MWS Cup 2019 課題1

2019/12/9

株式会社ソリトンシステムズ
荒木 粧子

今年の課題1

Drive-by Download
トラフィックから課題1-1を出題

問題を解いて、構成や出題文面
などのアドバイスや改善を支援

Drive-by Downloadで投下され
るマルウェアから課題1-2を出題

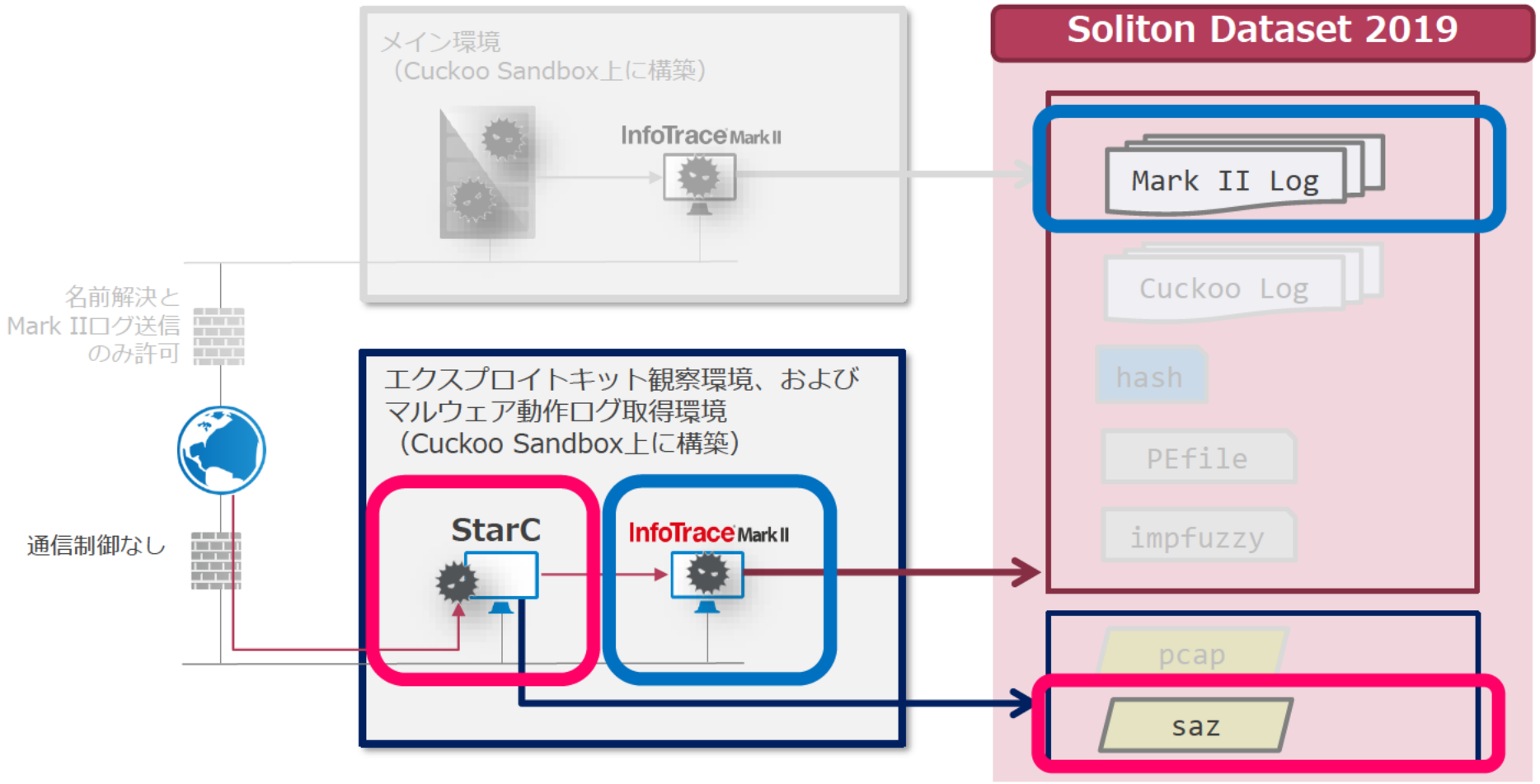
課題1 動的解析（DFIR）

- 目的
  - 実環境で観測されたDrive-by Downloadトラフィックと
    そこで得られた検体の動作ログから侵害を明らかにする
- 概要
  - Soliton Dataset 2019で提供されているデータと
    同じフォーマットの以下のファイルの解析

    - 課題1-1　Drive-by Downloadトラフィック（15点）
      - sazファイルで提供、Fiddler＋EKFiddleを利用して解析

    - 課題1-2　マルウェア動作ログ（10点）
      - テキストファイルで提供、Grep、同梱MK2Tree（Pythonツール）、Excel、独自ツールなどで解析

出題ファイル



課題1-1　課題1-2

課題1-2 マルウェア動作ログ解析

- 1-2-1 以下のWindowsの機構のうち、このマルウェアが永続化に利用したものを答えなさい。（1点）
  - Runレジストリ(T1060)
  - Windowsサービス(T1050)
  - タスクスケジューラ(T1053)
  - ショートカット(T1023)

  ※記載の番号はATT&CKのTIDです：https://attack.mitre.org/tactics/TA0003/

- 1-2-2 永続化されたマルウェア実行ファイルの所在(フルパス）とSHA256ハッシュ値および、永続化を実行したことが確認できるログのシリアル番号（SN）を記載しなさい。(1点 x 3の計3点)

- 1-2-3 このマルウェアの、この端末上での最終的な目的は何と推測できますか。目的と思われる特徴的な動作を2つ取り上げ、それぞれ何をしているのかをその根拠とともに推測し説明しなさい。(各3点 x 2の計6点)

1-2-1 解答：Windowsサービス（T1050）

■ sc.exeによるサービス登録

06/19/2019 06:18:21.921 +0900 **sn=124339 evt=ps subEvt=start** psGUID={18D7ADBE-2B60-4BCB-8C9B-490941536ECF} psPath="C:¥Windows¥System32¥**sc.exe**" **cmd="create tsiakhii binPath= ""C:¥Windows¥system32¥tsiakhii¥hpiydasc.exe /d¥""C:¥Users¥n4o¥AppData¥Local¥Temp¥142ecf4c1a3676df09e0fe183664fa7237569b4f.exe¥""""" type= own start= auto DisplayName= ""wifi support""""** psID=548 parentGUID={8D16B86F-F081-46FF-B832-99DAF182740E} parentPath="C:¥Users¥n4o¥AppData¥Local¥Temp¥142ecf4c1a3676df09e0fe183664fa7237569b4f.exe"

■ service.exeによるサービス登録時のレジストリ作成

06/19/2019 06:18:22.140 +0900 **sn=124354 evt=reg subEvt=setVal** psGUID={E21D6A6A-AB80-4B9F-9D2A-5283C38BF122} psPath="C:¥Windows¥System32¥**services.exe**" path="HKLM¥SYSTEM¥ControlSet001¥services¥tsiakhii" entry="ImagePath" valType=REG_EXPAND_SZ **valStr="C:¥Windows¥system32¥tsiakhii¥hpiydasc.exe /d""C:¥Users¥n4o¥AppData¥Local¥Temp¥142ecf4c1a3676df09e0fe183664fa7237569b4f.exe"""** arc=x86 packed=1 impKrnlCnt=22 **sha256=f31246e291fab40e21e419a9a45ae8d74a18dd5a724647c1ac48b32daae96eeb** size=14764544 sig=None

1-2-2 永続化されるファイルの流れ①

■マルウェア本体の起動
06/19/2019 06:18:15.406 +0900 sn=124237 **evt=ps subEvt=start psGUID={8D16B86F-F081-46FF-B832-99DAF182740E}**
psPath="C:¥Users¥n4o¥AppData¥Local¥Temp¥**142ecf4c1a3676df09e0fe183664fa7237569b4f.exe**"
psID=1616 packed=1 impKrnlCnt=22
sha256=a9b203e8f543256400950dd80dcf8223811efe471db7c6cf2123d0125aa7a68 size=80384
sig=None

■マルウェア本体がhpiydasc.exeというファイルを生成
06/19/2019 06:18:15.640 +0900 sn=124278 **evt=file subEvt=create psGUID={8D16B86F-F081-46FF-B832-99DAF182740E}**
psPath="C:¥Users¥n4o¥AppData¥Local¥Temp¥**142ecf4c1a3676df09e0fe183664fa7237569b4f.exe**"
path="C:¥Users¥n4o¥AppData¥Local¥Temp¥**hpiydasc.exe**" drvType=HDD

■ hpiydasc.exeの書き込み保存
06/19/2019 06:18:20.531 +0900 sn=124291 **evt=file subEvt=close psGUID={8D16B86F-F081-46FF-B832-99DAF182740E}**
psPath="C:¥Users¥n4o¥AppData¥Local¥Temp¥**142ecf4c1a3676df09e0fe183664fa7237569b4f.exe**"
path="C:¥Users¥n4o¥AppData¥Local¥Temp¥**hpiydasc.exe**" drvType=HDD **read=0 write=14764544**
pe=1 arc=x86 packed=1 impKrnlCnt=22
sha256=f31246e291fab40e21e419a9a45ae8d74a18dd5a724647c1ac48b32daae96eeb
size=14764544 sig=None

1-2-2 永続化されるファイルの流れ②

■cmd.exeにより、ファイル移動（/Cでコマンド実行、/Yで上書きチェック無し）
06/19/2019 06:18:21.343 +0900 sn=124324 **evt=ps subEvt=start** psGUID={A809B6C8-DA58-4E96-8B55-FC63508D2223} psPath="C:¥Windows¥System32¥**cmd.exe**" **cmd="/C move /Y ""C:¥Users¥n4o¥AppData¥Local¥Temp¥hpiydasc.exe"" C:¥Windows¥system32¥tsiakhii¥"** psID=2032 parentGUID={8D16B86F-F081-46FF-B832-99DAF182740E}
parentPath="C:¥Users¥n4o¥AppData¥Local¥Temp¥142ecf4c1a3676df09e0fe183664fa7237569b4f.exe" sha256=17f746d82695fa9b35493b41859d39d786d32b23a9d2e00f4011dec7a02402ae

■cmd.exeの実行結果が、ファイルイベントとしても観測されている
06/19/2019 06:18:21.531 +0900 sn=124337 **evt=file subEvt=rename** psGUID={A809B6C8-DA58-4E96-8B55-FC63508D2223} psPath="C:¥Windows¥System32¥**cmd.exe**" path="C:¥Users¥n4o¥AppData¥Local¥Temp¥hpiydasc.exe" drvType=HDD dstPath="C:¥Windows¥System32¥tsiakhii¥hpiydasc.exe" dstDrv=HDD pe=1 arc=x86 packed=1 impKrnlCnt=22
sha256=f31246e291fab40e21e419a9a45ae8d74a18dd5a724647c1ac48b32daae96eeb size=14764544 sig=None

1-2-2 解答

- 永続化されたマルウェア実行ファイルの所在(フルパス)
  - C:¥Windows¥system32¥tsiakhii¥hpiydasc.exe

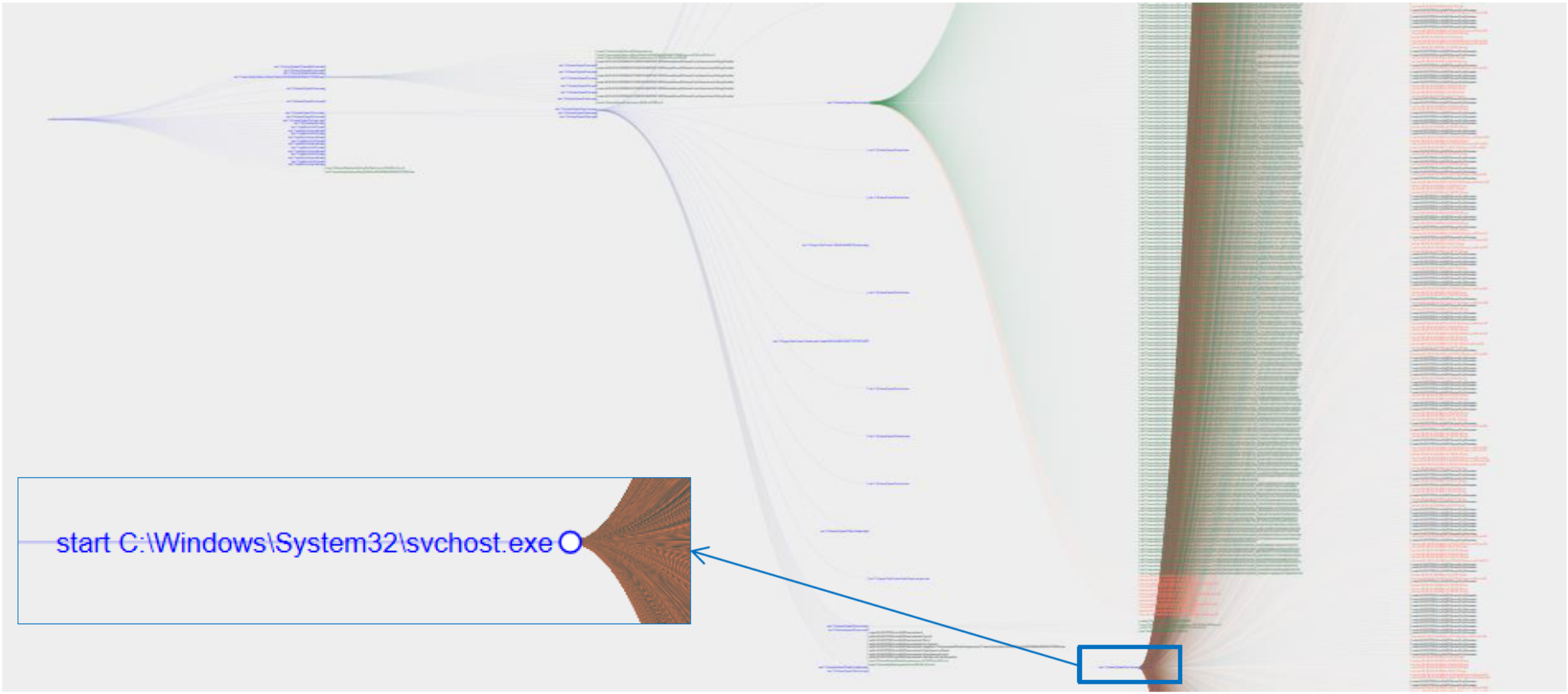
- SHA256ハッシュ値
  - f31246e291fab40e21e419a9a45ae8d74a18dd5a724647c1ac48b32daae96eeb

- 永続化を実行したことが確認できるログのシリアル番号（SN）（以下のいずれでも正解としています）
  - 124339 （sc.exeによるサービス登録）
  - 124354（ service.exeによるサービス登録時のレジストリ作成）

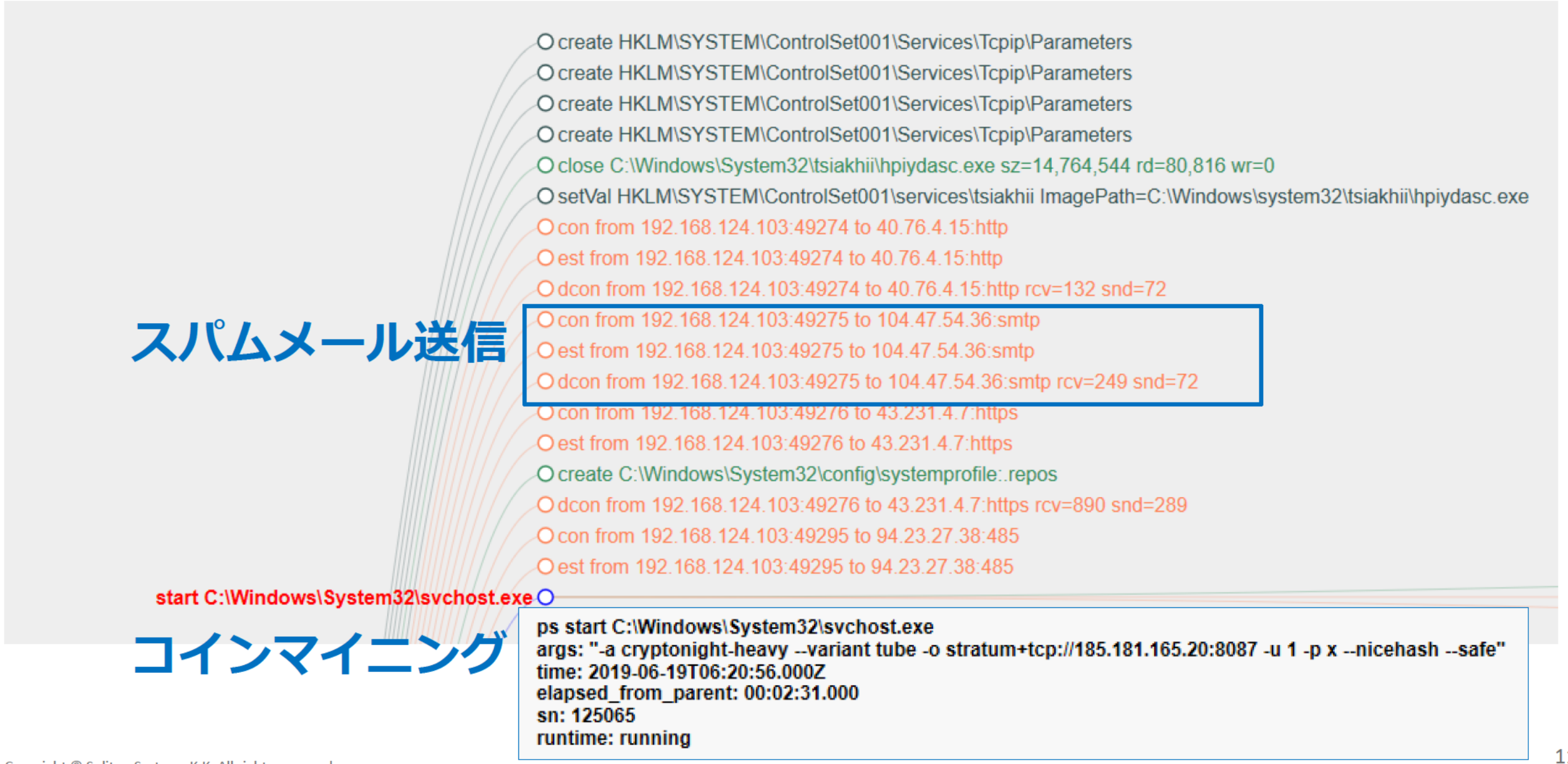
1-2-3 マルウェア挙動の全体像

ココの拡大
次ページ

：
（省略）
：



1-2-3 マイニング・スパムメール送信



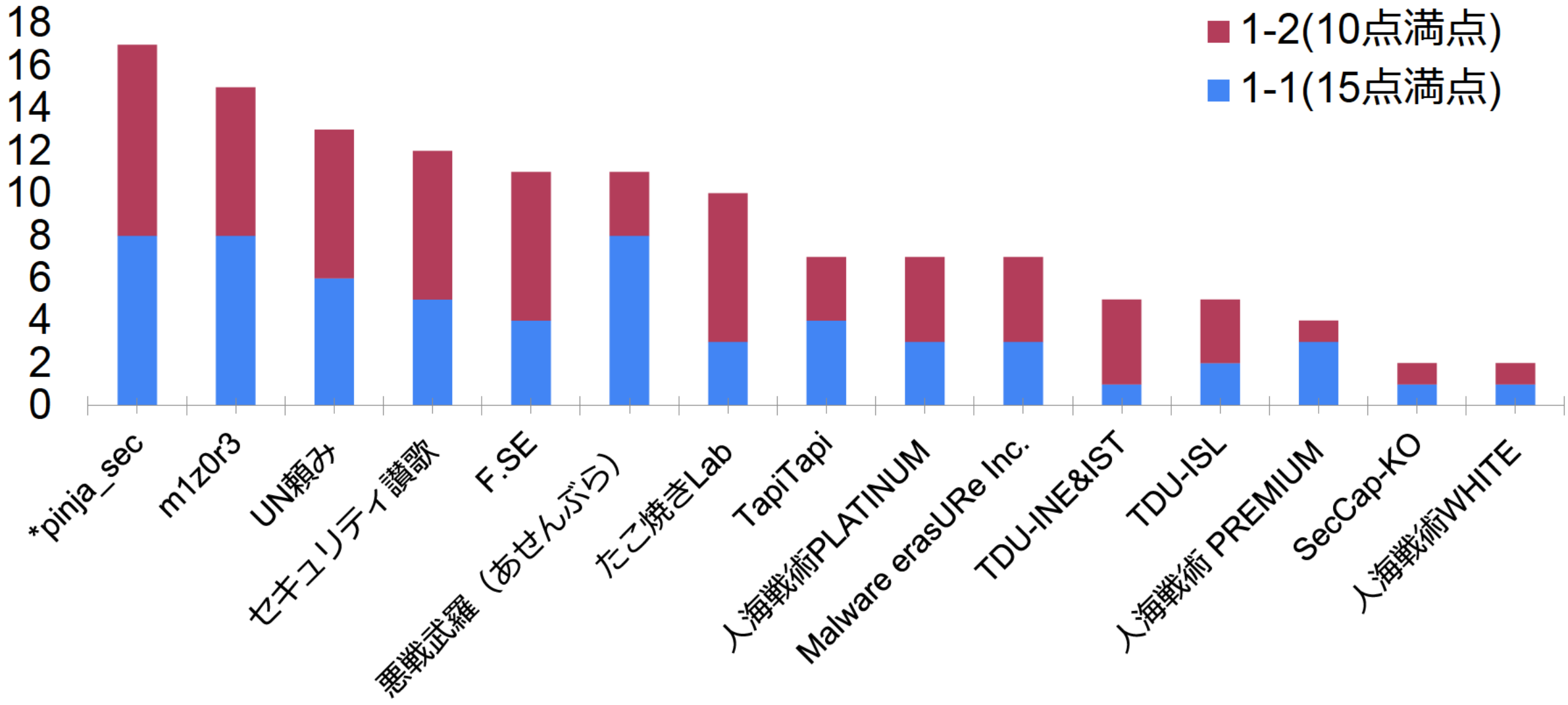
1-2-3 解答

- このマルウェアの、この端末上での最終的な目的は何と推測できますか。目的と思われる特徴的な動作を2つ取り上げ、それぞれ何をしているのかをその根拠とともに推測し説明しなさい。

  - スパムメール送信
    - 様々なIPアドレスの宛先ポート番号25/TCPに対してコネクションを確立していることから推察される
    - sn=124479など

  - コインマイニング
    - マイニングツールのコマンドらしきものが実行されていることから推察される
    - sn=125065など

課題１結果



課題１に関する参加者アンケート（抜粋）

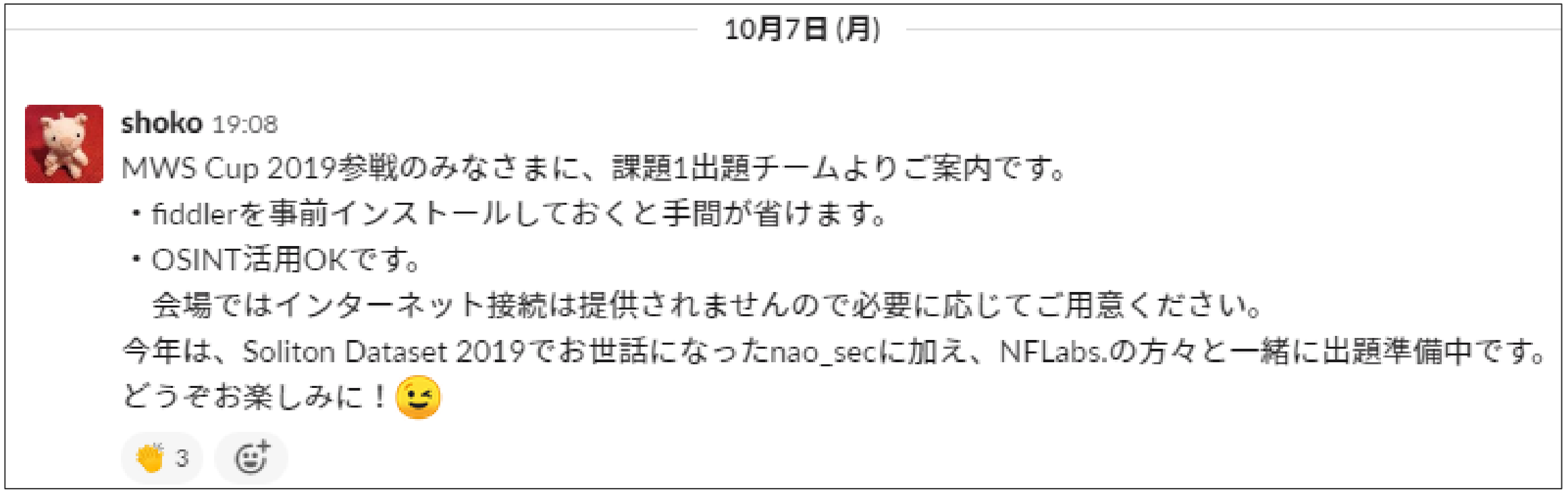
- 暗号関連の問題が分からなかった。オススメの勉強法があれば教えていただきたいです。
  →小池さんからの解説をご参照ください☺

- 自分は課題1-2だけ解いたのですが、怪しい挙動を追ったりチームメイトと相談しながらその挙動を推測していく作業は楽しかったです。(正解しているかどうかは別として....)
- 動的解析ログについてはもう少し細かく問題を作っていただけると解析の流れが見えて面白いかなと思いました。
- 得点を争うCupとなるので、少々むずかしいかもしれないですが、問題文の中にも出題意図や実際での現場での声（例えば、本当の現場ではこのログデータの３倍程度が平均ですなど、）があると、個人的により楽しく問題に取り組めるかと思いました。
  →大量のログから脅威検知するというのは、セキュリティログ解析分野における一つの命題でもあります。通常のWindowsの挙動を知っておくと、解析が早くなります。

- （強いて言えば）EK-fiddleを入れていると一瞬で解けるというお話がありましたが事前にアナウンスがあればより嬉しかったです。競技時間があまり長くない+NWは自分で用意する（通信制限のある形態のテザリング等が多いと想定される）という状況から当日インストールするのは難しいため
  →事前アナウンスしたのですが、周知が足りなかったかもしれません。

ヒント①

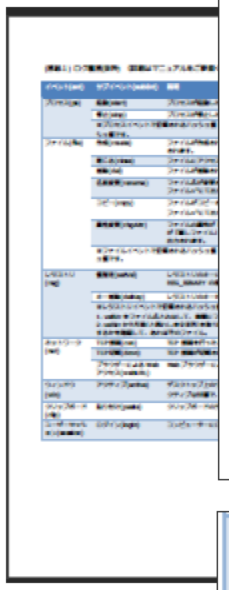
- Fiddlerについては、事前にSlackでご案内していました



Slack-MWS未参加の方は、こちらをご参照ください
https://www.iwsec.org/mws/mws_ml.html

ヒント②

- 問題ファイルにMark IIログ仕様概要や注意点を記載していました



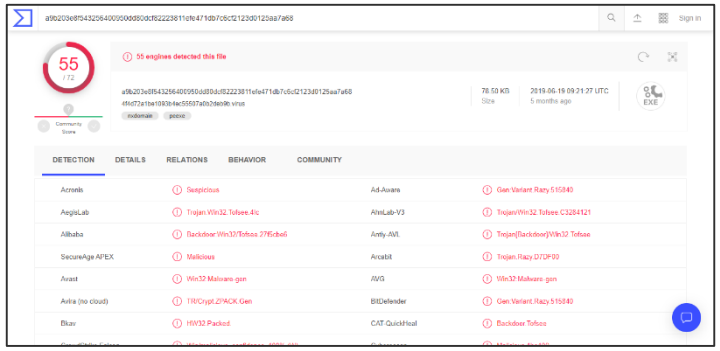
(別紙１) ログ種別(抜粋)　(詳細はマニュアルをご参照ください)

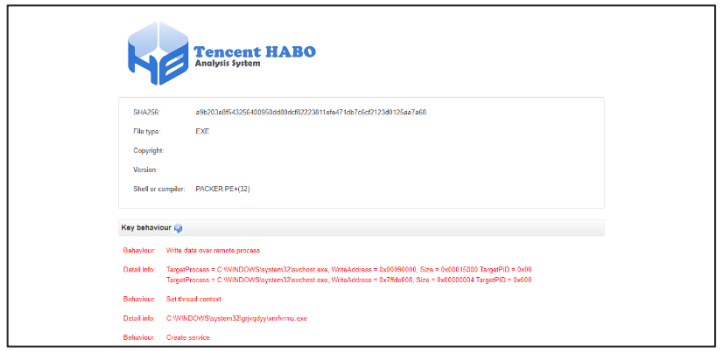
イベント(evt)	サブイベント(subEvt)	説明
プロセス(ps)	起動(start)	プロセスが起動したときに出力されます。
	停止(stop)	プロセスが停止したときに出力されます。
	※プロセスイベントで記載されるハッシュ値（sha256 等）は、psPath に記載されているファイルのハッシュ値です。	
ファイル(file)	作成(create)	ファイルが作成されたときに、対象ファイル path="<ファイルパス>"にて出力
レジストリ(reg)	値指定(setVal)	レジストリのキーに値が新規に設定されたとき、または値（値の種類が REG_BINARY の場合はサイズ）に変化があったときに、出力されます。
	キー削除(delKey)	レジストリのキーが削除されたときに、出力されます。
	※レジストリイベントで記載されるハッシュ値（sha256 等）は以下のファイルのハッシュ値です。 1. valStr をファイル名とみなして、実際にファイルが存在するかを確認して、あればそのファイル。 2. valStr から引数(と思わしき文字列)を取り除いたものをファイル名とみなして、実際にファイルが存在するかを確認して、あればそのファイル。	

※いずれも2019年12月9日時点の仕様です

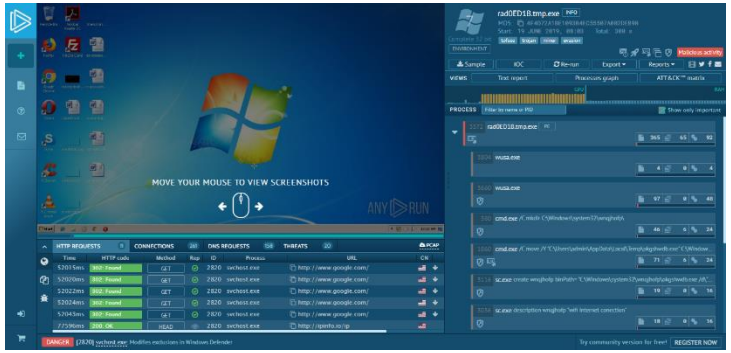
ヒント③ OSINT活用例

- マルウェアのハッシュ値をGoogle検索
  - 例）一次検体
    a9b203e8f543256400950dd80dcf82223811efe471db7c6cf2123d0125aa7a68
- VirusTotal (https://www.virustotal.com/gui/file/a9b203e8f543256400950dd80dcf82223811efe471db7c6cf2123d0125aa7a68/)





- any.run (https://app.any.run/tasks/1b5aa7f2-9829-4b8c-97f5-332e60bc763f/)



※いずれも2019年12月9日時点の情報です

まとめ

- Cupに関する事前情報や、問題文面・提供資料をよくお読みください
  - MWSCup申し込みページやSlack-MWSは要チェック

- 問題作成にご興味のある意欲ある方、また、ご意見・ご不明点などがありましたら、お気軽にSlackまでご連絡ください
  - Slack-MWSの#mwscupチャンネル、DMでもOKです