TERN調査
OpenChain Japan
Tooling Sub Working Group

株式会社　東芝　ソフトウェア技術センター
2020.07.28　Kouki Hama　濵　功樹
Toshiba Corporation
E-mail: kouki1.hama@toshiba.co.jp

Contents

- 01　TERN概要
- 02　TERNインストール方法
- 03　TERN実行方法
- 04　TERN出力結果
- 05　さいごに　Q&A

01

TERN 概要

詳細に入る前に

- コンテナとOSSコンプライアンス上の問題に関する参考情報
  - Docker containers: What are the open source licensing considerations?
    - The Linux Foundation April 24, 2020
    - https://www.linuxfoundation.org/publications/2020/04/docker-containers-for-legal-professionals/
  - Docker Containers for Legal Professionals　Author: Armijn Hemel, MSc.
    - https://www.linuxfoundation.org/blog/2020/04/docker-containers-what-are-the-open-source-licensing-considerations/
- Dockerコンテナの考慮すべき点
  - Dockerコンテナはその内にあるソフトの仕様を理解しなくても. 機能を利用できる場合がある
  - そのため, ライセンス情報などを考慮しないで中のソフトウェアを使うといったことが容易に発生する
  - 開発者はライセンスコンプライアンス問題が発生することも知らずに、各種ソフトウェアを無意識のうちに出荷してしまう可能性がある

次スライドで詳細

Dockerコンテナ：オープンソース ライセンスの考慮事項

- 各Dockerイメージは、1つまたは複数のレイヤーで構成
- ユーザーが使用しているイメージはすべてのレイヤーのビューに過ぎず、実行時には最終的なビューしか見えない
- 初めにあるレイヤーにソフトウェアをインストールしたとしても、(異なったライセンスの) 別バージョンをインストールする可能性がある
- ビューでは古いソフトは削除されたようにみえる
- 別レイヤーには上書きされる前の元のソフトウェアがまだ存在

- 配布にあたり, すべてのレイヤーに含まれるOSSのコンプライアンス義務を把握し遵守する必要がある

Dockerイメージを配布する際に注意すること

- Dockerをどのように配布するか確認
  - Docker fileのレシピのみ配布
  - イメージを配布
  - 上記の混合で配布
- 誰がOSSを配布しているか明らかにする
- 配布しているOSSが何かを明らかにする

TERNとは？

- コンテナ内のソフトウェアパッケージを調査するツール
- OSS
  - https://github.com/tern-tools/tern
  - ライセンス
    - BSD-2 License
  - Python3 で実装
  - 2018年にv0.1.0 を初リリース
    - Vmwareのエンジニアを中心に開発が進められている
    - 2020年7月, 2.1.0が最新のリリースバージョン
      - GitHubではv2.1.1として開発中
  - Scancodeを呼び出しライセンス分析を行うことが可能
    - https://github.com/nexB/scancode-toolkit
  - Cve-bin-toolを呼び出し脆弱性分析を行うことが可能
    - https://github.com/intel/cve-bin-tool

TERN (What is Tern 日本語訳)

1. overlayfs を使用して、最初にコンテナイメージの構築に使われる(BaseOS とも呼ばれる)ファイルシステムレイヤー をマウントする
   - 訳注　参考：Overlayfs : https://docs.docker.com/storage/storagedriver/overlayfs-driver/

2. chroot 環境の「コマンドライブラリ」からスクリプトを実行して、そのレイヤーにインストールされているパッケージに関する情報を収集する

3. 2.で収集した情報を基にして、コンテナイメージ内の残りのレイヤーについても、ステップ1と2の処理を繰り返し行う

4. 情報収集が完了すると、レポートが生成する。これは様々な形式をオプションで指定できる。標準フォーマットのレポートでは、インポートされた様々なソフトウェアコンポーネントの説明をレイヤーごとに記載している。また、Dockerfileが与えられた場合、レポートはファイルシステムのレイヤーに対応するDockerfileの行を表示する。

- 原文
  - https://github.com/tern-tools/tern#what-is-tern

02

TERNインストール方法

TERNインストール方法

現在は4種の方法が用意されている

- ネイティブインストール
  - リリースされてない最新の機能が使いたい場合

- Pypi を利用してインストール
  - https://pypi.org/project/tern/
  - 安定している最新のリリース版を利用したい場合

- Docker　利用インストール
  - 最新のリリース版を利用できるようになる

- Vagrant　利用インストール
  - Ubuntuの仮想環境を立ててTERNをインストールする
  - 仮想環境下でPypiからインストール

TERNインストール手順

現在は4種の方法が用意されている
詳細は　https://qiita.com/K-Hama/items/536a467b9ab68ef7db95　に記載

- ネイティブインストール

```
$ sudo apt-get install -y python3 python3-pip python3-venv attr
$ git clone https://github.com/tern-tools/tern.git
$ cd tern
$ pip3 install -r requirements.txt
$ python3 setup.py install
```

- Pypi (https://pypi.org/project/tern/)を利用してインストール

```
$ sudo apt-get install -y python3 python3-pip python3-venv attr
$ pip3 install tern
```

- Docker　利用インストール

```
$ git clone https://github.com/tern-tools/tern.git
$docker build –t ternd .
```

- Vagrant　利用インストール

```
$ git clone https://github.com/tern-tools/tern.git
$ cd tern/vagrant
$ vagrant up
$ vagrant ssh
```

(参考)Python3の仮想環境下でインストールする場合

仮想環境作成
$ python3 -m venv ternenv
$ cd ternenv

仮想環境利用開始
$ source bin/activate

仮想環境下にternインストール (pip利用の場合)
$(ternenv) pip install tern

仮想環境終了
$deactive

03

TERN実行方法

TERN実行方法

- ネイティブ/Pypi を利用してインストールした場合の実行方法
  $ tern report -o output.html –f html -i debian:buster

- Dockerでインストールした場合の実行方法
  $ ./docker_run.sh workdir ternd "report -i debian:buster" > output.txt

- Vagrant　利用インストール
  $ vagrant ssh #イメージの中に入る
  $ tern report -i debian:buster -o output.txt

- オプションの意味 (共通)
  - [-o]: 出力ファイル
  - [-f] : 出力形式 (json, html, yaml, spdxtagvalue)
  - [-i] : 分析イメージ
  - [-x]: 外部ツールの利用 (scancode ,cve-bin-tool)

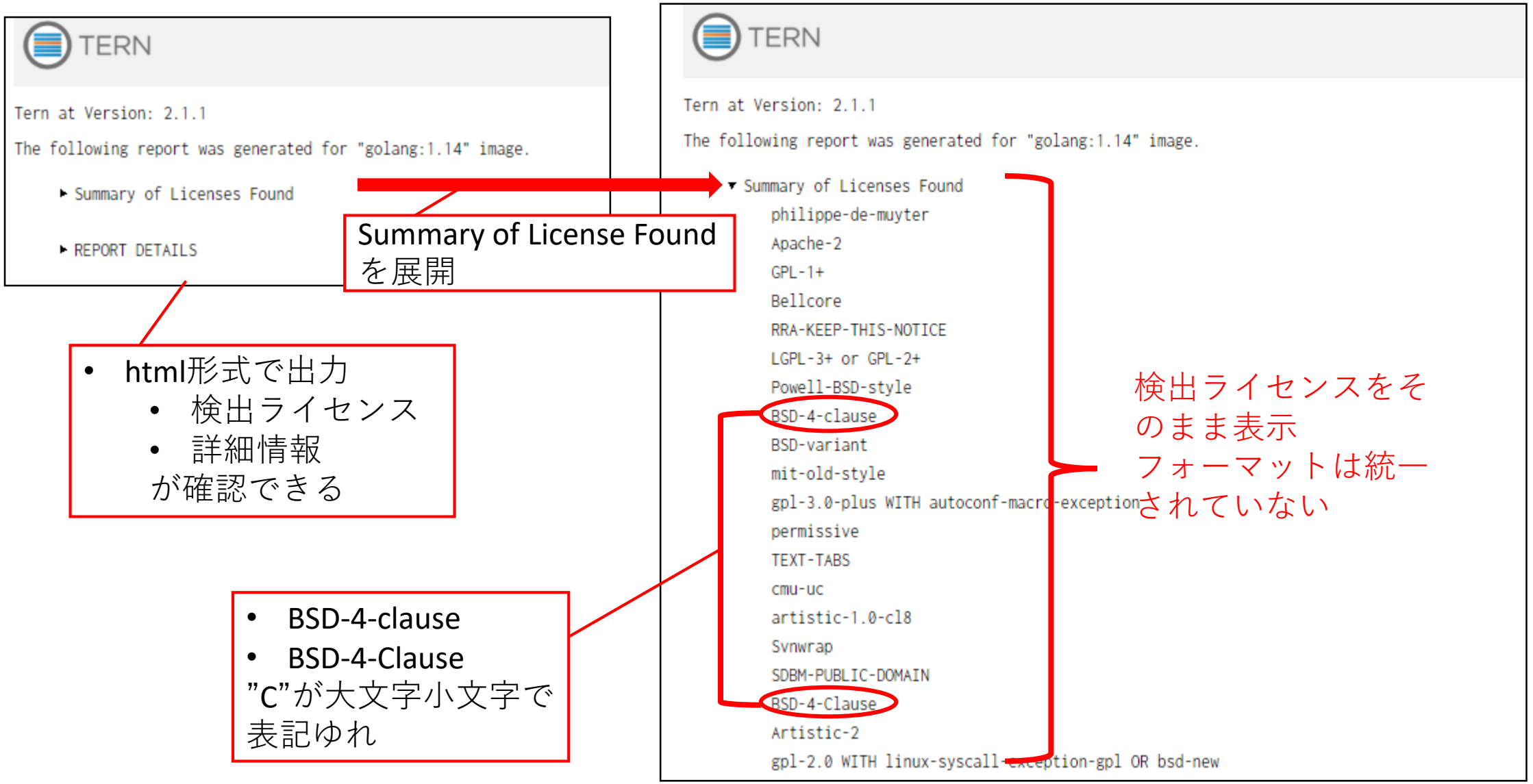
TERN実行方法（拡張機能の利用）

- Ternがインストールされているのと同じ環境でscancode, cve-bin-toolをインストール
  - $ pip3 install　scancode-toolkit cve-bin-tool
  - オプション　–x でツールを選択して実行(以下実行例)
    $ tern report -x scancode -i golang:1.12-alpine -o scancode-tern.txt
    $ tern report -x cve_bin_tool -i golang:1.12-alpine -o cve-bin-tool-out-put.txt
  - 数時間かかることがある

04

TERN出力結果

TERN出力結果



TERN出力結果詳細

TERN

Tern at Version: 2.1.1

The following report was generated for "golang:1.14" image.

▸ Summary of Licenses Found

▾ REPORT DETAILS

▾ images : [1]

▾ golang :

repotag : golang:1.14

name : golang

tag : 1.14

▾ manifest : [1]

▾ 00d970a31e :

Config : 00d970a31ef2a9b2fd2c49beb416fdb6a3590b64c7b9283b21dabeb50a57c267.json

▸ RepoTags : [1]

▸ Layers : [7]

▸ config :

▸ layers : [7]

checksum_type : sha256

checksum : d31a307a7e42116adb00d8d70971dbf228460904dd9b6217e911d088aa4b650c

▸ checksums : [0]

▸ origins : [1]

▸ repotags : [1]

▸ history : [13]

▾ 00d970a31e :

Config : 00d970a31ef2a9b2fd2c49beb416fdb6a3590b64c7b9283b21dabeb50a57c267.json

▸ RepoTags : [1]

▾ Layers : [7]

49228ffec533c170fea375471f4b760203b122c8e82e6efb53500c9bd5db20d3/layer.tar

869d77ecc1f7f81fe32deb26aca44533888858ac9dae27511984871bf7b32e96/layer.tar

f089d48df966b1b668aea852445db79ea9ec3f5ddaaf6b7dea9d5b44ae5d4cda/layer.tar

9066e467b0909baddcd5a8a4cc699b210a68798b25a63f864d6efbf3e5c2dc3a/layer.tar

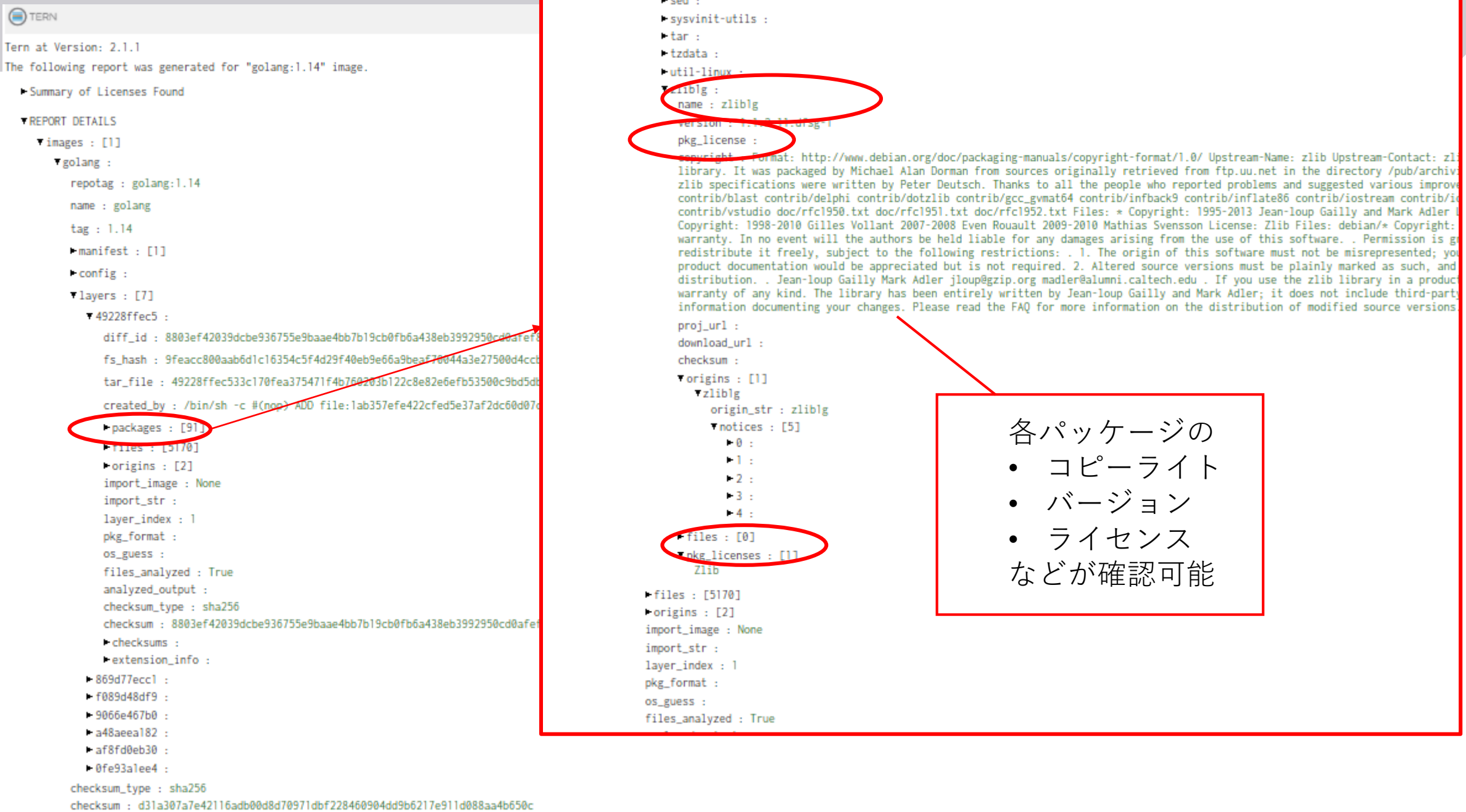
a48aeea18266836386dfa5e14573b2cb96d8c3c49eef6f6b71f369a881d37993/layer.tar

af8fd0eb30b6ba25459d58bed5328c62a4e516812814c05de1455e63bd28b11c/layer.tar

0fe93a1ee4307b83cf53b6b7f764a39516425dcd8b9257f866c3989ceb11b5e2/layer.tar

Report Detailsを展開

レイヤーごとに分析



05

さいごに　Q&A

さいごに

- TERNの所感
  - OSSの中では一番進んでいるコンテナパッケージ分析ツール
  - 商用ツールと比較しても対応パッケージが多い印象
    - 厳密に調査したわけでない
  - 開発途上な部分もある
    - レポートの表記ゆれなど
  - 現時点では, TERNを利用するだけでなく, コードをしっかり分析してイメージを配布することが必要.
- TERNについての質問ができる場所
  - Github issue
    - https://github.com/tern-tools/tern/issues
  - Vmwareのslack ternチャンネル
    - https://code.vmware.com/join
  - 他, To Do Group など

Q & A