ANTI-MALWARE AND ANTI-CYBERATTACKS ENGINEERING WORKSHOP

MWS Cup 2025
表層解析課題

解説資料

作問チーム紹介

- OSINT/表層データ分析 課題担当
  - 石田 裕貴
    - 国立研究開発法人情報通信研究機構　専門研究員
    - 株式会社日立システムズ
    - 東京情報大学　特別研究員
  - 池澤 隆人
    - アマゾンウェブサービスジャパン合同会社
- マルウェア分類 課題担当
  - 杉山 孔亮
    - 早稲田大学

作問方針

- OSINT情報の収集と分析
  - 生成AI（Web検索特化）が普及して、簡単な調べ物はすぐに答えを得られる
  - AIが出す答えを正しいと判断するのは人であることから、様々な情報源や限られた情報を精査し、読み解く課題を目指した
- マルウェア分類
  - データの分析は、マルウェア分類のコア技術である
  - 得るラベルを自由に選択できるが、得られる数に制限があるときに、どのラベルを得るのかを考える必要がある
  - データ分類の基礎となる、データを見る力を問う課題を目指した

課題ヒント

ハッシュ値系の問題の解法

- OSINT調査の第一歩は、検索エンジンを活用することです。
  - 例えば、Google検索してみたりすると色々出てくると思います。

- また、インターネット上の様々なサービスを活用してみてください。
  - サンドボックスツール
    - Any.run
    - Tria.ge
    - Hybrid Analysis
  - マルウェアインテルDB
    - Malpedia
    - MalwareBazaar
    - VirusTotal

Malware Configとは

- マルウェアが動作するための設定情報のこと

- このファイルには様々な設定値が記載されています
  - C2サーバのアドレス
  - 実行パラメータ
  - 固有設定情報など

- これらの設定値はマルウェアを動かすことで入手できます
  - ただ、今回はマルウェアの動作は必要ありません。
  - つまり、、、、

MITRE ATT&CK - 1

- 実際のサイバー攻撃で使われる戦術（Tactics）と技術（Techniques）を体系的に整理したナレッジ

- MITRE ATT&CKは、主に次の2つの要素で構成されています。
  - 戦術（Tactics）：攻撃段階の目的（例: 初期アクセス、実行、権限昇格など）
  - 技術・サブ技術: 各フェーズの具体的な実現方法（例: PowerShell実行, DLLインジェクション）

- MITRE ATT&CKから得られる情報。
  - 緩和策（Mitigations）：各攻撃手法に対応する防御や回避策
  - 攻撃者グループ（Adversary Groups）：実際にその手法を用いた攻撃グループの情報
  - ソフトウェア（Software）：攻撃に用いられたツールやマルウェア名

MITRE ATT&CK - 2

戦術 (Tactic)	技術 (Technique)	ID	内容の例
初期アクセス (Initial Access)	フィッシング (Phishing)	T1566	添付ファイル付きのスパムメールで感染を開始
実行 (Execution)	Microsoft Officeマクロの実行	T1204.002	Word文書のマクロを利用してPowerShellを起動
永続化 (Persistence)	自動起動登録やスケジュールタスクの作成	T1053	再起動後もマルウェアが実行されるよう設定
権限昇格 (Privilege Escalation)	サービス実行コンテキストの悪用	T1543	ローカルサービスを利用して権限を取得
防御回避 (Defense Evasion)	レジストリ改変、難読化スクリプト使用	T1112, T1027	検知回避のため設定値変更や暗号化を実施
資格情報アクセス (Credential Access)	認証情報の窃取	T1003	Windows Credential Managerなどから情報抽出
発見 (Discovery)	システム・ネットワーク探索	T1010, T1012, T1018	OS構成、レジストリ、リモートホストを調査
横展開 (Lateral Movement)	SMB/管理共有利用	T1077	管理者共有を利用して他端末へ感染拡大
コマンド&コントロール (C2)	HTTP/HTTPS通信利用	T1071.001	C2サーバと暗号化通信でデータ送受信
影響 (Impact)	他マルウェア（TrickBot, Ryuk）の展開	T1105	外部ペイロードをダウンロードして実行

Yaraルールとは - 1

- YARAルールは、マルウェアを特徴パターンで検出・分類するためのルール定義
  - 主にマルウェア解析や脅威ハンティング、インシデント対応で使われる

- パターンマッチングによるマルウェア識別ツール
  - ファイルやメモリの中から特定の文字列、バイト列、コード断片、レジストリキーなどの特徴を活用
  - 既知のマルウェアやその亜種を検知する

Yaraルールとは - 2

```
rule Emotet_Sample
{
    meta:
        author = "researcher"
        description = "Detects Emotet malware sample"
    strings:
        $a = "Emotet" nocase
        $b = "powershell.exe"
    condition:
        any of them
}
```

ルール名

作成者・日付などのメタ情報

検出対象とする文字列など

検出条件

Yaraルールとは - 3

- Yaraルールはインターネット上で公開されている場合があります
  - GitHub
  - Ransomware.live
  - Malpedia
- また、VirusTotalにヒントが載っている場合もあるかも、、、？

マルウェア分析課題について

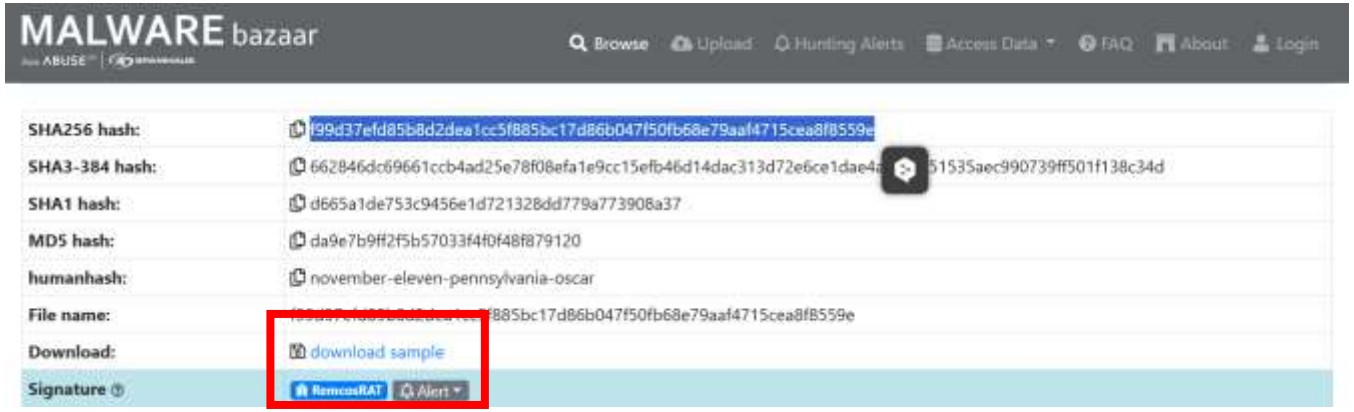
- 今年はチャレンジ課題です！
- 時間内に色々な発想で試行錯誤してください！

- ヒント：
  1. 最も簡単な方法は，ラベル付け可能回数分ランダムにラベル付けし，予測モデル or ルールを構築することです
  2. 機械学習分野において，ラベルの能動的な付与は能動学習 (active learning) と呼ばれています
  3. ラベル付けの方法は運営が用意したサーバへの問い合わせに限りません

解説資料

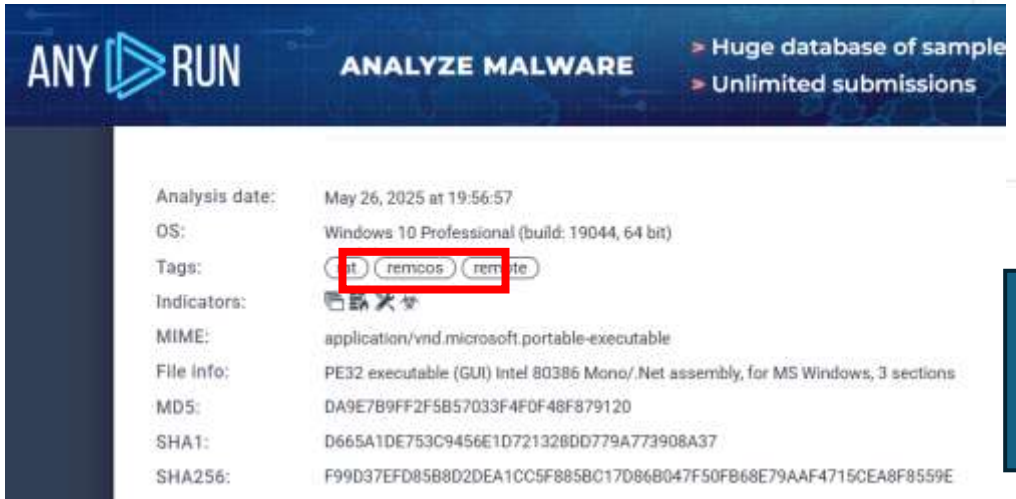
大問1-解説1

- 設問1：ハッシュ値から、マルウェアの正体に迫る
  - f99d37efd85b8d2dea1cc5f885bc17d86b047f50fb68e79aaf4715cea8f8559e



Dアラート

HASH	ed3ea3727b6cbd1a9d61ada18691a1f818684cfe6176d7ec169382d3ba291669 91ae1212b3d4c4112acfe2c2486496ea2fc154187f9ab550bda700b0dc79933e 9ba46e8561485a060cc078670a712929a720b3c0411d6f210bf7212daac1c922 037021326fb46dd4933362d61999395aa16b465da9e1982f0d7afe451363e58a 0f4e6f7614cf00cf85be7db1de34ccc084097896568be62edd797d6b230af7f9 4793e354d8d9a69b5531f22d754242d9b7c0d48abec1c3e9cd4220dd4118c8f1 508fed588de61ca93af80ca5d22262aeff30e7f5ac4ddc10b693004fa86f1325 6abf50de85eba273e1cecaa09feac0e6eefd68cdfac3ea157535f60586b6762f 7ffe8a4132d889f1f228f07b2dae7bb1c0d84f5df0a77491ca51349af83319af acd87909176a110298433ce9bae62ed83f17d8109e9e1a56c24b8287cb515be9 ae27f2678e9feedf735c377121079f2bd43d612d8cd4b59d83cc9406f736267e af5511ac5632d5154382e9c786170158411b672c51232626d2e5c11ba64e8ad3 f99d37efd85b8d2dea1cc5f885bc17d86b047f50fb68e79aaf4715cea8f8559e a19755474c76335eebca3020026b79c750c0a74a8057816a766f6b6ccc1dbb0 a524b0d5bd20b6624fa53e8523aedaaaaf9575be7284e60956d90aa4d9ee031c 43d5c6a0002b66d7af2585366c65c46db1699beec899018e70a67ce53f060c8c 19a9db29d66bd418d0ed7f6c7bd429889fd738b24b16337ad894fae34bbdab69 749a63eca6363c2e605aed49ec04a927eb981d93efe954816f96a5f08030fd05 0247c04b3abd28cc6b1e0fa8a6a49fdda74e978fe603818faf3f85992fe3af04 a707691c49a0a0faa8cfb39c56fd6c37afb9af1f83d85c341ab070cd83826acb 376908c182929b0756563bb13bd162ca5bf915ddfa7cd3e0949573a9b3cc8f7d 92aa70584a20613fbbdbd69b3ca04afdcc228a8f26e7bde02d7c3ca139305422	Remcos



設問1答え：Remcos or RemcosRAT

大問1-解説2

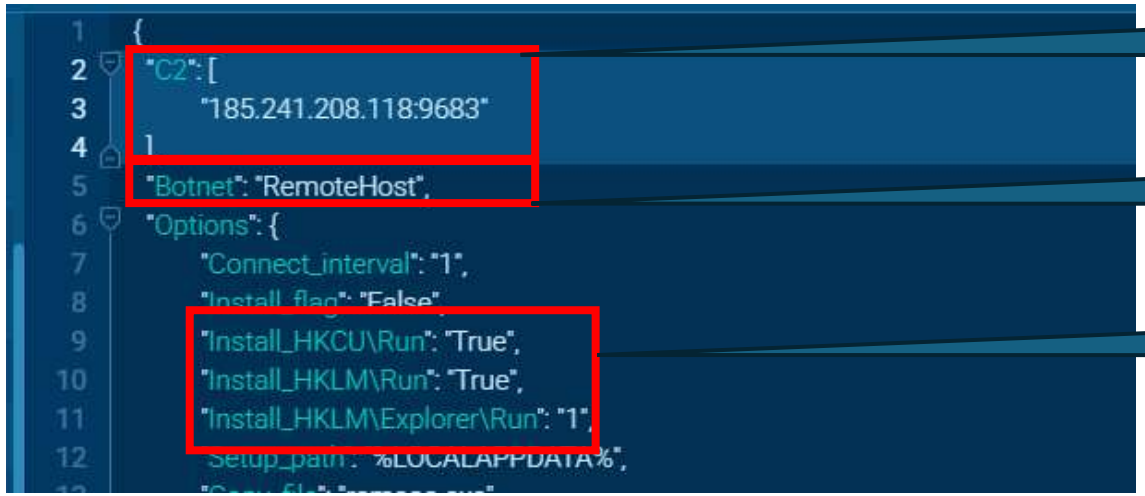
- 設問2：設問1のハッシュ値を上記ファミリ名として特定した根拠または信頼できる情報源を3つ述べよ（記述式解答）

設問2答え：
- さきほどの3つを答えればOK
- 別のものでも何でも良いです

大問1-解説3

- 設問3：設問1の検体のconfigを探し出し、configに含まれる以下の値を答えよ
  - C2
  - Botnet
  - 永続化のためのレジストリキー

- 答え：先ほど見つかっているAny.runを確認する



大問2-解説1

- 統合監視システムのログから、マルウェアの正体に迫る

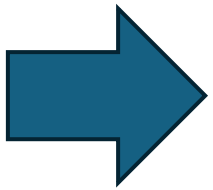
- STIX（Structured Threat Information eXpression）v2
  - サイバー脅威情報を機械可読・構造化して表現する国際標準
  - OASISによって策定・管理（現行はv2.1）
  - フォーマットは軽量なJSON形式

- 今回の課題に含まれるオブジェクト
  - indicator（検知指標：IP,ドメイン,ハッシュなど）
  - attack-pattern（攻撃手法、MITRE ATT&CK連携）

大問2-解説2

- 設問1：JSON中に含まれるMITRE ATT&CKテクニックのうち、永続化で使用される技術を全て答えよ

T1005
T1018
T1021.001
T1021.002
T1021.004
T1033
T1036
T1037
T1047
T1059
T1059.001
T1078
T1105
T1133
T1135
T1219
T1406
T1422
T1424
T1471
T1482
T1490
T1555
T1562
T1567
T1598



ID: T1133

Sub-techniques: No sub-techniques

Tactics: Persistence, Initial Access

ID: T1037

Sub-techniques: T1037.001, T1037.002, T1037.003, T1037.004, T1037.005

Tactics: Persistence, Privilege Escalation

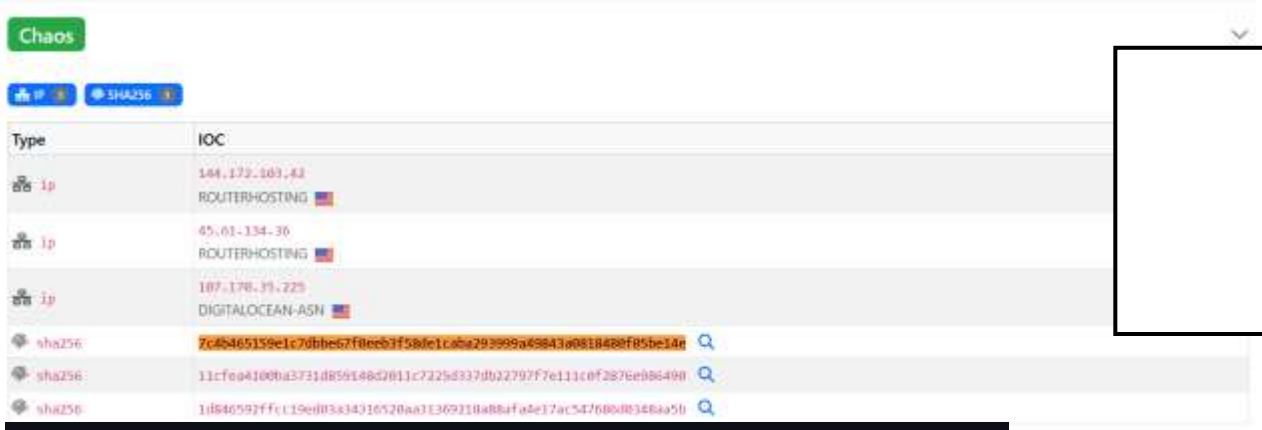
ID: T1078

Sub-techniques: T1078.001, T1078.002, T1078.003, T1078.004

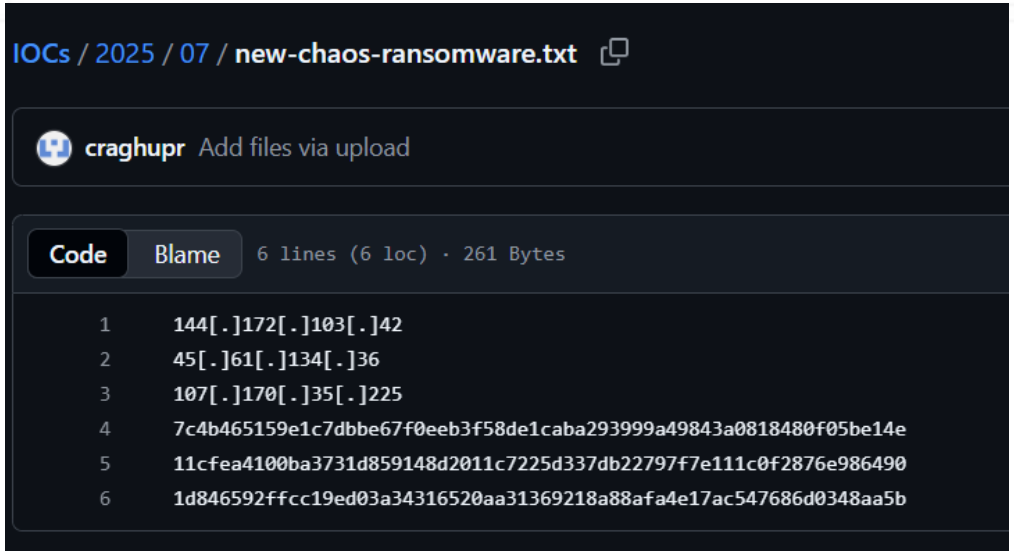
Tactics: Defense Evasion, Persistence, Privilege Escalation, Initial Access

大問2-解説3

- 設問2：観測されたindicatorについて調査を行い，このマルウェアに関係する攻撃アクターの名称を答えよ

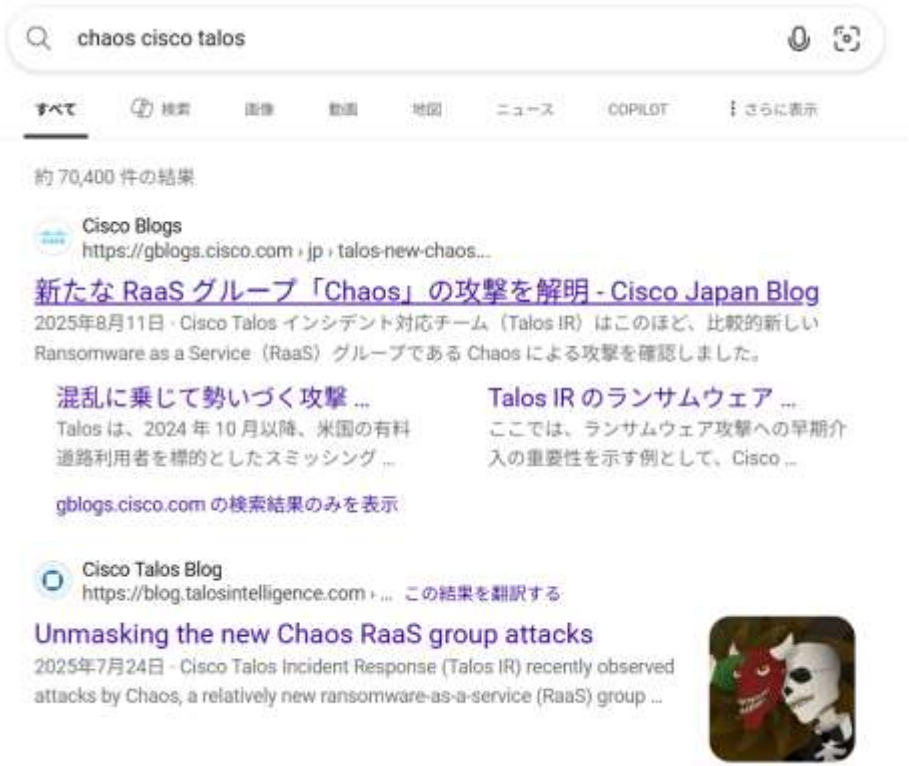


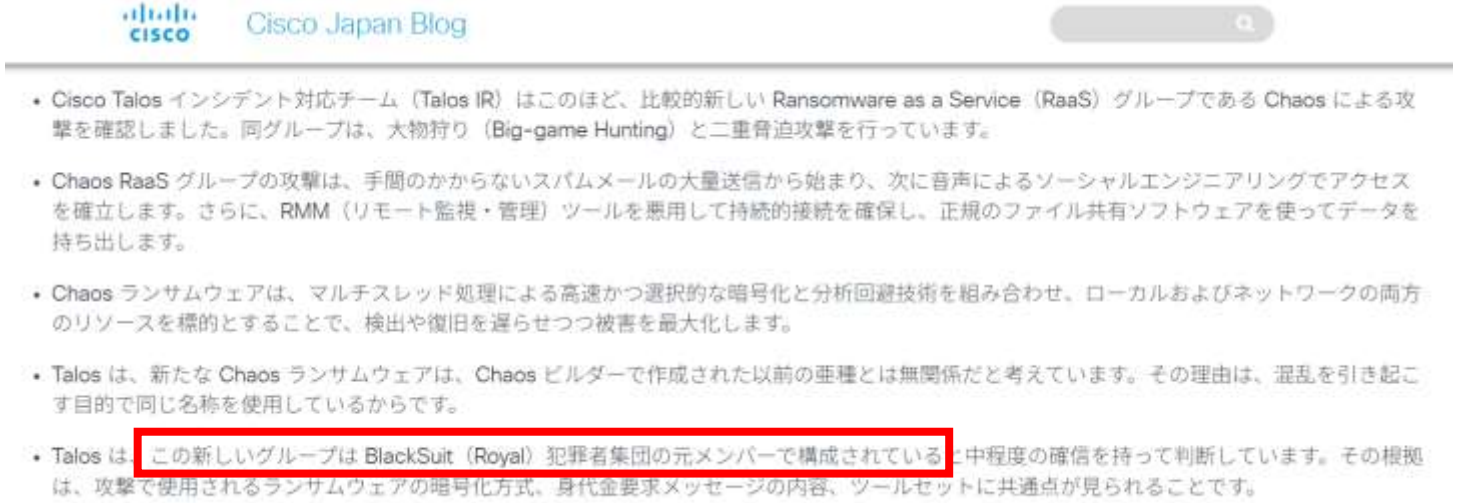
解法：Google検索してみる
答え：Chaos



大問2-解説4

- 設問3:設問2で回答した攻撃アクターについて調査しているCISCO Talosのレポートを捜索し、そのレポート中でこの攻撃アクターとの関係が示唆されている別の攻撃アクターの名称をすべて答えよ。

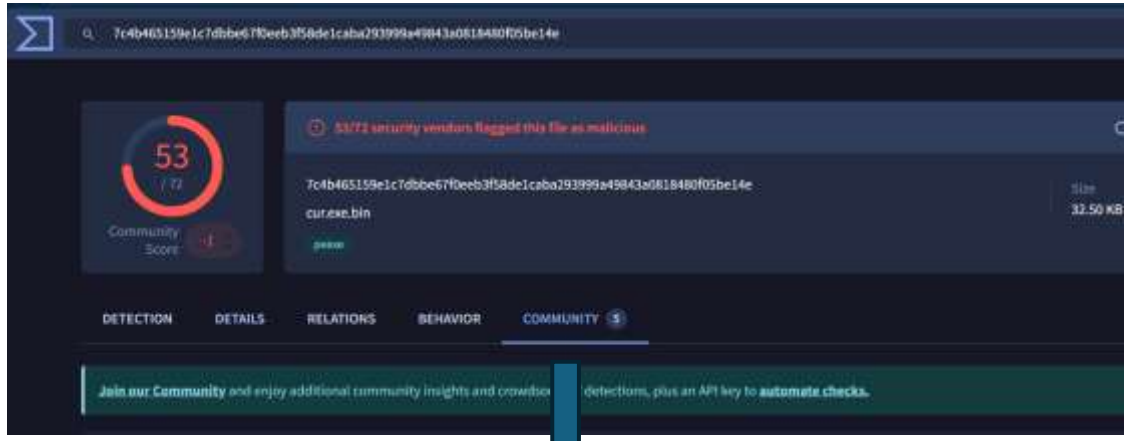




設問3答え：
BlackSuit;Royal

大問2-解説5

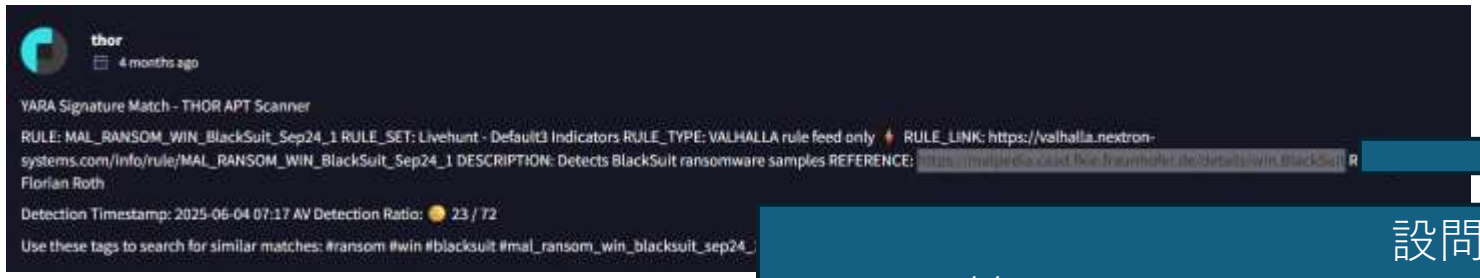
- 設問4:JSONに含まれる検体を検出するためのyaraルールを VirusTotalを起点に探し出し、yaraルールのURLを回答せよ



Yara Rules

```
▼ [TLP:WHITE] win_blacksuit_auto (20241030 | Detects win.blacksuit.)
rule win_blacksuit_auto {

    meta:
        author = "Felix Bilstein - yara-signator at cocacoding dot com"
        date = "2024-10-31"
        version = "1"
        description = "Detects win.blacksuit."
        info = "autogenerated rule brought to you by yara-signator"
        tool = "yara-signator v0.6.0"
        signator_config = "callsandjumps;datarefs;binvalue"
        malpedia_reference = "https://malpedia.caad.fkie.fraunhofer.de/details/win.blacksuit"
        malpedia_rule_date = "20241030"
        malpedia_hash = "26e26953c49c8efafbf72a38076855d578e0a2e4"
        malpedia_version = "20241030"
        malpedia_license = "CC BY-SA 4.0"
        malpedia_sharing = "TLP:WHITE"
```



設問4答え：
https://malpedia.caad.fkie.fraunhofer.de/details/win.BlackSuit

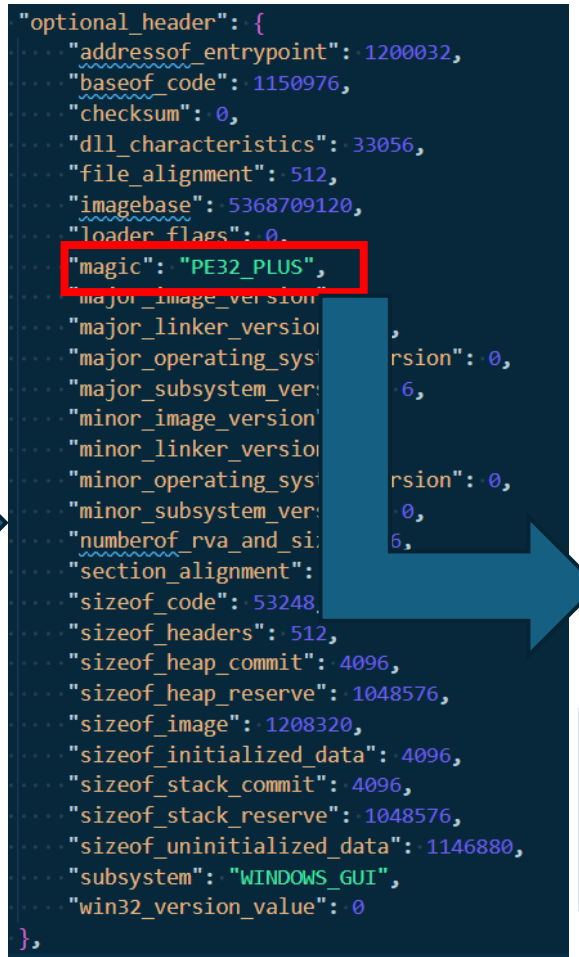
大問3-解説

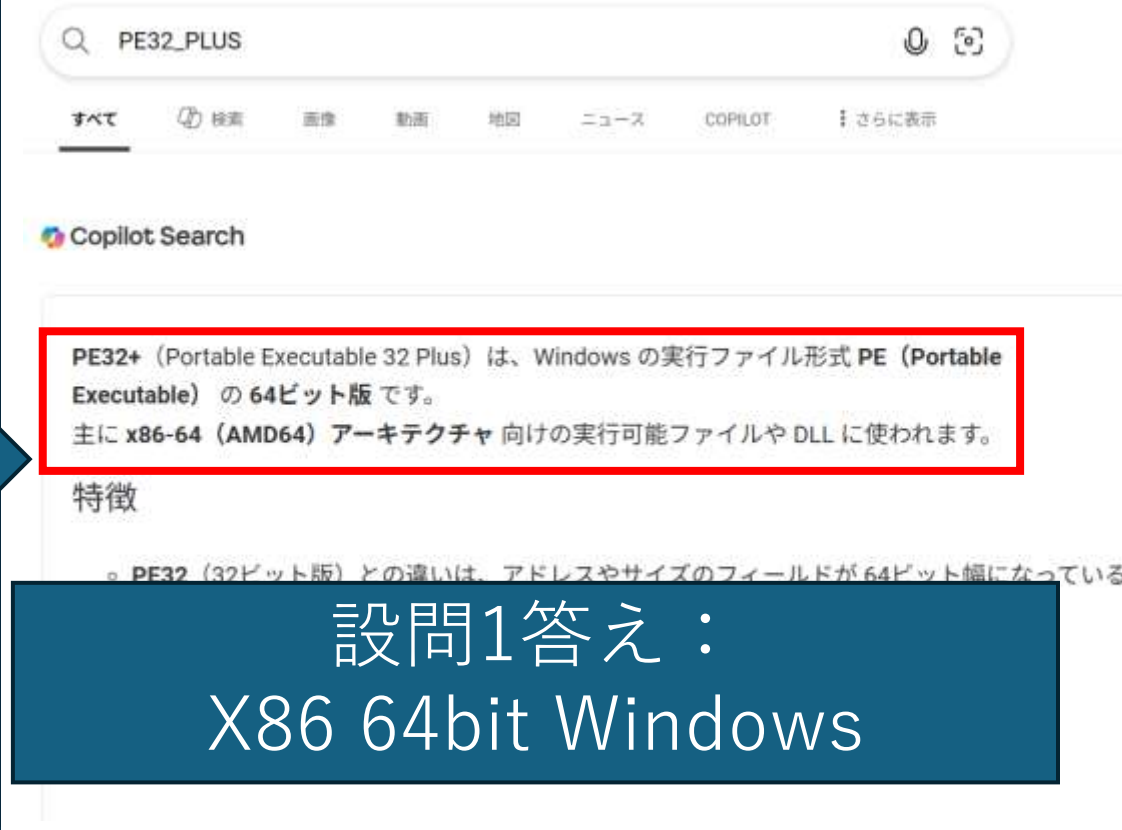
- 設問1：下記のハッシュ値には複数のマルウェアが含まれている。最も多いマルウェアのファミリを選択肢から選べ。

ハッシュ値	ファミリ名
8d8fa3e6cd440a6b01feceb1c96fd0f59fb1e14ca004f8b733b03a957a653d0e	Lumma Stealer
1c2ec4c72c2f31a327b6ba4dfe27a607d311578e25d96cf34c54845eea986f36	Lumma Stealer
b5fb602add12067761efa2188bd0f2e5f70b572b302002723957edebc6e6abd8	Lumma Stealer
3aa011528c4d261a82a0698a5be19d47c4114e2443b93617978fe7f34957930f	Lumma Stealer
fc0fa7c20adf0eaf0538cec14e37d52398a08d91ec105f33ea53919e7c70bb5a	Remcos
6040690d4a93599a2d6408673973ae0e13b06fc8235902de2bbe543f7a1654f6	Phobos
027cc450ef5f8c5f653329641ec1fed91f694e0d229928963b30f6b0d7d3a745	Petya
40cd9af379380f67324675924f1f0f9a948a5940e1e76792a74b009a9e733f92	Lumma Stealer
c3d05ab9b173ea57cafa187bdaaf88be04fb8f13c87a21ff6baafb0bb8eeeb72	Lumma Stealer
752e5cf9e47509ce51382c88fc4d7e53b5ca44ba22a94063f95222634b362ca5	Petya
64b0b58a2c030c77fdb2b537b2fcc4af432bc55ffb36599a31d418c7c69e94b1	Petya
8710e87642371c828453d59c8cc4edfe8906a5e8fdfbf2191137bf1bf22ecf81	Remcos
de5992f7c92351d1011fbece2d4bf74ecfc3b09f84aedb12997a2c3bf869de2c	Remcos
2704e269fb5cf9a02070a0ea07d82dc9d87f2cb95e60cb71d6c6d38b01869f66	Phobos
a91491f45b851a07f91ba5a200967921bf796d38677786de51a4a8fe5ddeafd2	Phobos
2468e5bb596fa4543dba2adfe8fd795073486193b77108319e073b9924709a8a	Lumma Stealer

大問4-解説1

- 表層解析ツールLIFEの結果を読み解く問題
- 設問1：この実行ファイルはどのようなマシンを対象にしていますか？





設問1答え：
X86 64bit Windows

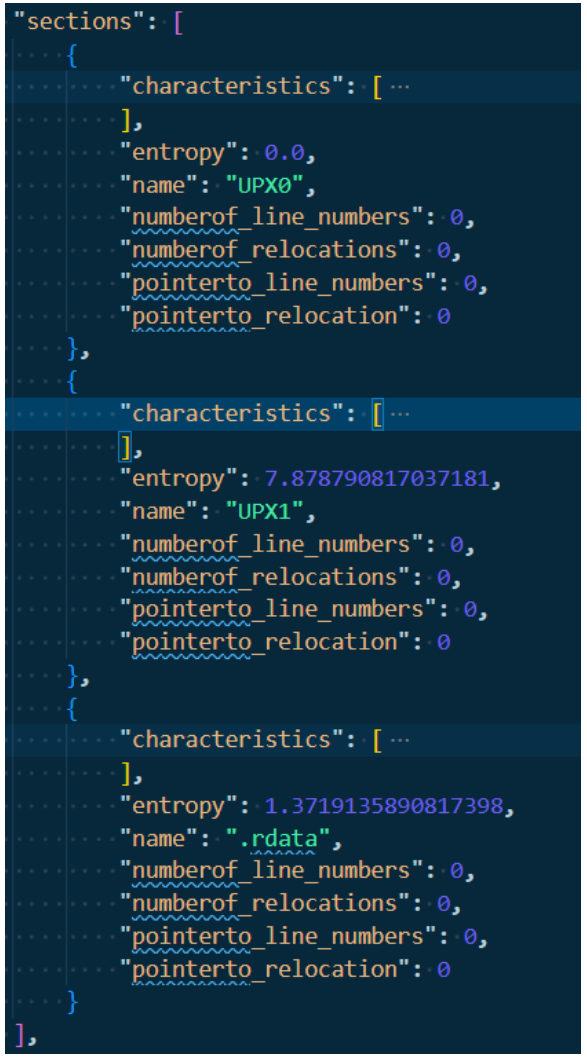
大問4-解説2

- 設問2：次の文章はインポートされている関数に関する説明である。インポートされている関数名を答えよ。
  - 指定されたモジュール（DLL）を呼び出し元プロセスのアドレス空間にロードする関数：
    - LoadLibraryA
  - 呼び出し元プロセスとその全てのスレッドを終了させる関数：
    - ExitProcess
  - ロードされたDLLからエクスポートされた関数または変数のアドレスを取得する関数：
    - GetProcAddress
  - 呼び出し元プロセスの仮想アドレス空間内の領域の保護を変更する関数：
    - VirtualProtect

設問2答え：
Importテーブルを見ればよい

大問4-解説3

- 設問3：最もエントロピーが高いセクション名を答えよ



セクション名	エントロピー
UPX0	0
UPX1	7.878790817037181
.rdata	1.3719135890817398

設問3答え：
UPX1

大問5のモチベーション

予測モデルを実際に構築する場面では．．．

- 訓練データセットの収集・作成から始まる

  → 訓練データセットの収集・作成計画が必要

- 正確なラベル付けはコスト（人件費・工数）が高い

  → 集めたインスタンス全てに付与できるとは限らない

本課題のメインの設定

ラベル付けをするインスタンスを全て選ぶ必要がある

大問5の特徴

1. ラベルを付与する訓練インスタンスを選ぶ必要がある
2. 多くのデータ分析コンペティションよりも競技時間が短い
3. （一部の）テストデータに対する精度の確認ができない
4. 考えたアプローチのやり直しができない

事前知識に基づいて，事前に計画を練る必要がある

先に練るべき計画

1. 使用する学習アルゴリズム or 分類ルール

2. ラベルを付与するインスタンスの選び方

基本方針 1

使用する学習アルゴリズム or 分類ルール

- 事前知識：
  - マルウェア分類はルールベースによる分類と相性がいい
    （e.g. YARAルール，APIのimportの有無，データが正規分布に従わない）
  - strings情報は難読化されていることも多く，
    YARAルールのみでの分類は難しい
- 採用： 決定木を用いた機械学習アルゴリズム
  - Random Forest[1], lightGBM[2], XGBoost[3], catBoost[4] etc…
  - これらのアンサンブルを採用するのが無難
    （実行時間に厳しい制約が無いため）

基本方針２

ラベルを付与するインスタンスの選び方

- 最も簡単な方法：1000個ランダムに選ぶ
- 選択的にインスタンスを選ぶ方法：**能動学習 (active learning)**[5]
  - 基本的な手順：
    1. 少量個ランダムに選んでラベル付け
    2. ラベル付きデータで予測モデルを学習
    3. 予測モデルの予測結果の不確実性が高いサンプルにラベル付け
    4. 2,3を繰り返す
  - 能動学習は，**modAL**[6]というpythonのライブラリで簡単に適用可能
  - 全てランダムに選ぶ場合よりもF1 scoreが0.03~0.05ほど向上

回答例

1. YARAルールを用いて可能なインスタンスにラベル付け
2. 初期のラベル付けインスタンスの決定
   - 1.でのラベル付け結果とクラスタリングによる代表インスタンスの選定により，ランダムよりも効率的にラベル付けするインスタンスを選択
3. 現時点のラベル付きサンプルで予測モデルを学習
   - lightGBM, XGBoost, catBoostなどのアンサンブルモデル
   - ラベルなしデータの活用（半教師あり学習[9]による多様体正則化[10]など）
4. 予測モデルの出力に基づいて新たにラベル付けするインスタンスを決定
   - uncertainty sampling[7], entropy sampling[8]など
5. 手順3. 4.を，ラベル付け回数が上限に達するまで繰り返す
6. 残ったラベルなしデータを用いてPseudo Labeling[11]を実行
   - ラベルなしデータに予測モデルで擬似的な教師ラベルを付与し，学習

参考文献

[1] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.
[2] Ke, Guolin, et al. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in neural information processing systems 30 (2017).
[3] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.
[4] Prokhorenkova, Liudmila, et al. "CatBoost: unbiased boosting with categorical features." Advances in neural information processing systems 31 (2018).
[5] Cohn, David, Les Atlas, and Richard Ladner. "Improving generalization with active learning." Machine learning 15.2 (1994): 201-221.
[6] Danka, Tivadar, and Peter Horvath. "modAL: A modular active learning framework for Python." arXiv preprint arXiv:1805.00979 (2018).
[7] Lewis, David D. "A sequential algorithm for training text classifiers: Corrigendum and additional data." *Acm Sigir Forum*. Vol. 29. No. 2. New York, NY, USA: ACM, 1995.
[8] Settles, Burr, and Mark Craven. "An analysis of active learning strategies for sequence labeling tasks." *proceedings of the 2008 conference on empirical methods in natural language processing*. 2008.
[9] Blum, Avrim, and Tom Mitchell. "Combining labeled and unlabeled data with co-training." Proceedings of the eleventh annual conference on Computational learning theory. 1998.
[10] Belkin, Mikhail, Partha Niyogi, and Vikas Sindhwani. "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples." Journal of machine learning research 7.11 (2006).
[11] Lee, Dong-Hyun. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks." Workshop on challenges in representation learning, ICML. Vol. 3. No. 2. 2013.

さいごに

Towards：次回以降（サステナビリティ）

- 持続可能なMWS Cupの実施のために、
  作問に協力していただけるメンバーを募集しています。

特に、表層解析課題を担当してチームはCup委員が一番少ないです！
ぜひ皆様のご協力を、よろしくお願いいたします！！！！！

お願い

- WriteUpの共有をお願いします！
- 今回、WriteUpに得点は付きませんが、どんな取り組みをしたのか、良かった点、悪かった点などをぜひ教えて下さい。
- 次回以降のMWSCupの参考にさせていただきます。