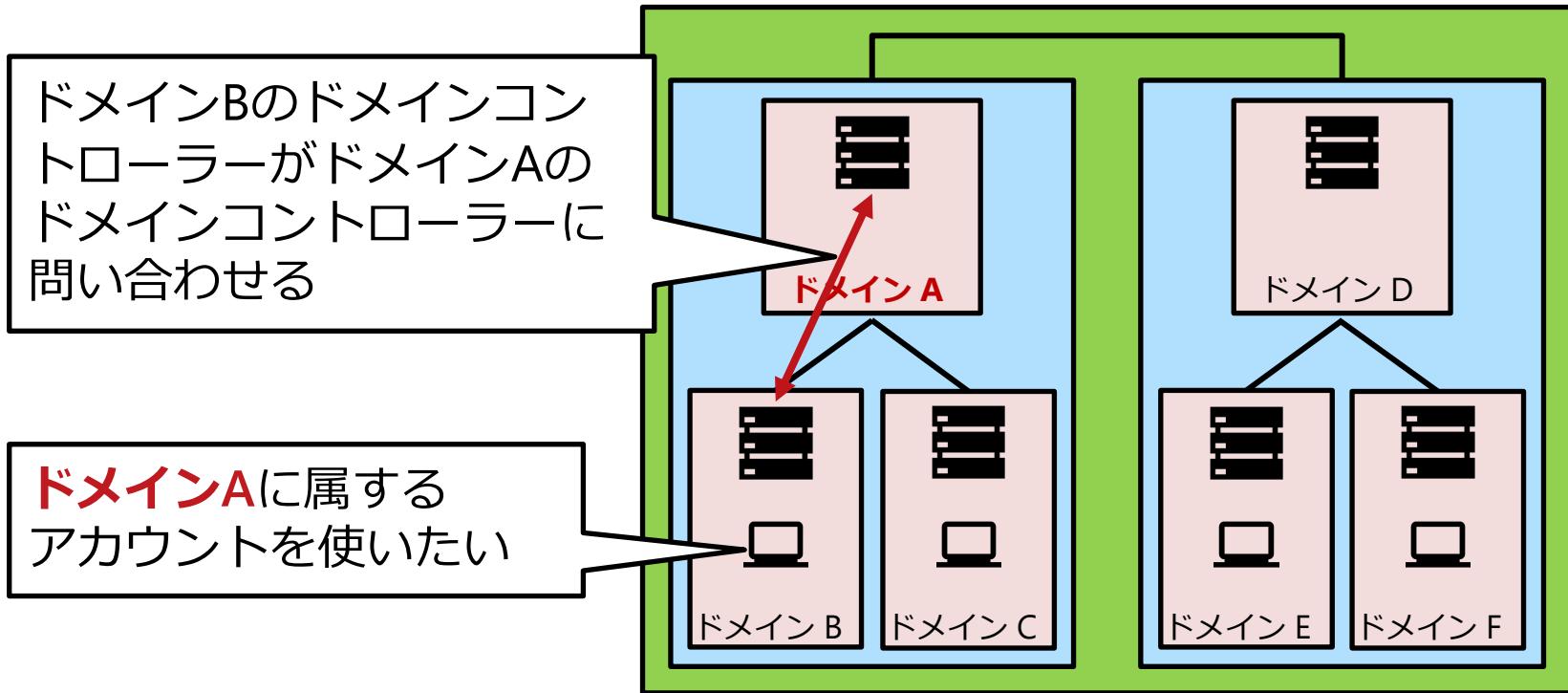
- ドメインの親子関係のみで形成された構造
- 子は親のサブドメインを用いる
- 事業部制・独立部隊がいる・子会社etc...

## 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...



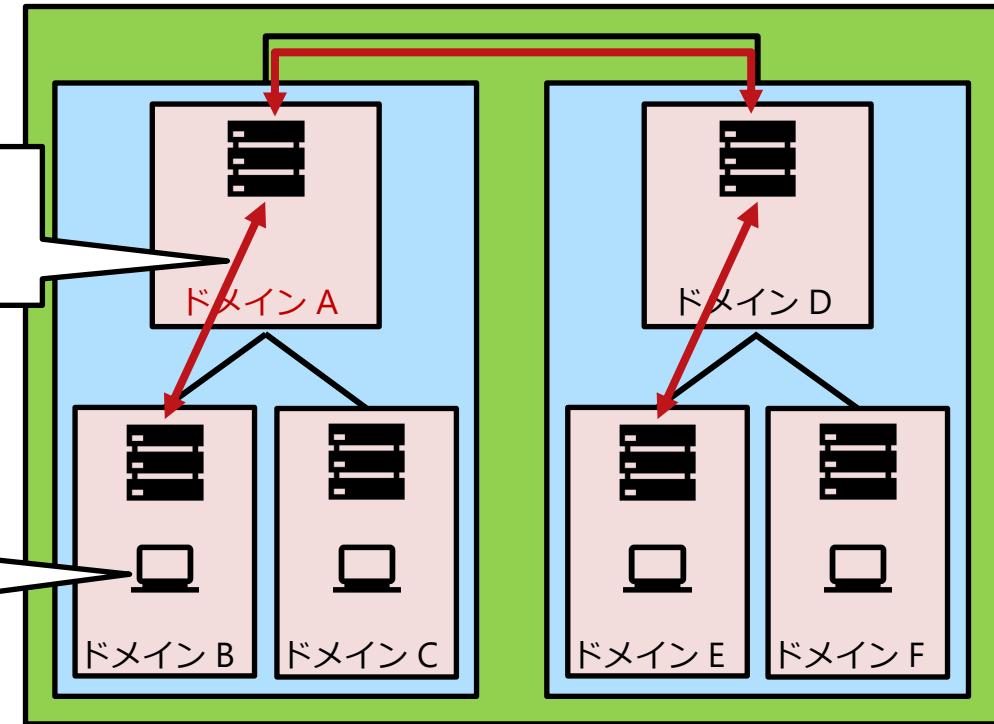
# ドメインとフォレスト



# ドメインとフォレスト

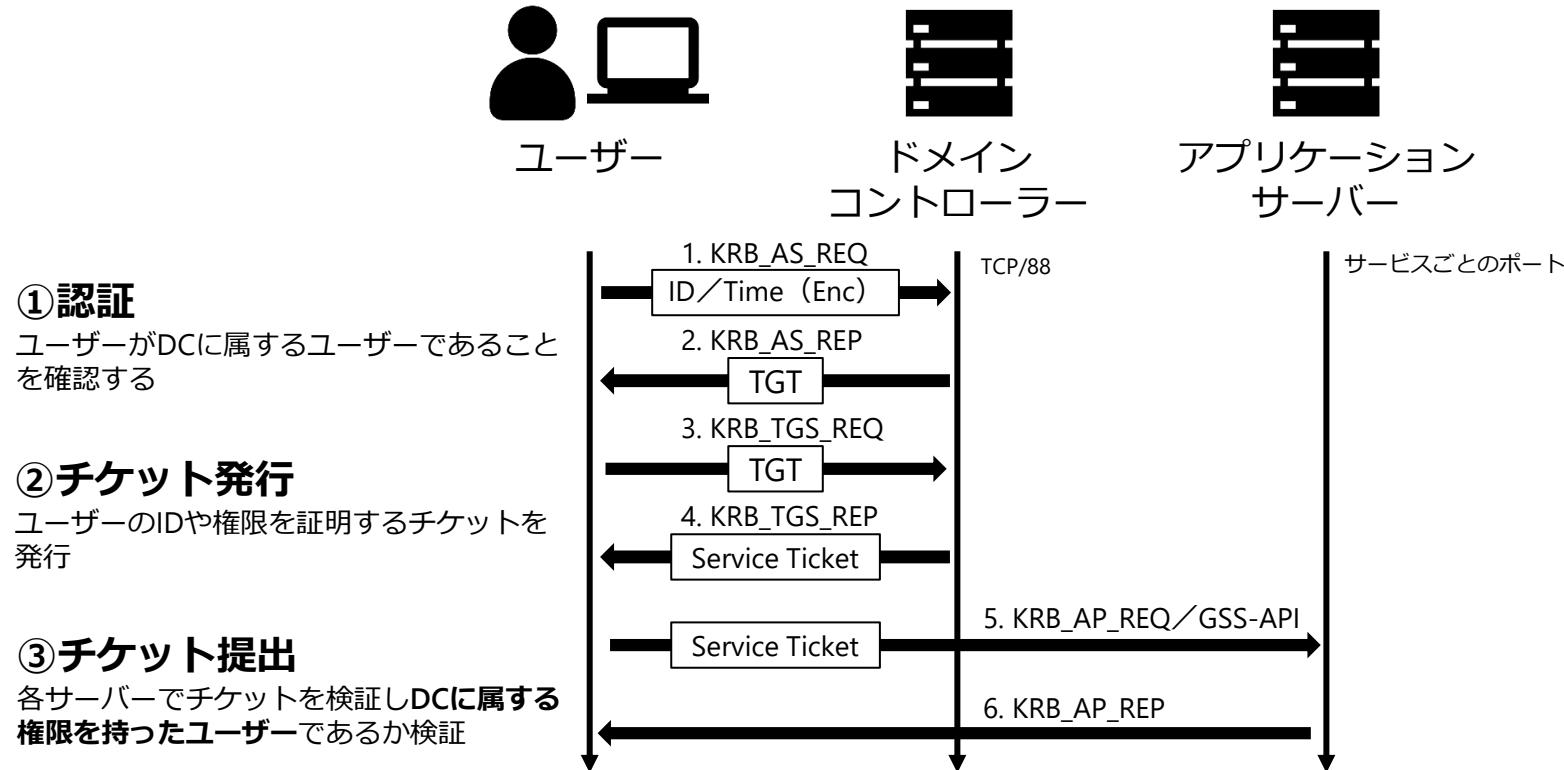
BがAに、 AがDに、 DがEに  
問い合わせさせる

ドメインEに属する  
アカウントを使いたい



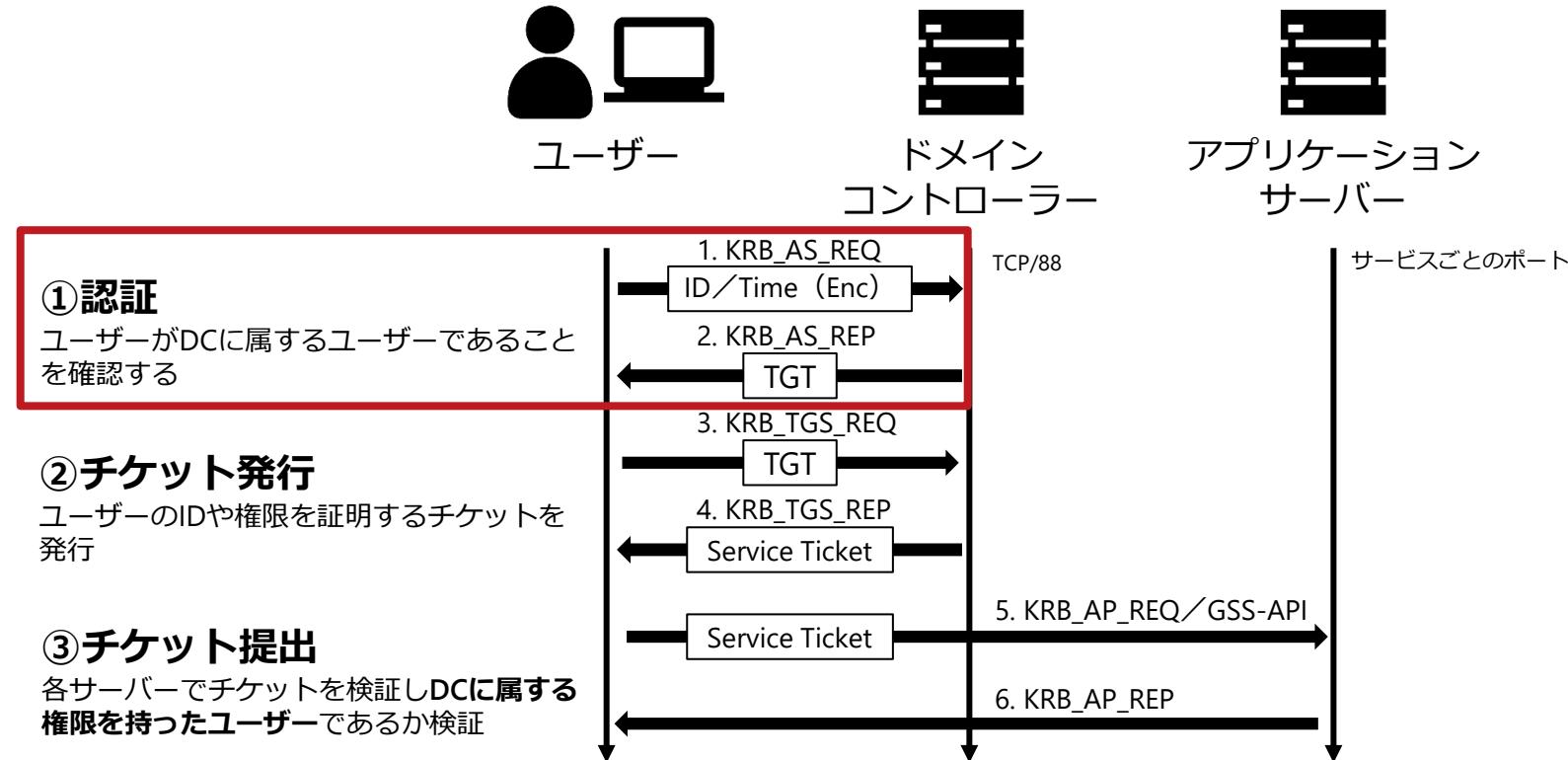
# ADにおける認証／認可の仕組み

## RFC 4120 - The Kerberos Network Authentication Service (V5)



# 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

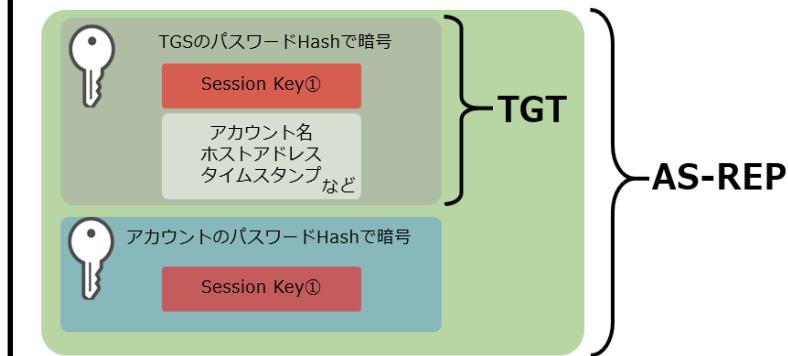
- TGTをもらうための認証情報を含んだリクエスト
  - ✓ アカウント名
  - ✓ 端末のIPアドレス
  - ✓ 時刻を自身のパスワードハッシュ（NTLM）で暗号化した値
  - ✓ nonce（リプレイ攻撃防止のための乱数）

## 2. KRB\_AS REP

- 時刻を復号し一致していればTGTとSession Key①（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）

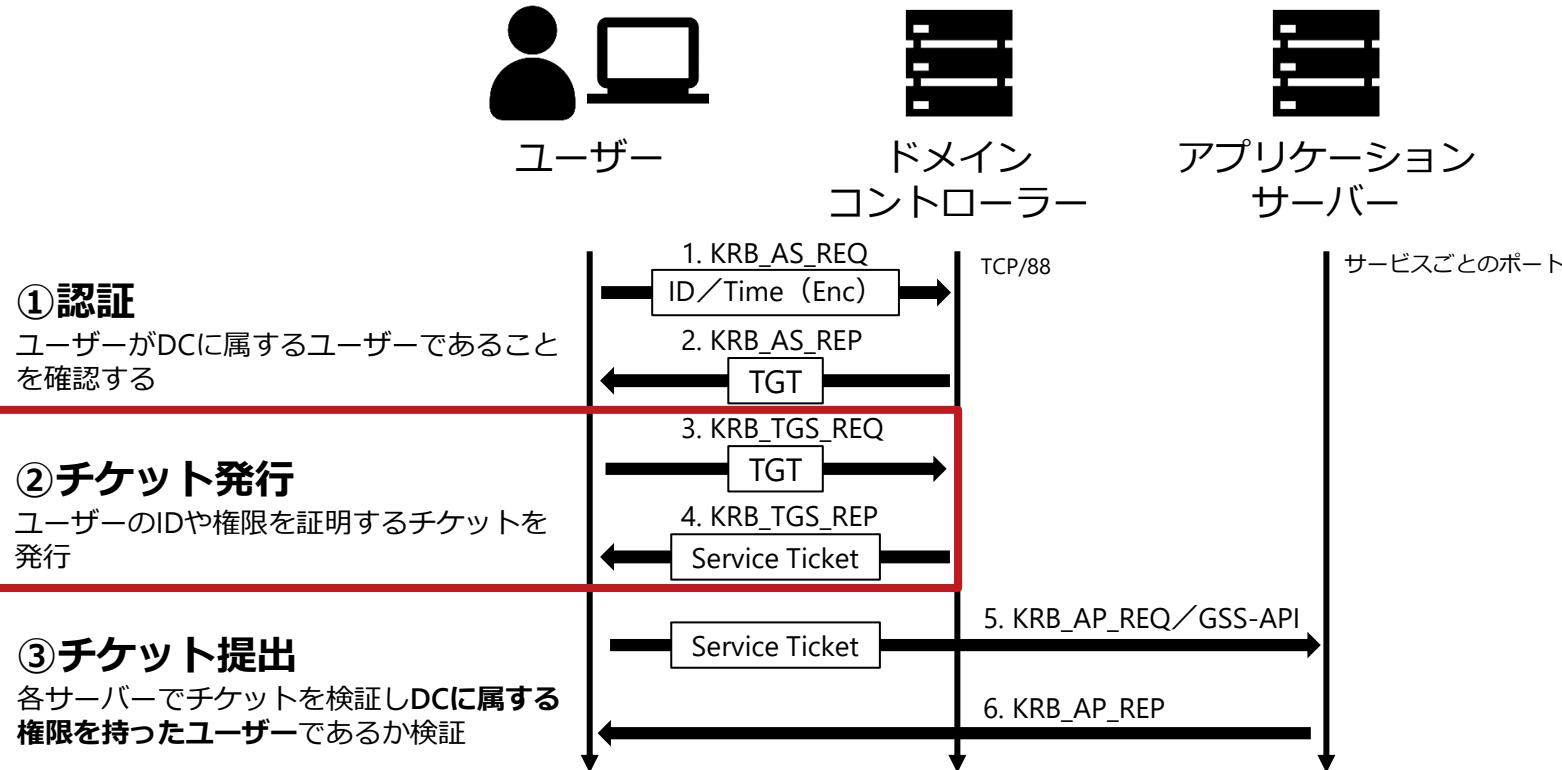


※1 ADはユーザーのPWを知っているので時刻を暗号化／復号できる  
※2 ユーザーはkrbtgtのPWを知らないのでTGTを復号できない

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)



## ②チケット発行：KRB\_TGS\_REQ/REP

### 3. KRB\_TGS\_REQ

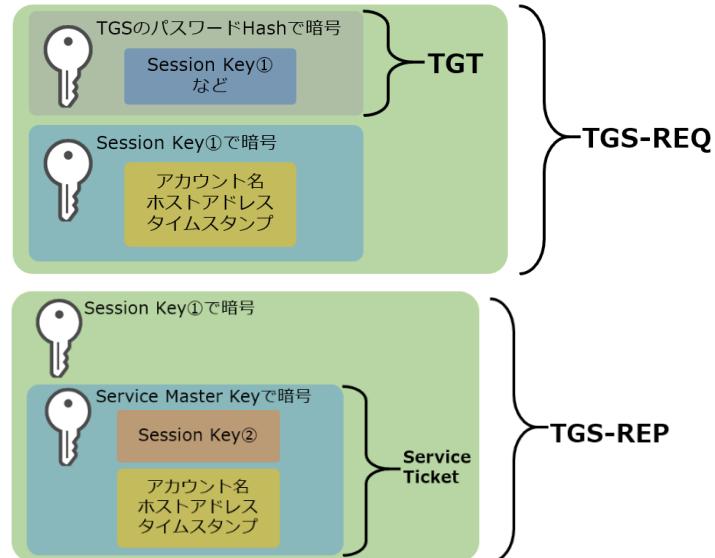
- Session Key①でアカウント名等を暗号化して送信
- サービスプリンシパル名 (SPN) でサービスを指定 (Session Keyで暗号化)
- SPNはアプリケーションサーバ（内で動くサービス）を特定するためのIDで、DC内で一意

### 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

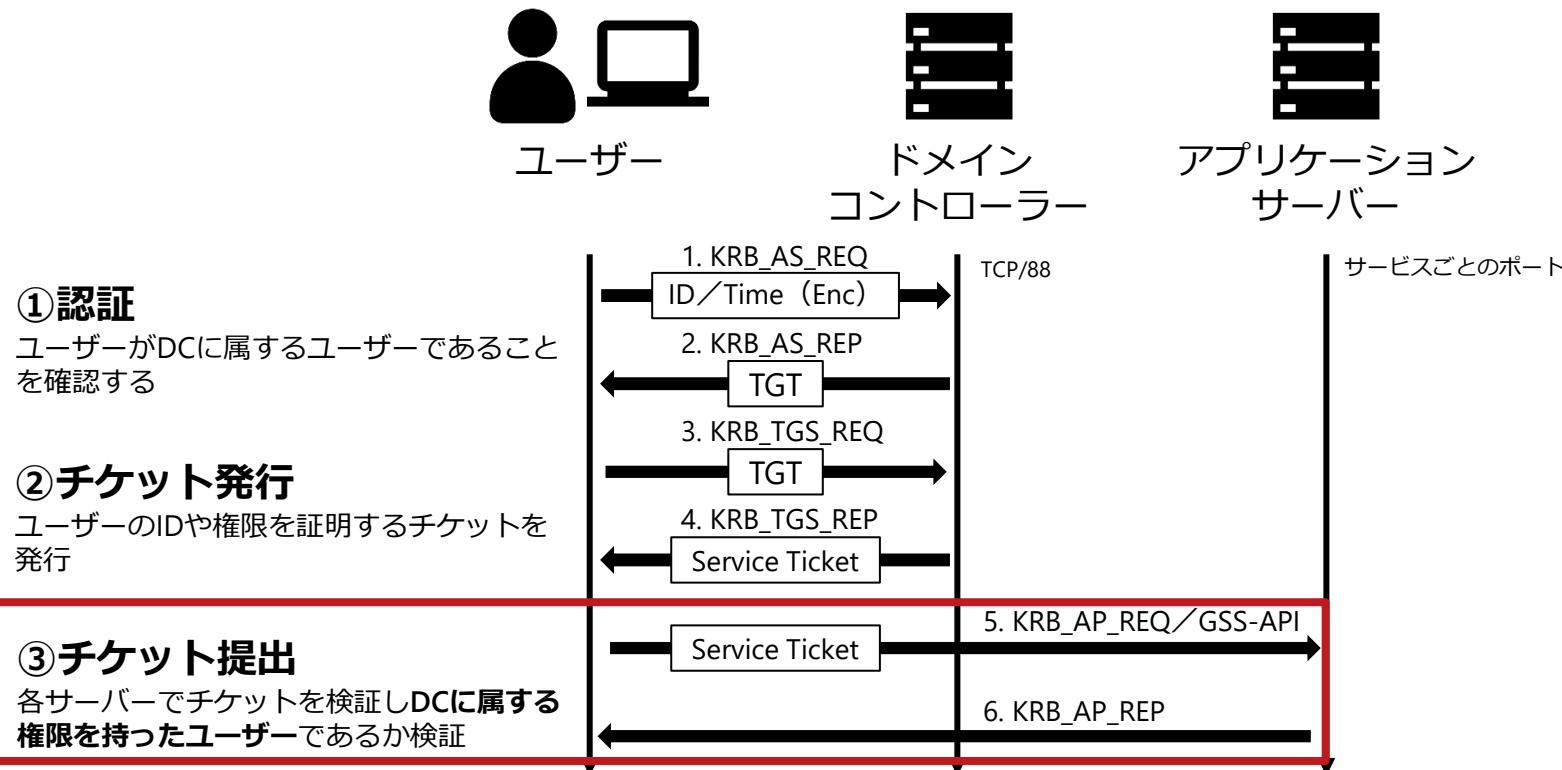
対象サービスで使えるアカウントを認証するためのチケット



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？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.pcappng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>KRB5
- 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

# 確認演習：config設定前

SMB-pcapng

ファイル(F) フォルダ(O) 表示(V) 移動(G) キャプチャ(C) 分析(A) 統計(S) 電話(W) 無線(W) ソール(I) ヘルプ(H)

表示ファイル… <Ctrl+>/ を適用

No.	Time	Source	Destination	Protocol	Length	Info
1	2023-07-31 17:45:10.806935	10.10.100.105	10.10.100.4	SMB	127	Negotiate Protocol Request
2	2023-07-31 17:45:10.808662	10.10.100.4	10.10.100.105	SMB2	306	Negotiate Protocol Response
3	2023-07-31 17:45:10.808764	10.10.100.105	10.10.100.4	SMB2	366	Negotiate Protocol Request
4	2023-07-31 17:45:10.809571	10.10.100.4	10.10.100.105	SMB2	430	Negotiate Protocol Response
5	2023-07-31 17:45:10.810549	10.10.100.105	10.10.100.4	SMB2	3669	Session Setup Request
6	2023-07-31 17:45:10.812127	10.10.100.4	10.10.100.105	SMB2	314	Session Setup Response
7	2023-07-31 17:45:10.812399	10.10.100.105	10.10.100.4	SMB2	194	Tree Connect Request Tree: \\ad-win-C.handonlab.local\IPC\$
8	2023-07-31 17:45:10.812850	10.10.100.4	10.10.100.105	SMB2	138	Tree Connect Response

---

```

> Security mode: 0x02, Signing required
> Capabilities: 0x00000001, DFS
Channel: None (0x00000000)
Previous Session Id: 0x0000000000000000
Blob Offset: 0x00000058
Blob Length: 3523
v Security Blob: 60820dbf05620b601050502a0820db330820dafa030302e06092a864882f71201020206...
  v GSS-API Generic Security Service Application Program Interface
    OID: 1.3.6.1.5.5.2 (SPNEGO - Simple Protected Negotiation)
      v Simple Protected Negotiation
        v negTokenInit
          > mechTypes: 4 items
            mechToken: 60820d7106092a864886f71201020201006e820d6030820d5ca003020105a10302010ea...
          v krb5_blob: 60820d7106092a864886f71201020201006e820d6030820d5ca003020105a10302010ea...
            KRBS OID: 1.2.840.113554.1.2.2 (KRBS - Kerberos 5)
            krb5_tok_id: KRBS_AP_REQ (0x0001)
        v Kerberos
          v ap-req
            pwno: 5
            msg-type: krb-ap-req (14)
            Padding: 0
          > ap-options: 20000000
            ticket
              tkt-vno: 5
              realm: HANDSONLAB.LOCAL
            > sname
              v enc-part
                etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
                kvno: 3
                cipher: 2a37af80e2ad4975bb155ff7elca136a6f17106f628bdb7e4e8e24fecbf4e7005b8ed39...
                authenticator

```

```

0170 03 a2 82 04 e9 04 82 04 e5 2a 37
0180 75 bb 15 5f f7 e1 ca 13 6a 6f 17
0190 e7 e4 e8 e2 4f ec bf 4e 78 05 08
01a0 98 ad ce 75 b4 9a 78 d0 88 48 92
01b0 b8 ad 03 f2 eb b1 ac 98 f3 e0 36
01c0 04 fe a5 a9 94 56 21 87 ff e0 ab
01d0 09 1b 87 73 cd 7b 1c 88 bb 6e e1
01e0 c3 72 a7 84 9f 22 33 1a 54 87 2a
01f0 66 97 f9 80 1e 19 1e f8 ea 2a 50
0200 3e 07 71 88 9b 70 ef f5 d2 61 00
0210 f6 ce 7b bc b9 6f 60 47 d3 b1 a7
0220 2f df da 16 e8 8e 01 59 d8 27 31
0230 7a 5e 74 00 9e fa 64 cb 39 4e 4b
0240 03 a9 ea a8 11 4f 33 72 3e 5f d8
0250 bc 45 29 40 80 17 e5 ed 8e e6 a3
0260 5d 3f c2 c0 c3 9d 37 b5 29 6c 62
0270 c1 57 be 70 9e 45 e4 52 28 81 2f
0280 85 d5 d3 81 46 7c ec 18 8d cc f7
0290 e8 e4 c7 29 f0 41 b8 1e 98 7d 58
02a0 d0 18 69 06 46 1f ff f8 6d 54 58
02b0 fe bc de b1 7b 9d ad dd 37 67 ed
02c0 75 4c 79 aa 50 48 a0 3f b2 05 a0
02d0 3b 74 a9 7b 19 e1 54 c3 e6 39 09
02e0 2b d9 da 1e 03 74 d2 22 a5 50 e3
02f0 11 43 14 d3 35 17 18 46 21 ec 0c
0300 ba fa b5 a1 86 0a 71 04 40 9f 5c
0310 a8 34 c8 49 61 fc 50 3e 2a fe 9f
0320 92 c8 29 49 4c e9 1e b1 52 e9 84
0330 6f 39 92 10 53 30 0d 4b a1 ac c5
0340 d4 94 ca 4a 77 e0 43 cc b8 7c d3
0350 aa 87 05 e5 5c ea 0b 87 b1 e3 9c
0360 74 82 2a 94 e7 df 59 cb 88 90 35
0370 fe 88 ea e7 c1 55 35 58 a4 27 bf
0380 35 1c d9 20 3b 44 68 99 4b fc dd
0390 d7 4e 67 c1 57 3c ad 1c 5e ff 72

```

## 確認演習：config設定後

演習

The screenshot shows a Wireshark capture of SMB traffic and a detailed analysis of a Kerberos ticket exchange.

**Wireshark SMB Traffic:**

No.	Time	Source	Destination	Protocol	Length	Info
1	2023-07-31 17:45:18.806935	10.10.100.185	10.10.100.4	SMB	127	Negotiate Protocol Request
2	2023-07-31 17:45:18.808662	10.10.100.4	10.10.100.105	SMB2	306	Negotiate Protocol Response
3	2023-07-31 17:45:18.808764	10.10.100.105	10.10.100.4	SMB2	366	Negotiate Protocol Request
4	2023-07-31 17:45:18.809571	10.10.100.4	10.10.100.105	SMB2	430	Negotiate Protocol Response
5	2023-07-31 17:45:18.810548	10.10.100.105	10.10.100.4	SMB2	3669	Session Setup Request
6	2023-07-31 17:45:18.812127	10.10.100.4	10.10.100.105	SMB2	314	Session Setup Response
7	2023-07-31 17:45:18.812399	10.10.100.105	10.10.100.4	SMB2	194	Tree Connect Request Tree: \\ad-win-C.hansonlab.local\IPC\$
8	2023-07-31 17:45:18.812850	10.10.100.4	10.10.100.105	SMB2	138	Tree Connect Response

**Kerberos Analysis:**

- Security Blob:** 60820dbf06062b0601050502a0820db330820dafa030302e06092a864882f71201020206...
- GSS-API Generic Security Service Application Program Interface**
  - OID:** 1.3.6.1.5.5.2 (SPNEGO - Simple Protected Negotiation)
  - Simple Protected Negotiation**
    - negTokenInit**
      - mechTypes:** 4 items
      - mechToken:** 60820d7106092a864886f71201020201006e820d6030820d5ca003020105a10302010ea2..
    - krb5\_blob:** 60820d7106092a864886f71201020201006e820d6030820d5ca003020105a10302010ea2...
      - KRB5 OID:** 1.2.840.113554.1.2.2 (KRB5 - Kerberos 5)
      - krb5\_tok\_id:** KRB5\_AP\_REQ (0x0001)
  - Kerberos**
    - ap-req**
      - pvno:** 5
      - msg-type:** krb-ap-req (14)
      - Padding:** 0
- ticket**
  - tkt-vno:** 5
  - realm:** HANSONLAB.LOCAL
  - sname**
  - enc-part**
    - etype:** eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
    - kvno:** 3
    - cipher:** 2a37af80e2ad4975bb155ff7elcal36a6f17106f628bde7e4e824fecbf4e7005b8ed39...
      - Decrypted keytab 18 usage 2 using keytab principal ad-win-C@hansonlab.local (id=keytab.4 same**
      - encTicketPart**
        - Padding:** 0
        - flags:** 40a50000
        - key**

## 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

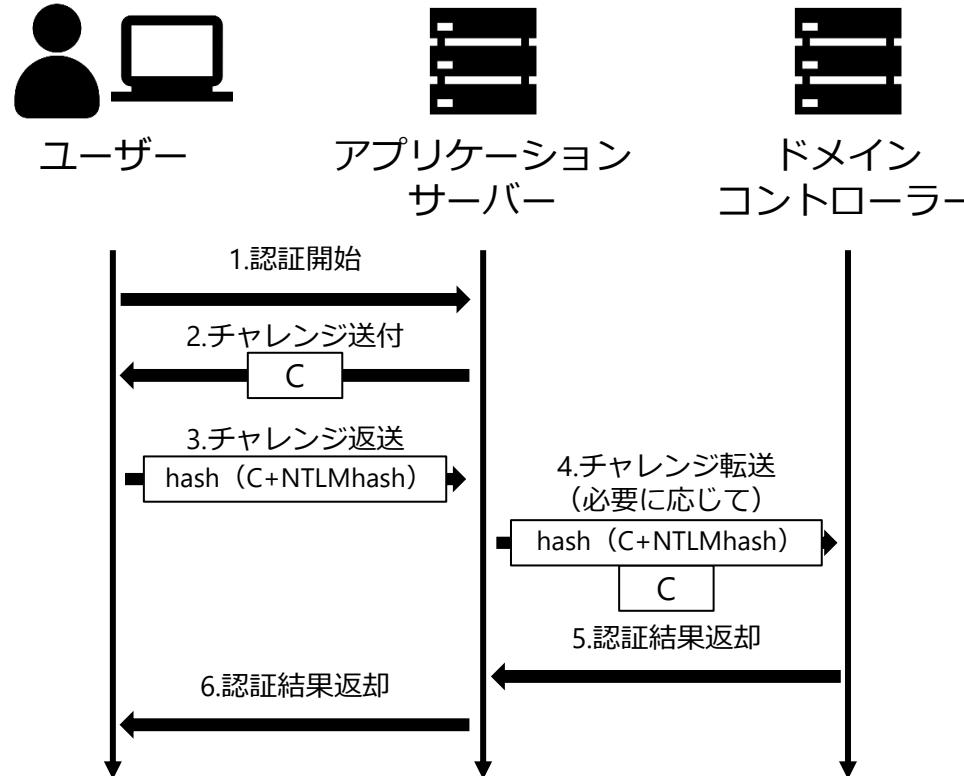
## 回答

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

# その他の認証方式

## NTLM認証



# NTLM認証とは

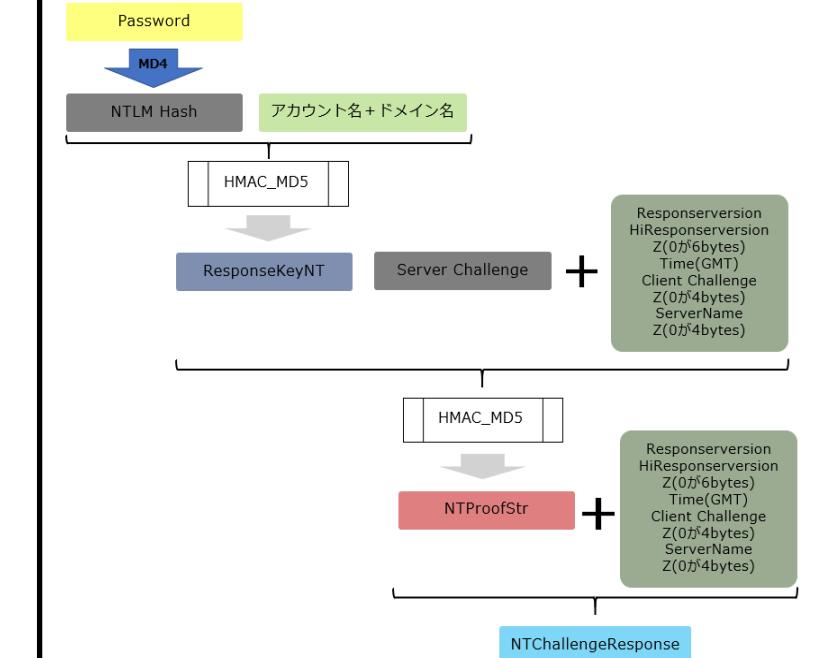
## チャレンジ-レスポンス方式

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

# 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

# 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

# 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

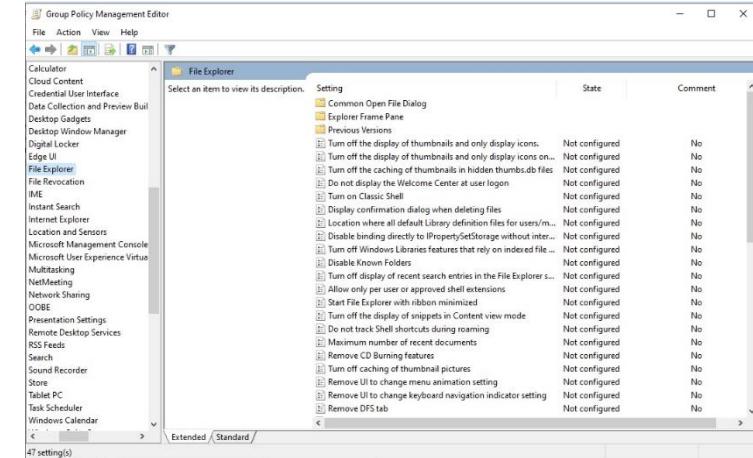
## □ ルールを定めたGPOをDCの各グループにリンク

- ✓ 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

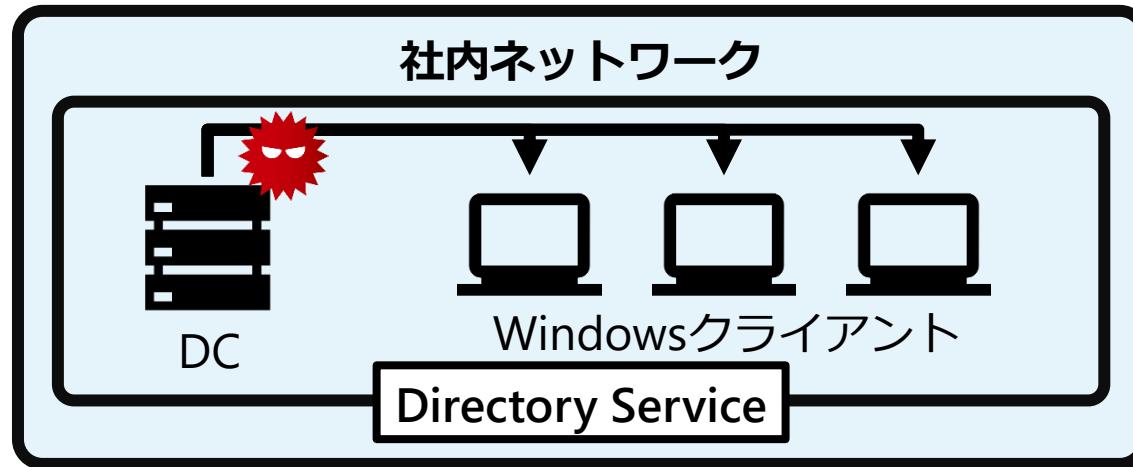
## □ 通常は専用エディターを使って操作



# 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ー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



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

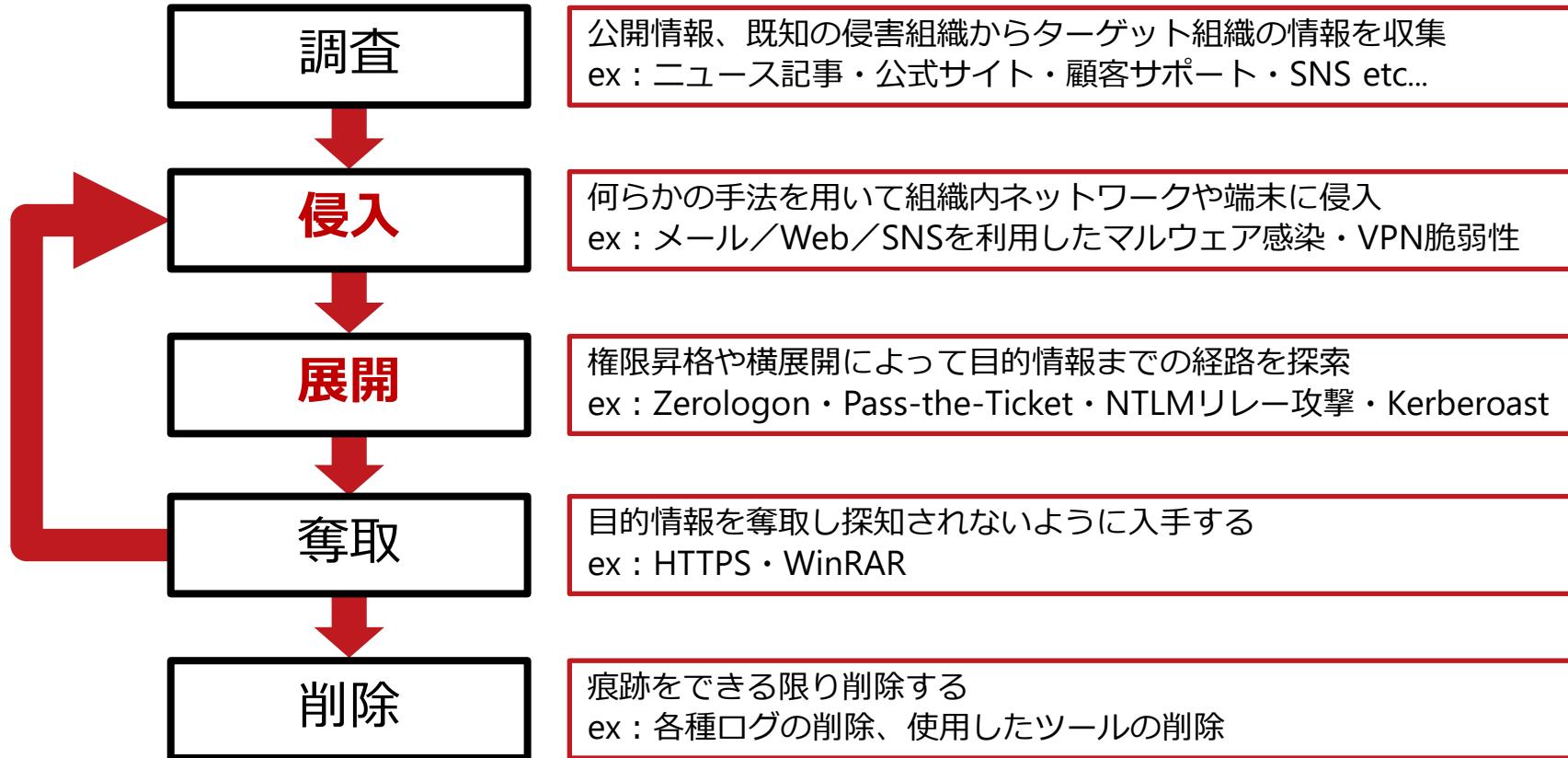
4

Windowsイベントログの分析

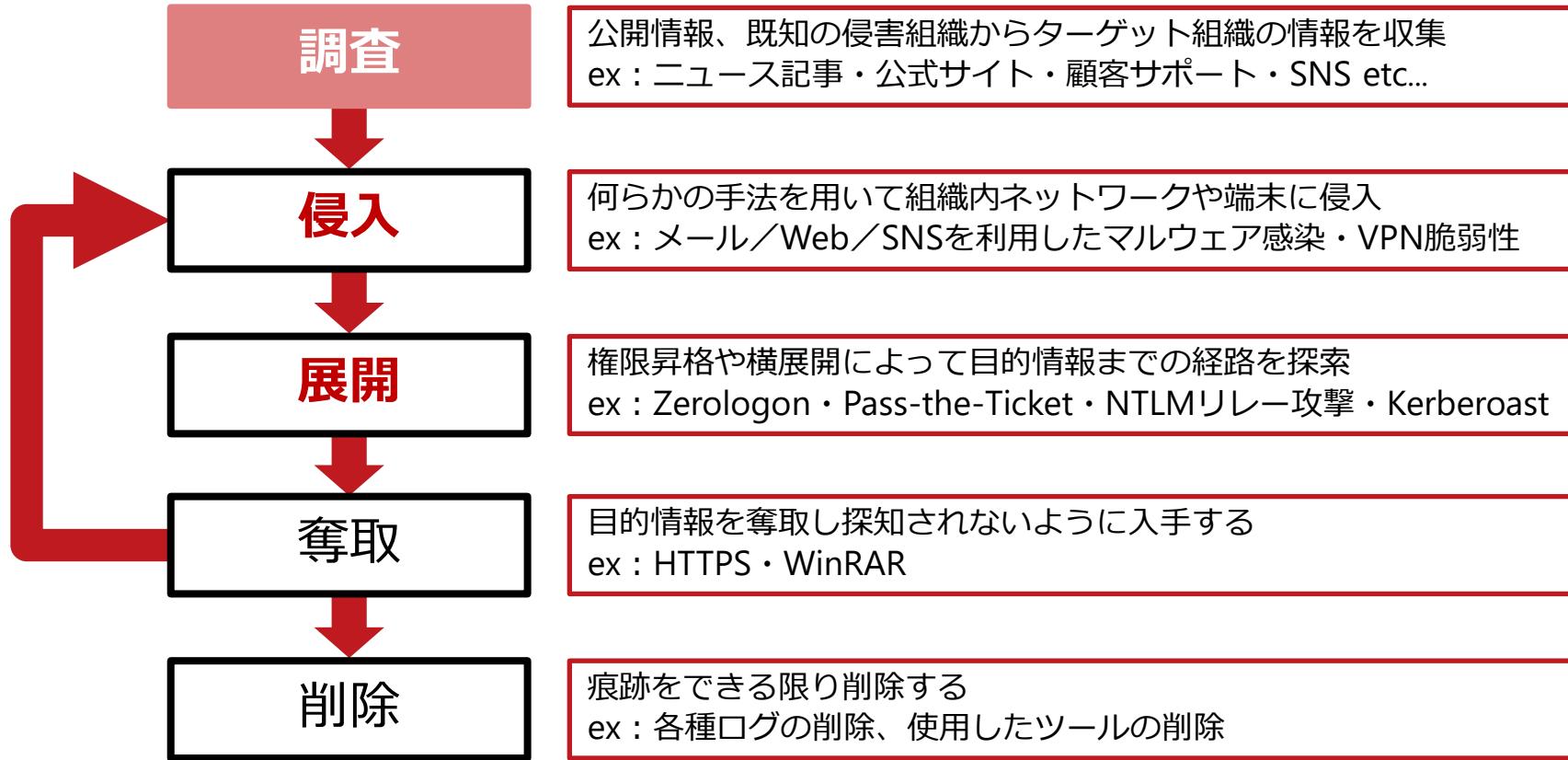
5

ハンズオン

# 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

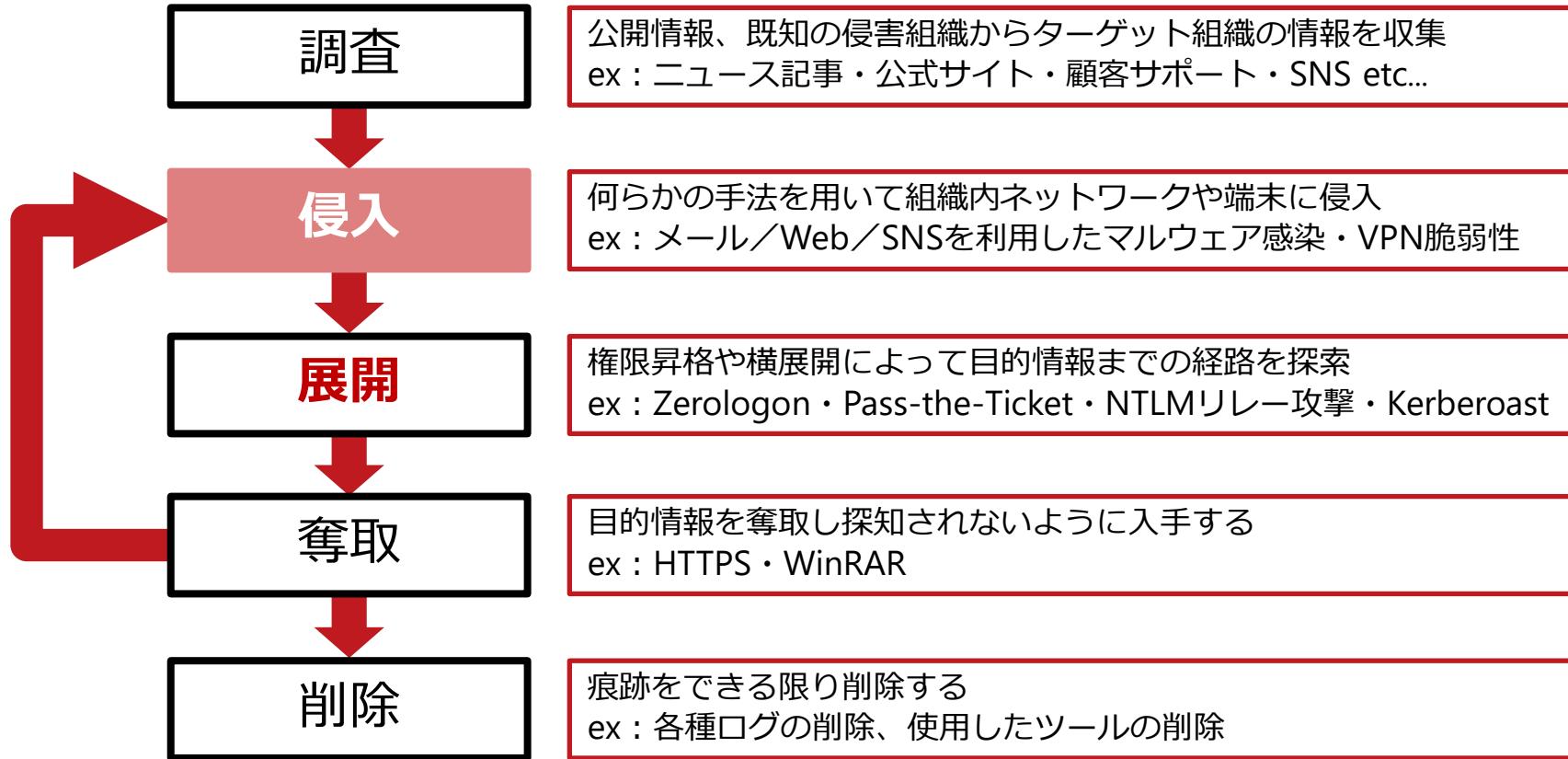
## EASMとは

- 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

# 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

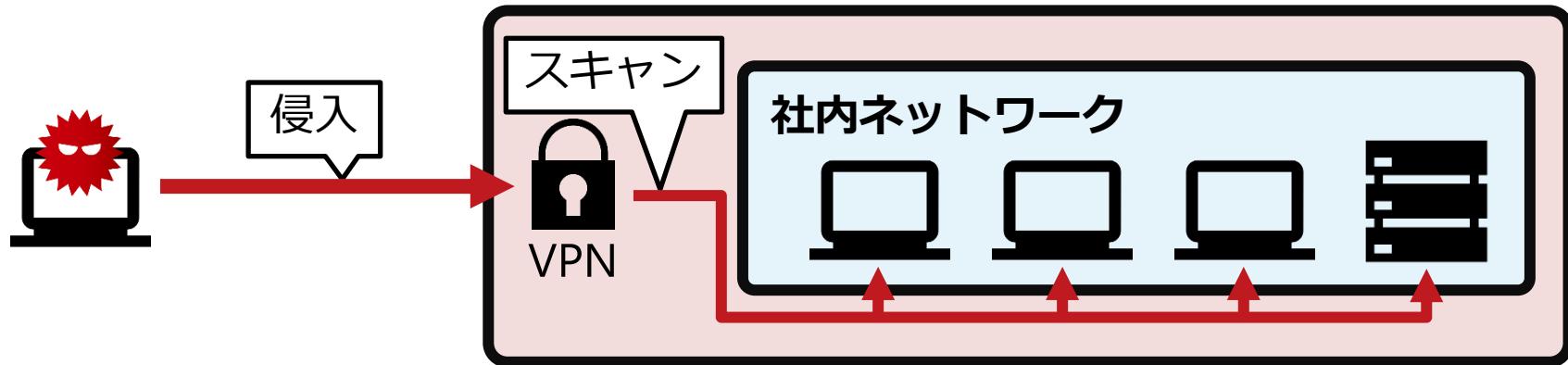


100%侵入を防ぐことは難しいが、  
対策を取ることでリスクの低減につながる

# 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



# 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

# 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



攻撃者が乗っ取ったアカウント

Dear [REDACTED].  
Nice to meet you.  
May I ask a question please?



ターゲットアカウント

Hi [REDACTED].  
Yes, of course.



攻撃者が乗っ取ったアカウント

Do you satisfy with your salary?



ターゲットアカウント

Why is that important to you?



攻撃者が乗っ取ったアカウント

If you want I can introduce you new good job.  
Part-time job is also possible.  
What do you think?



ターゲットアカウント

I am looking for a part-time job, work remotely.  
Thank you.



攻撃者が乗っ取ったアカウント

I will send you our JD.  
Is it okay?



ターゲットアカウント

ok



攻撃者が乗っ取ったアカウント

Here I attach you.



攻撃者が乗っ取ったアカウント

Attach file

Please review.

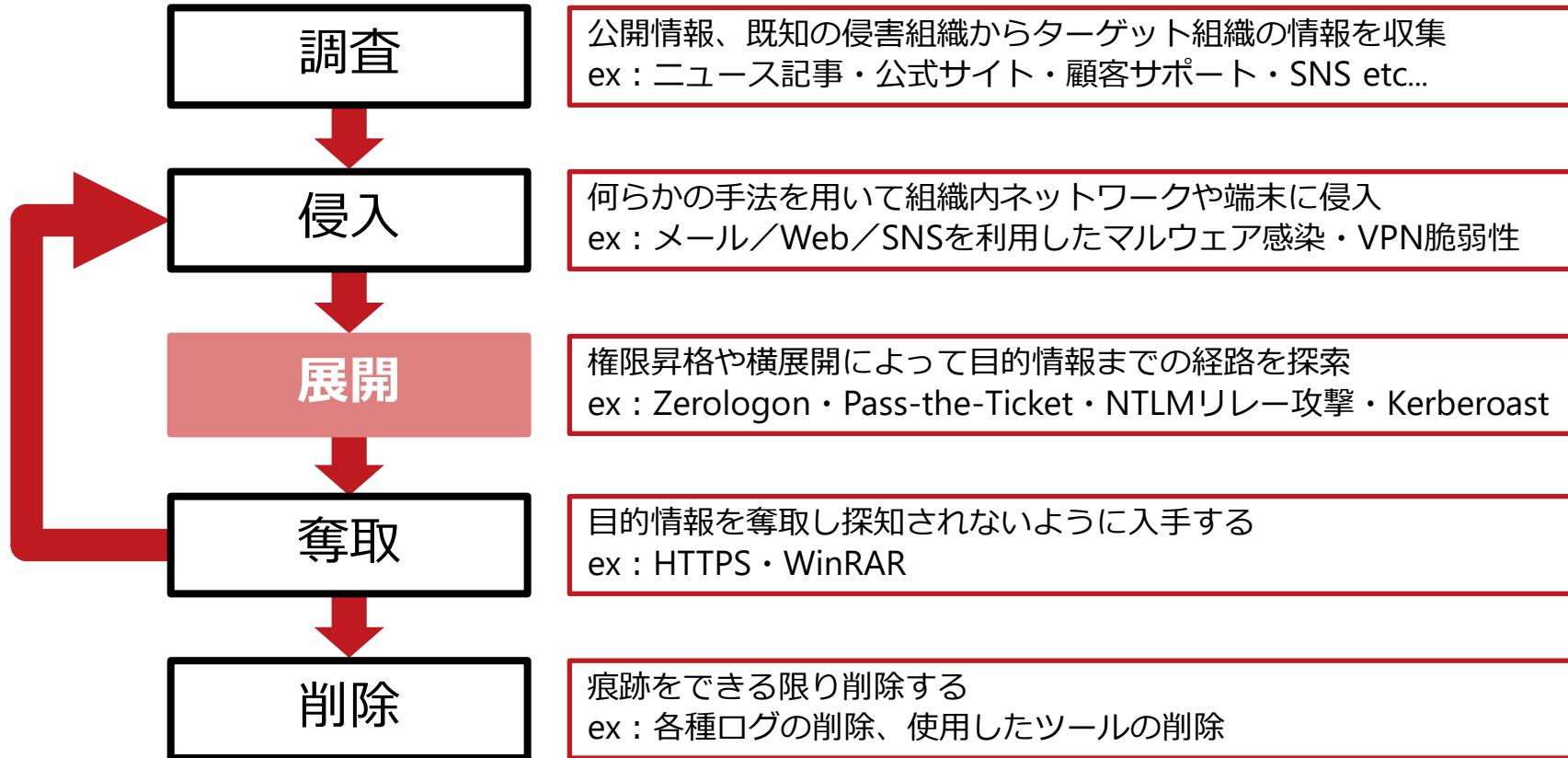


攻撃者が乗っ取ったアカウント

Password is [REDACTED]  
Please check and let me know your opinion.

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)

# 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）を持つドメインユーザーを使用しない

# 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

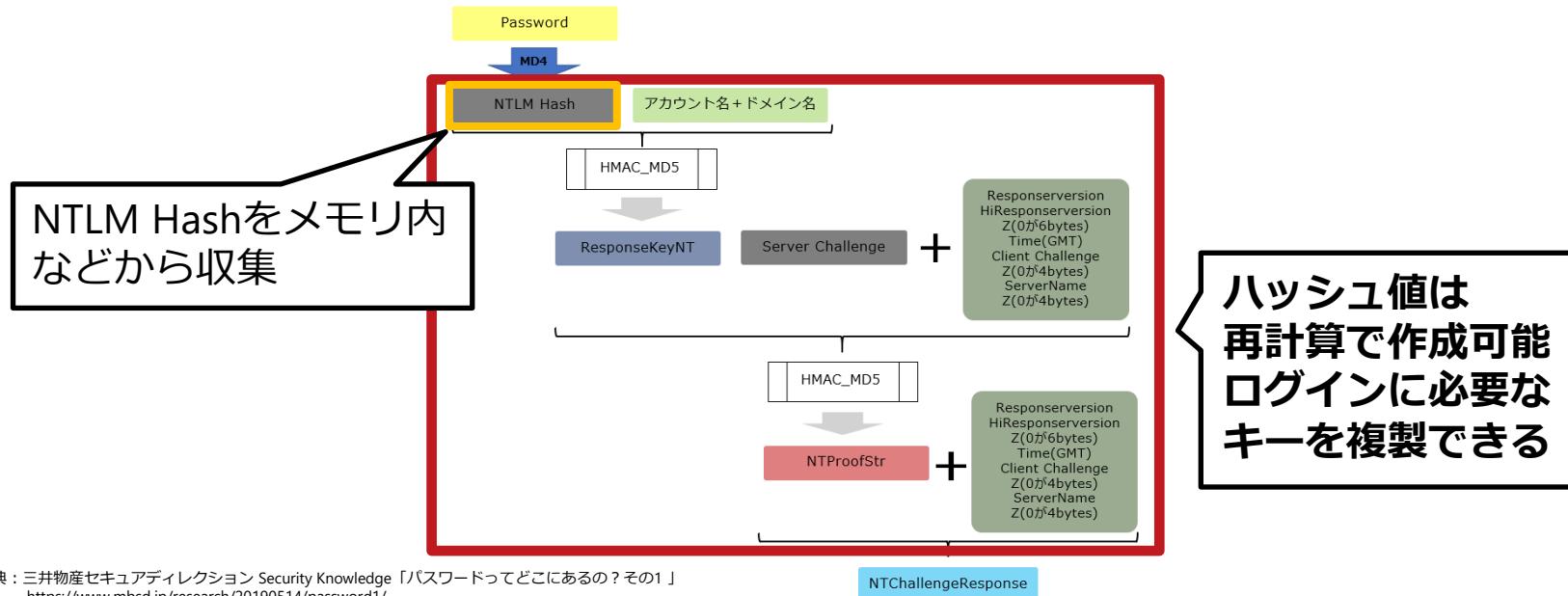
## Linux環境における攻撃手法

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

# 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、それを使っ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



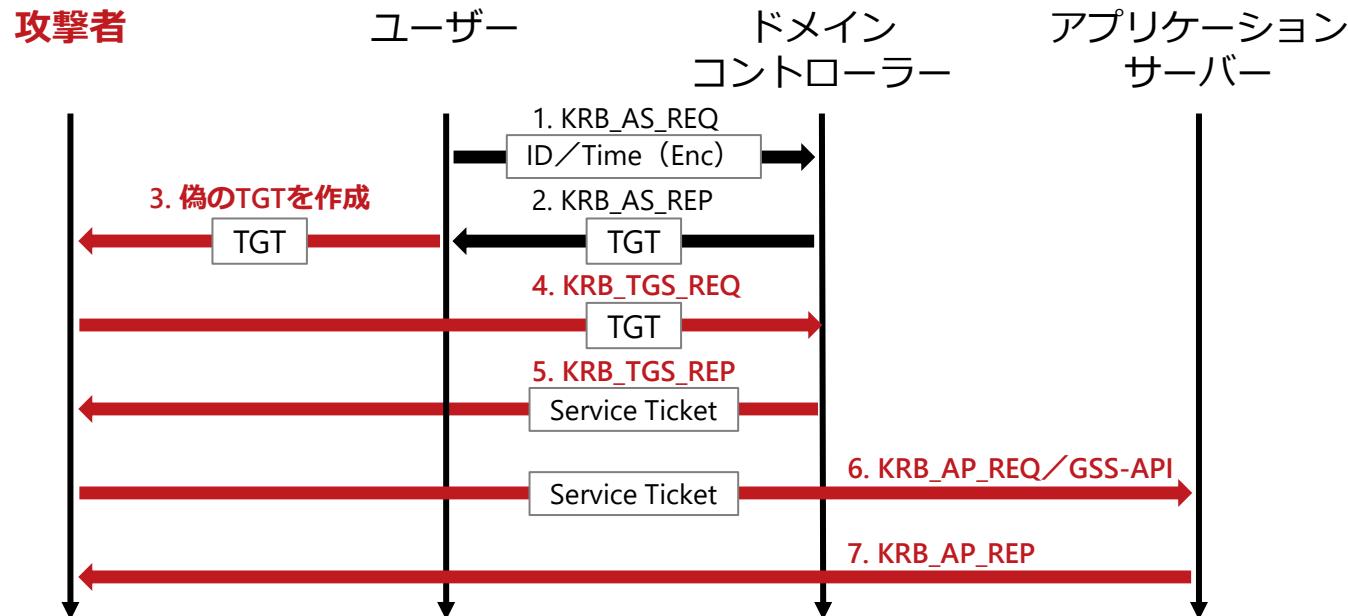
出典：三井物産セキュアディレクション Security Knowledge 「パスワードってどこにあるの？その1」  
<https://www.mbsd.jp/research/20190514/password1/>

NTChallengeResponse

# 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



# NTDSダンプ

## 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¥
```

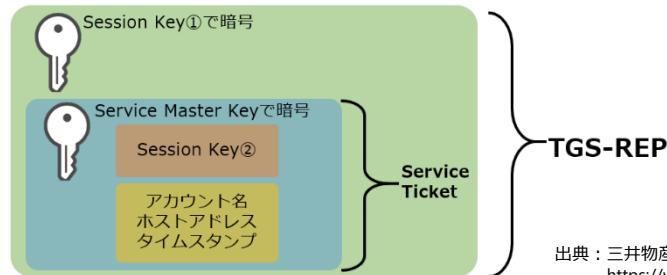
```
# NTDS.ditからパスワードハッシュを抽出
```

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

# 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



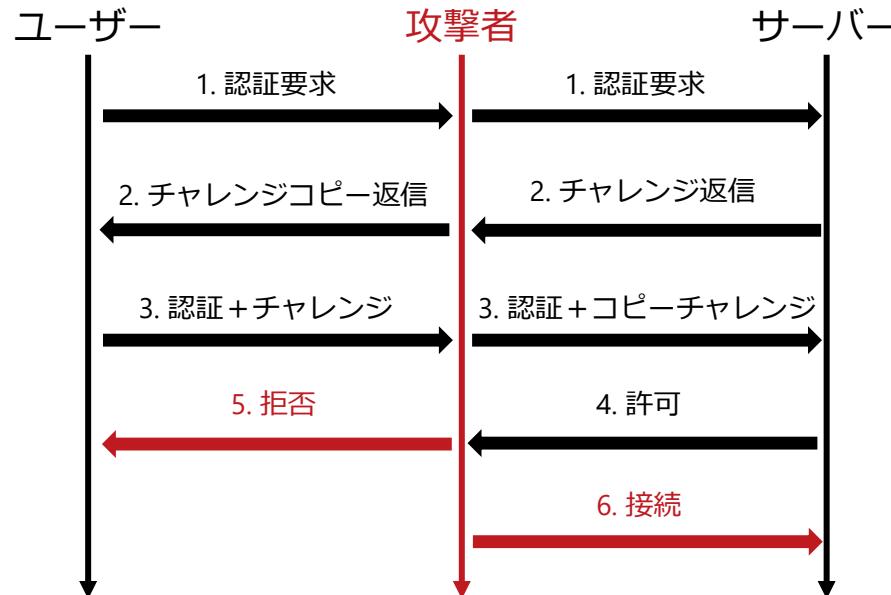
出典：三井物産セキュアディレクション Security Knowledge 「パスワードってどこにあるの？その2」  
<https://www.mbsd.jp/research/20190520/password2/>

→ 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



# 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)

# 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)

# 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

# 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

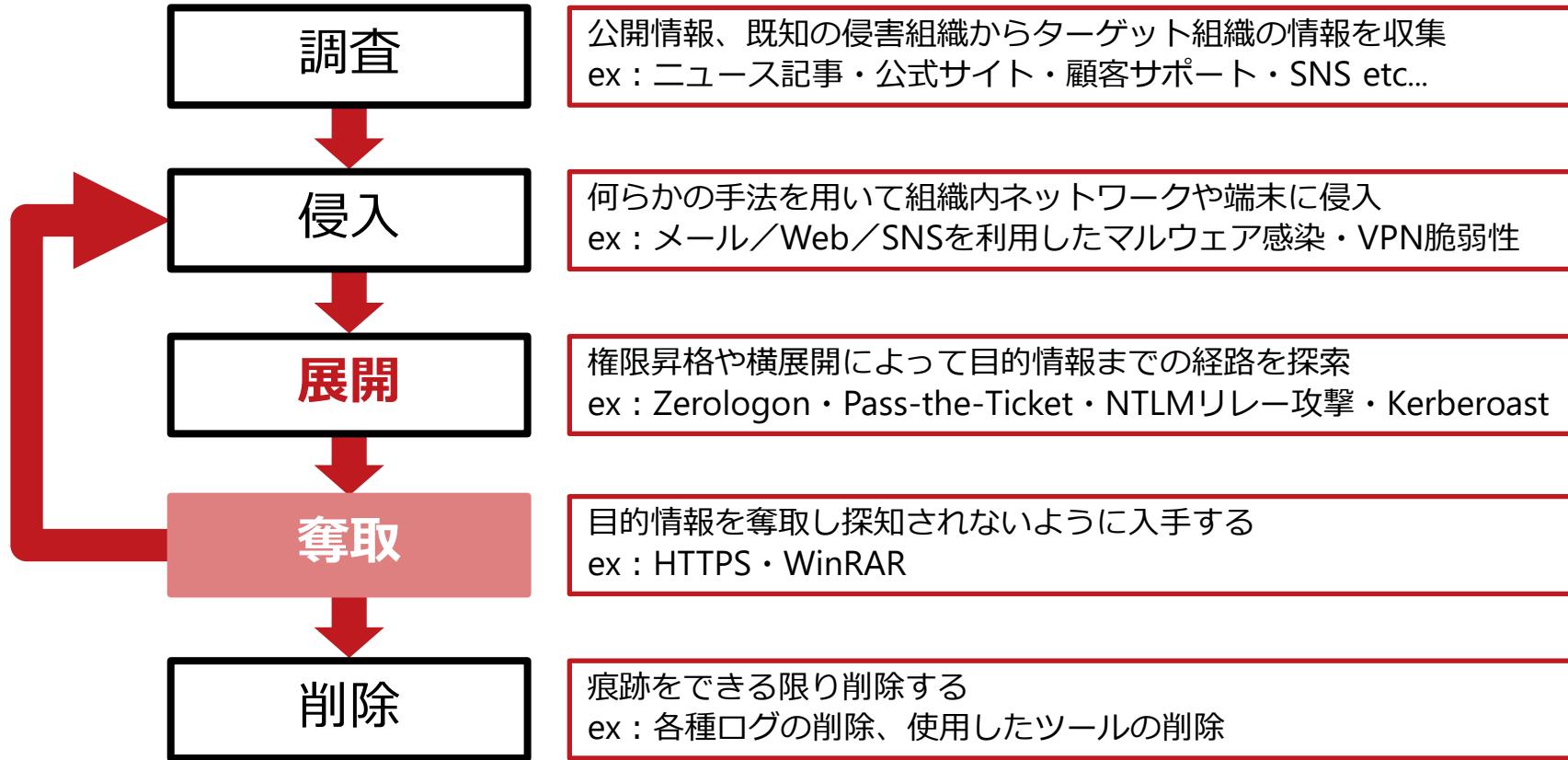
## SAM

- 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

# 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

- 攻撃者の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

## 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

# 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

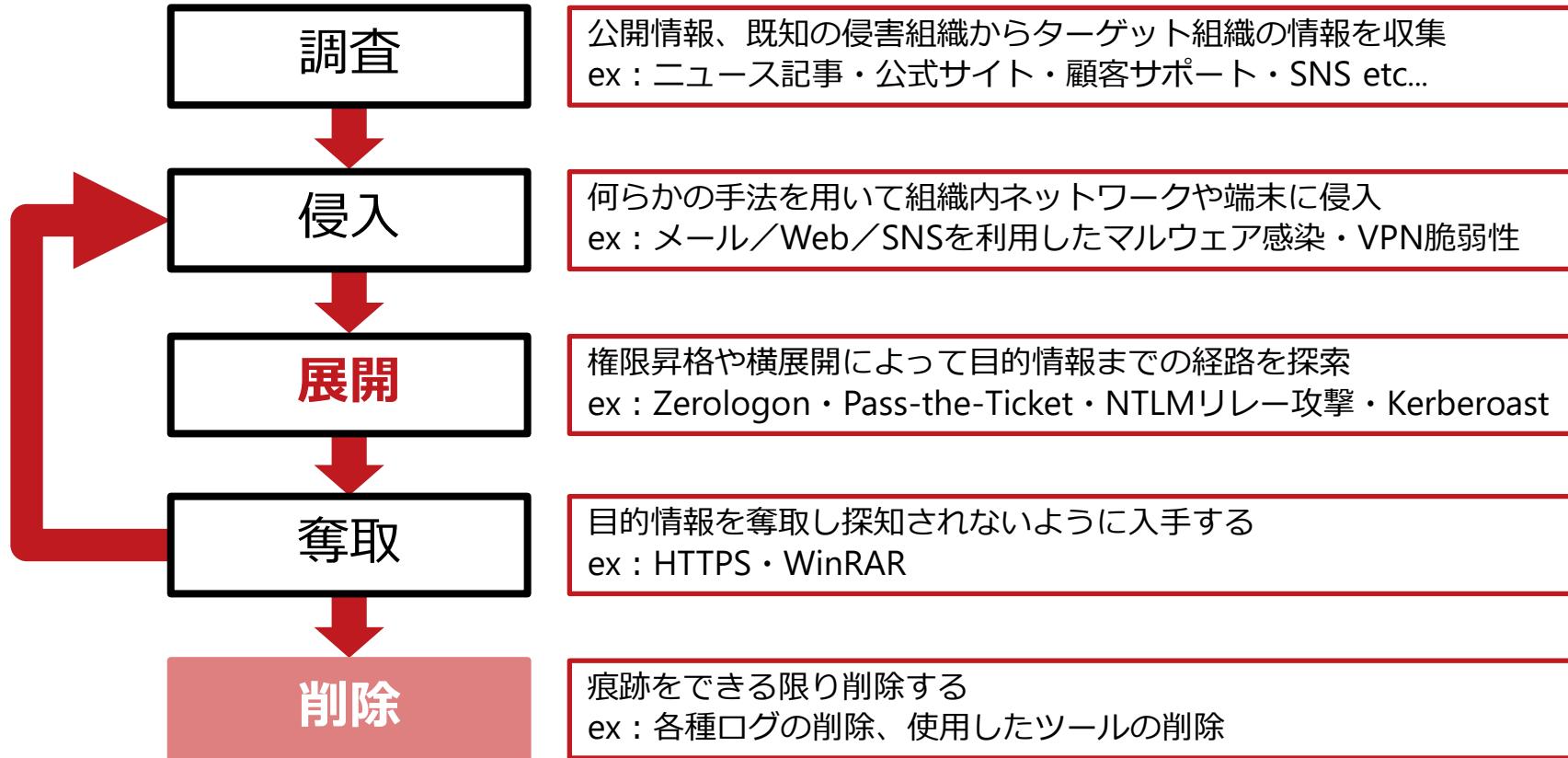
## SSH、HTTPS、SOCKS5のトンネリング通信

- 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

# 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

- 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

# 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

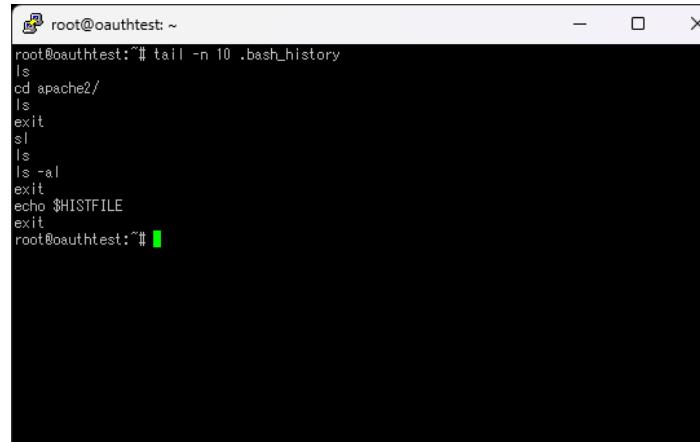
- 日時や出力は記録されない

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

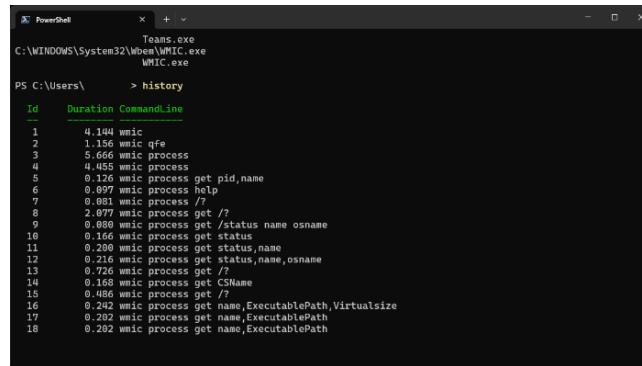
# 参考：コマンド実行のログ（PowerShell）

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

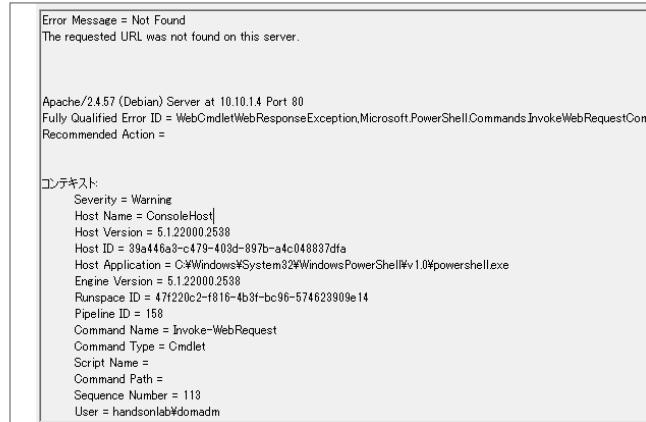
Windowsイベントログにも残る

- 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



A screenshot of a Windows PowerShell window titled "PowerShell". The title bar shows the path "C:\WINDOWS\System32\Wbem\WMIC.exe". The main area displays the command history with columns for Id, Duration, and CommandLine. The log shows 18 entries related to the WMIC command, such as "wmic process get pid,name" and "wmic process get /?".

Id	Duration	CommandLine
1	4.144	wmic
2	1.156	wmic qfe
3	5.666	wmic process
4	4.455	wmic process
5	0.000	wmic process get pid,name
6	0.097	wmic process help
7	0.081	wmic process /?
8	2.077	wmic process get /?
9	0.080	wmic process get /status name osname
10	0.166	wmic process get status
11	0.029	wmic process get /status,name
12	0.216	wmic process get /status,name,osname
13	0.726	wmic process get /?
14	0.168	wmic process get CSName
15	0.486	wmic process get /?
16	0.242	wmic process get name,ExecutablePath,Virtualsize
17	0.202	wmic process get name,ExecutablePath
18	0.202	wmic process get name,ExecutablePath



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

- ロ 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

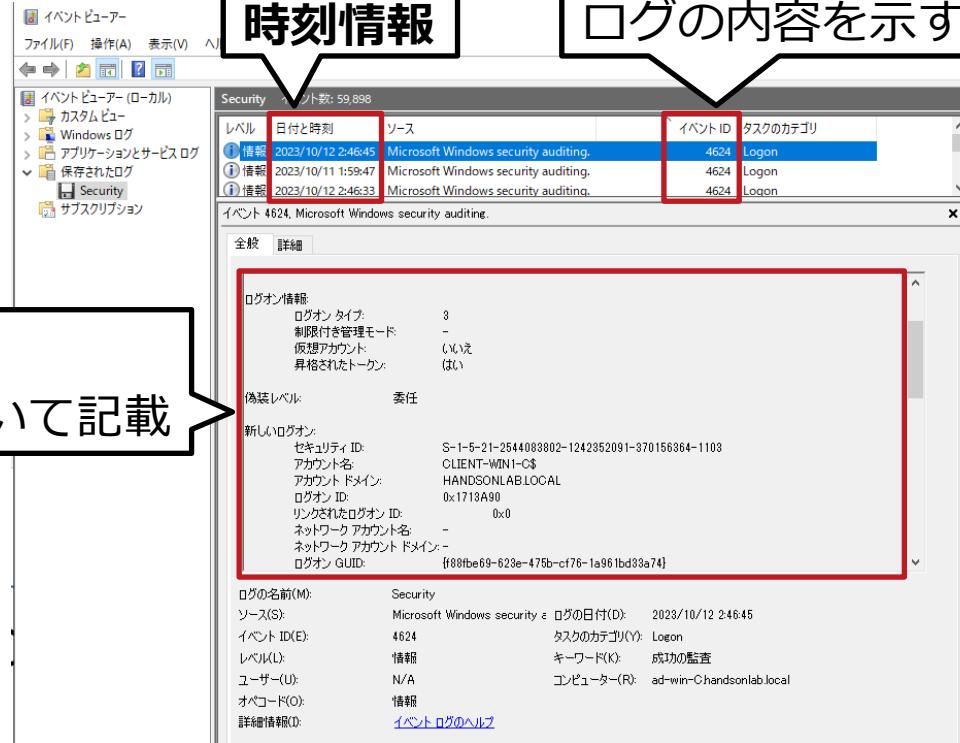
- Windows Client／Server双方に標準搭載されている
- 動作が重い
- 最低限の検索機能がある

## 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）

# イベントビューアー

## イベントビューアーの見方



## 詳細情報

ログの詳細について記載

# 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-3e3b0328c30d}" />
  <EventID>4624</EventID>
  <Version>2</Version>
  <Level>0</Level>
  <Task>12544</Task>
  <Opcode>0</Opcode>
  <Keywords>0x8020000000000000</Keywords>
  <TimeCreated SystemTime="2023-10-11T17:46:45.505811Z" />
  <EventRecordID>48181</EventRecordID>
  <Correlation ActivityID="{c3279d6b-4269-0000-ec27-5d2df2fb901}" />
  <Execution ProcessID="672" ThreadID="4228" />
  <Channel>Security</Channel>
  <Computer>ad-win-C.hands-onlab.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-1-5-21-2544083802-1242352091</Data>
```

# 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）

# 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

イベントビューアー

ファイル(F) 操作(A) 表示(V) ヘルプ(H)

Security イベント数: 59,898

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 2023/10/12 2:46:45 Microsoft Windows security auditing. 4624 Logon

情報 2023/10/11 1:59:47 Microsoft Windows security auditing. 4624 Logon

情報 2023/10/12 2:46:33 Microsoft Windows security auditing. 4624 Logon

イベント 4624, Microsoft Windows security auditing.

全般 詳細

ログオン情報

ログオンタイプ: 3  
制御された登録なし  
仮想アカウント: いいえ  
昇格されたトークン: はい

偽装レベル: 委任

新しいログオン

セキュリティ ID: S-1-5-21-2544083802-1242852091-370156364-1103  
アカウント名: CLIENT-WIN1-C\$  
アカウント ドメイン: HANDSONLAB.LOCAL  
ログオン ID: 0x1713A90  
リンクされたロゴン ID: 0x0  
ネットワーク アカウント名: -  
ネットワーク アカウント ドメイン: -  
ログオン GUID: {f88fbe69-629e-475b-cf76-1a961bd3a74}

ログの名前(M): Security  
ソース(S): Microsoft Windows security イベント ID(D): 2023/10/12 2:46:45  
イベント ID(E): 4624 タスクのカテゴリ(Y): Logon  
レベル(L): 情報 キーワード(K): 成功の監査  
ユーザー(U): N/A コンピューター(R): ad-win-Handsonlab.local  
オペコード(O): 情報  
詳細情報(D): イベントログのヘルプ

ログオンタイプ

ログイン方法を示す情報

# 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。
3	Network	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、プレーンテキスト（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。
10	RemoteInteractive	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>

# 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

# 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 ウィルス対策 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
- 使用しているウィ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

# 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

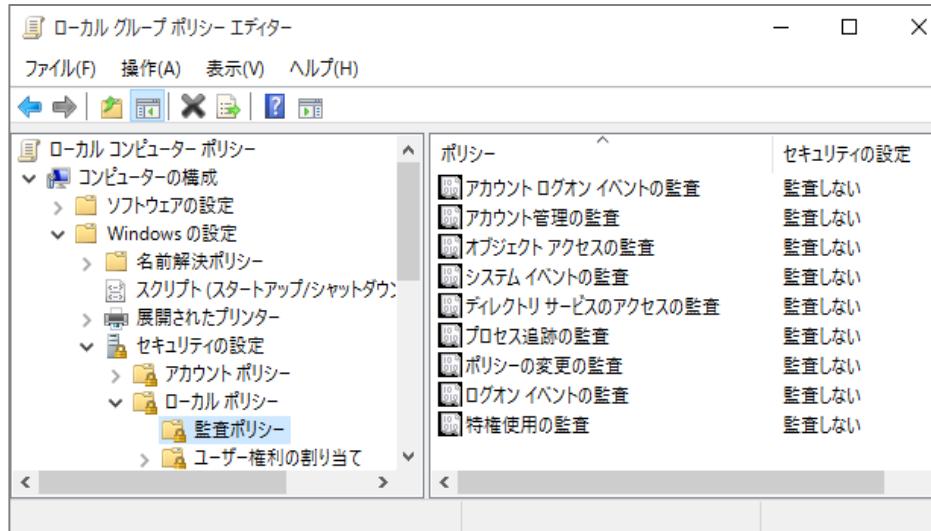
- 監査ポリシーを有効にすることで、**ログが増加する**
  - ✓ ログのローテーションが早くなり古いログが残りにくくなる
- 監査ポリシーを有効化する場合は、**イベントログの最大サイズの変更**もあわせて検討
  - ✓ イベントビューアーやwevtutilコマンドで変更可能

# 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

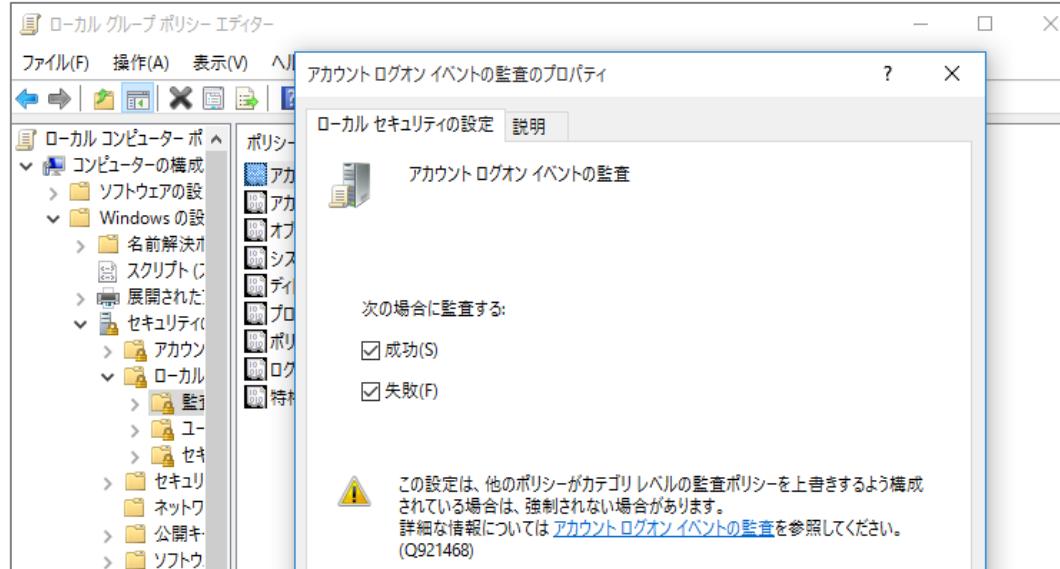
- ✓ [コンピューターの構成] → [Windowsの設定] → [セキュリティの設定] → [ローカル ポリシー]  
→ [監査ポリシー]



# 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>

# 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)

# 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>

# 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>

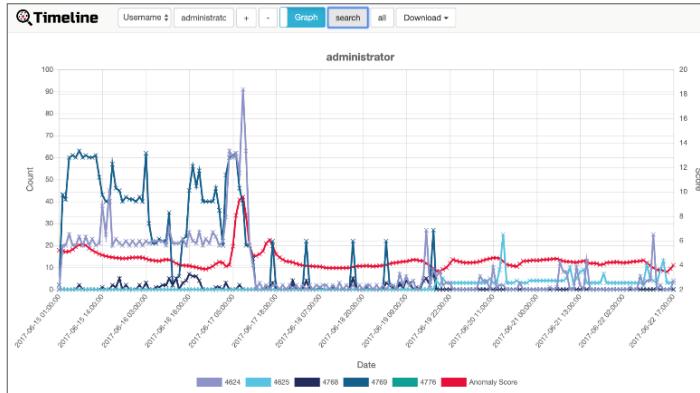
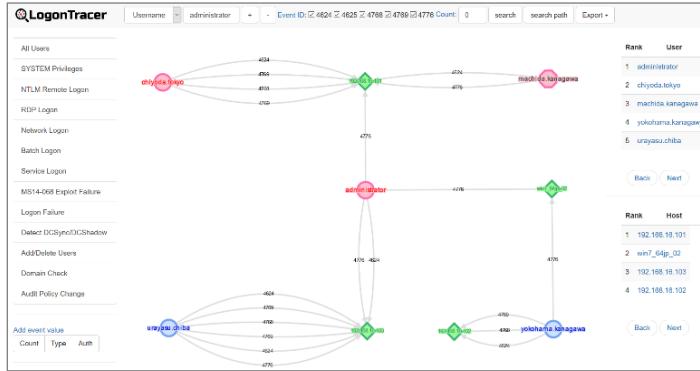
# LogonTracer

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>

# 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を検知
  - 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

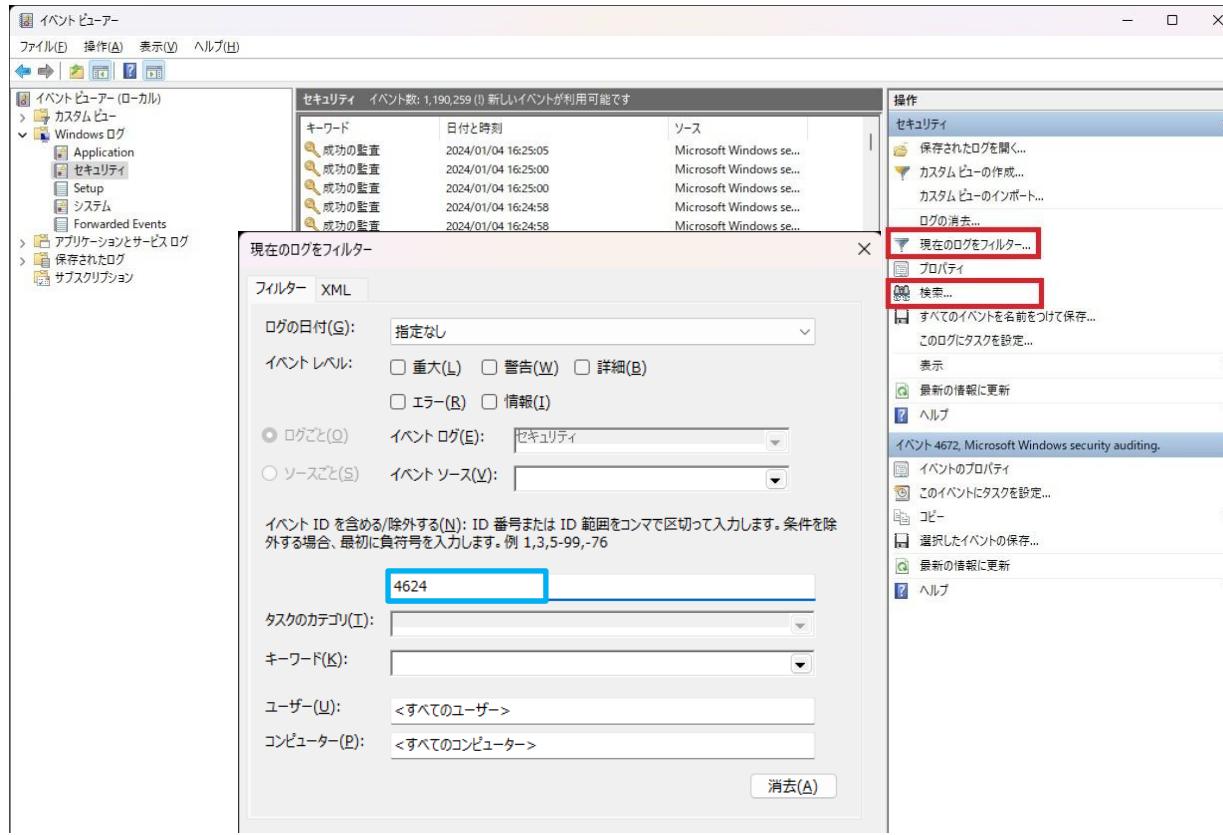
## 2. 画面右の“検索”や“フィルター”等で絞り込んで目的のログを探す

- 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

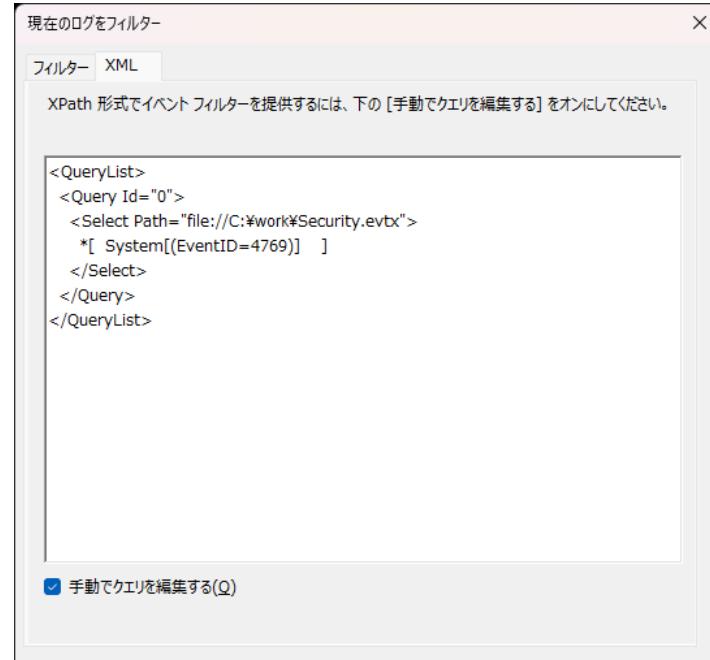
# XPathによるフィルター

イベントビューアーでは、XPath1.0を  
利用可能

□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>



# 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 >

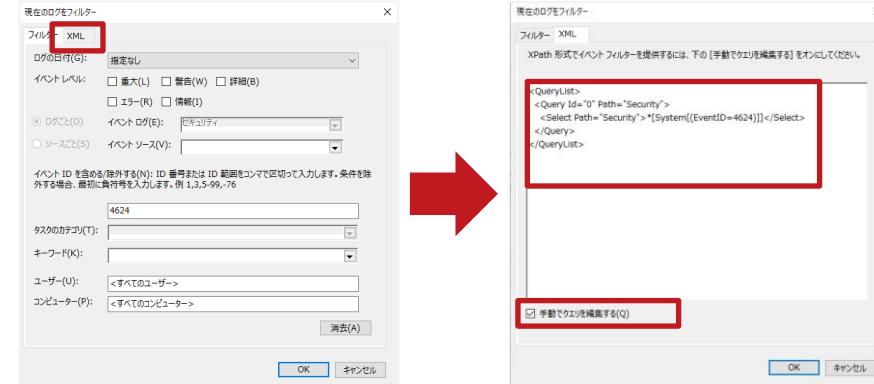
</exampleB>

</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を用いて、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

# 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
```

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

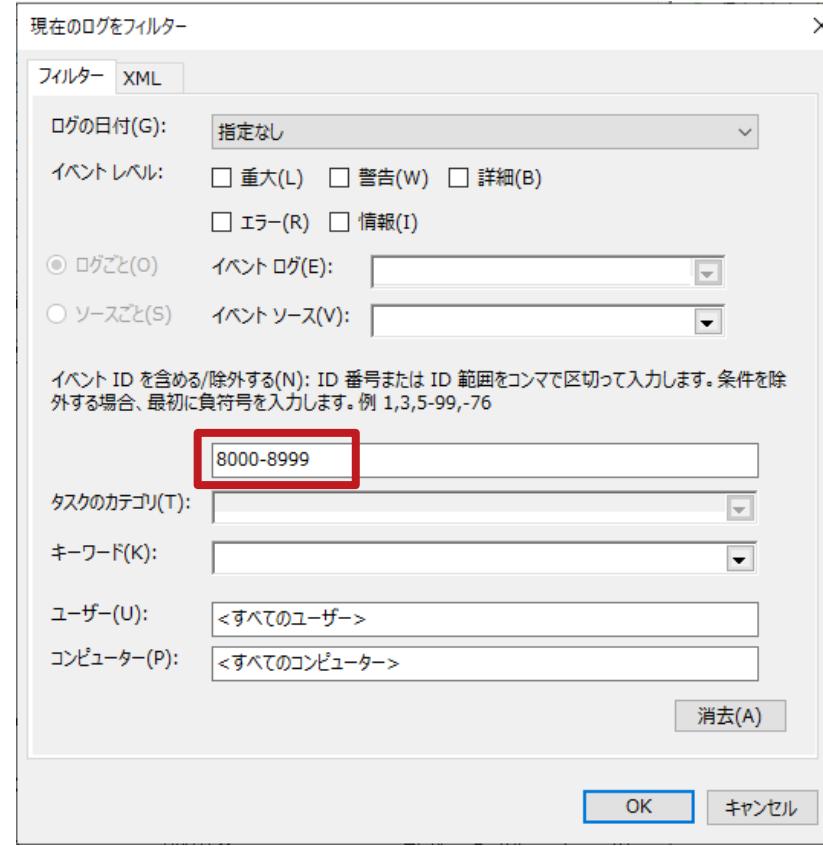
## 問題

- 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



## 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。

## 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

## 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, the navigation pane displays 'イベントビューアー (ローカル)' with options like '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 shows a log titled 'sample1 イベント数: 42,628'. A message indicates how to filter logs. Below is a table of events:

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
情報	2023/07/21 18:45:33	Microsoft Windows security auditing.	4688	Process Creation

A detailed view of the first event (Event ID 4688) is shown at the bottom. It has tabs for '全般' and '詳細'. The 'XML で表示(X)' option is selected. The XML content is partially visible:

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

## 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
```

## 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。

## 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

## 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

The screenshot shows the Windows Event Viewer interface. On the left, the navigation pane lists 'イベント ビューアー (ローカル)' and several log sources: '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'. The main pane displays a log entry for 'sample1' with a total of 42,628 events. The entry details are as follows:

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 17:18:43	Microsoft Windows security auditing.	4624	Logon

A detailed view of the selected event (Event ID 4624) is shown in a modal window. The window has tabs for '全般' and '詳細', with 'XML で表示' (Display as XML) selected. The XML content is partially visible:

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

## 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。

## 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

# ハンズオン 4

演習

回答

The screenshot shows the Windows Event Viewer interface. On the left, the navigation pane lists 'イベント ビューアー' (Event Viewer), 'ファイル(F)', '操作(A)', '表示(V)', and 'ヘルプ(H)'. Below this are sections for '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' (selected), and 'サブスクリプション'. The main pane displays event logs for 'sample2' with a total of 43,179 events. A filter dropdown says 'フィルター: フィルター オプションの設定からフィルターの構成を表示するには'.

Events listed:

- 情報 2023/07/21 18:18:34 TaskScheduler 141 の登録が削除されました
- 情報 2023/07/21 18:18:34 Windows Defender 5001 なし
- 情報 2023/07/21 18:18:34 Microsoft Windows security audit... 5379 User Account Management
- 情報 2023/07/21 18:18:34 Microsoft Windows security audit... 5379 User Account Management
- 情報 2023/07/21 18:18:33 Microsoft Windows security audit... 5379 User Account Management
- 情報 2023/07/21 18:18:33 Microsoft Windows security audit... 5379 User Account Management

A callout box highlights the second event with the text 'Windows Defenderの停止'.

On the right, a detailed view of event 5379 is shown. The title is 'イベント 5379, Microsoft Windows security auditing.' and it has tabs for '全般' and '詳細'. The '全般' tab contains the message '資格情報マネージャーの資格情報が読み取られました。' and the '詳細' tab shows the following details:

サブジェクト:
セキュリティ ID: S-1-5-21-121538365
アカウント名: testadmin001
アカウント ドメイン: client-win1-C
ログオン ID: 0xECFBDB6
読み取り操作: 資格情報の列举

A callout box highlights the account name 'testadmin001' in the subject details with the text 'Windows Defenderの停止時にログインしているアカウント'.

At the bottom of the event details, a note states: 'このイベントは、ユーザーが資格情報マネージャー内に格納されている資格情報の読み取り操作を実行したときに発生します。'