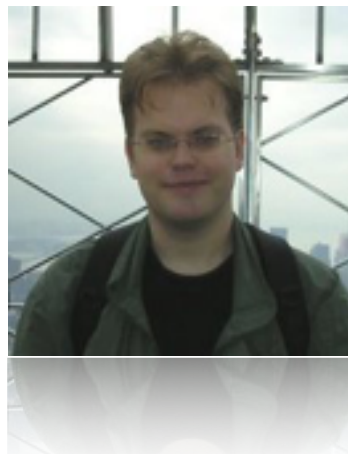


Tracing Software Build Processes to Uncover License Compliance Inconsistencies



Sander van der Burg
Delft University of
Technology



Eelco Dolstra
Logic Blox, Inc.



Shane Macintosh
Queen's University

Julius Davies
University of British
Columbia

Daniel M. German
University of Victoria

Armijn Hemel
Tjaldur Software
Governance

背景

- ❖ 開発者は、ソフトウェアを効率的に開発するため、OSSコンポーネントを再利用している
- ❖ ただし、再利用するOSSコンポーネントがGPLやLGPLなど制約が厳しいライセンスであった場合、ソースコードを公開する義務が発生する
- ❖ 商用のライセンス（ソースコードを公開しないライセンス）のソフトウェアの一部としてOSSを再利用する場合、ライセンス不整合が発生する可能性がある

GPL, LGPLのソースコード公開

✧ GPLのコピーレフト条項

- GPLのソフトウェアを再利用し作成されたソフトウェア（2次著作物）はGPLでなければならない
- GPLでなければならない→ソースコード公開

✧ GPL, LGPLの2次著作物とは

- ★ GPLのソフトウェアがディレクトリ内に存在するソフトウェア
- ★ GPLのソフトウェアとビルド時にリンクするソフトウェア
 - ・ 静的リンク・・・バイナリを作成する時にリンクする（GPL,LGPL）
 - ・ 動的リンク・・・プログラムを実行する時にリンクする（GPLのみ）

ライセンスの不整合を調べるには、ビルドプロセスを明らかにする必要がある

問題点

- ✧ ビルド時の詳細が分りにくいためライセンスアセスメントを行いにくい
 - 使われていないソースファイルが存在する
 - 大量のライブラリとリンクする
 - リンクまでを考慮したライセンスアセスメントを行える手法やツールがない

ライセンスアセスメントのための
ビルドをトレースするツールが必要!!

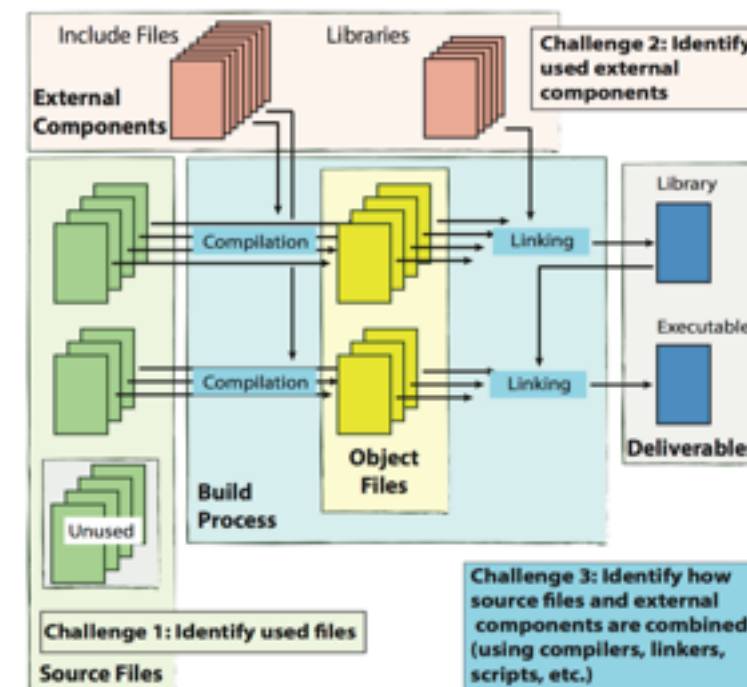
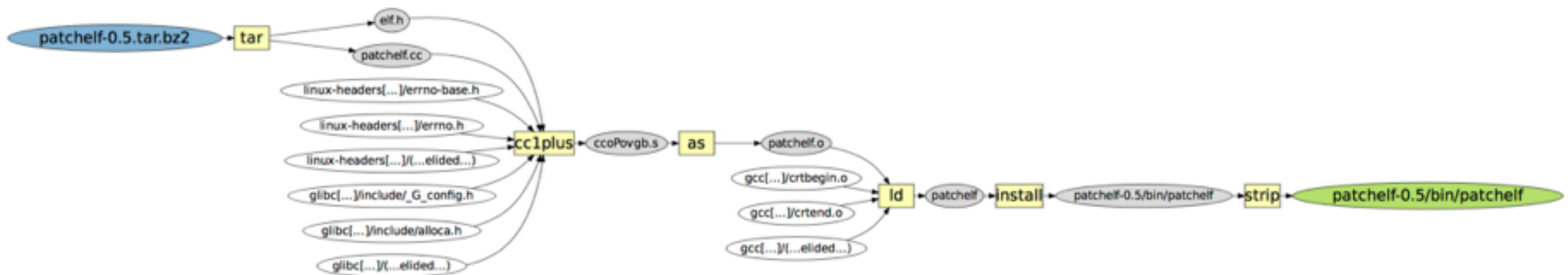


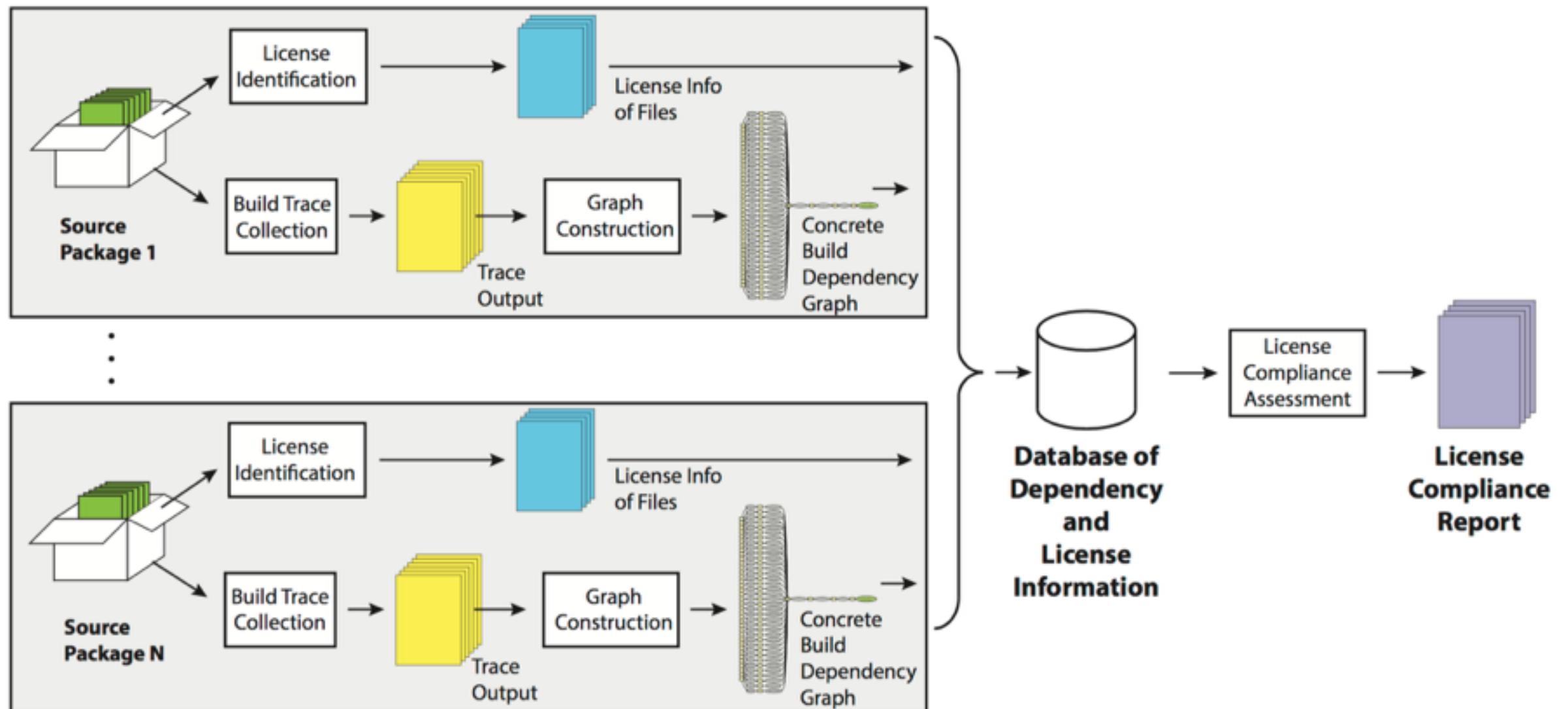
Figure 1: Challenges of license compliance assessment.

目的

- ❖ 目的：ライセンスアセスメントを行うためのビルドトレース手法を提案する
- ❖ アプローチ
 - ビルドをステップ単位で追跡し，Concrete Build Dependency Graph(CBDG)を生成する



提案手法の概要



提案手法(1/2)

1. License Identification

- ソフトウェアに含まれるソースファイルのライセンスを Ninkaを用いて特定

2. Build Trace Collection

- ビルド時に、読み書きしたファイルをステップ単位で記録していく

- straceコマンドを利用

- ★ strace -f make install

```
open("patchelf.o", O_RDWR|O_CREAT|O_TRUNC, 0666)
```

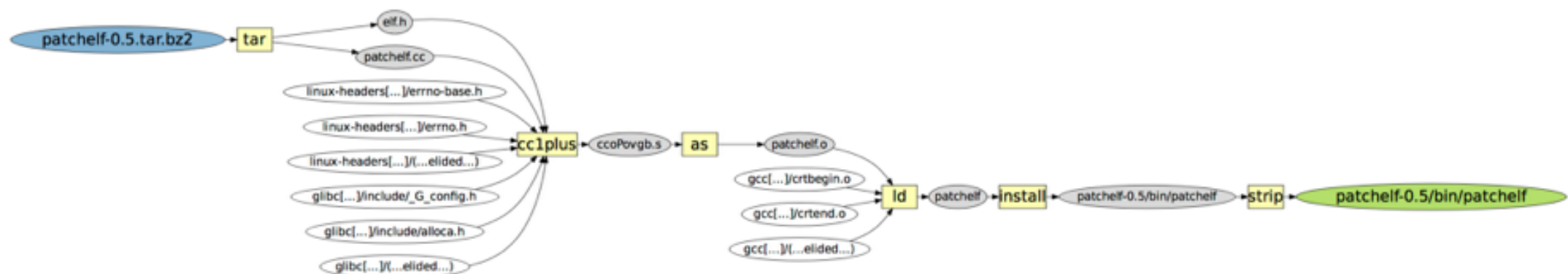
Furthermore, because g++ must read the patchelf.cc source file and all of the headers that it includes, the trace will include system calls, such as:

```
open("patchelf.cc", O_RDONLY)
open("/usr/include/c++/4.5.1/string",
      O_RDONLY|O_NOCTTY)
open("elf.h", O_RDONLY)
```

提案手法(2/2)

❖ 3.Graph Construction

- トレース結果から, CBDGを生成する
 - ★ ビルドの各ステップ(Vt)と, そのステップで読み書きしたファイル(Vf)の情報を抽出する
 - ★ Vt . . . <tid,program,args>
 - ★ Vf . . . <fid,path,license> *既出の場合作成しない



ケーススタディ

- ✧ 提案手法の有用性を検証するため、ケーススタディを行う
 - RQ1:ビルドで使用するファイルを正確に特定できているか
 - RQ2:ライセンス不整合を確認できるか
- ✧ データセットとなるソフトウェア

Table 1: An overview of the studied systems.

Package	Version	Domain	Size (SLOC) [†]	Files
Aterm	2.5	Program transformation	21.5k	133
Opkg	0.1.8	Software package management	22.2k	132
Bash	4.1	Command shell	115.7k	1,114
CUPS	1.4.6	Printer management	142.8k	1,398
Xalan	2.7.1	XML processing	176.0k	1,334
OpenSSL	1.0.0d	Digital communication security	308.0k	2,105
FFmpeg	0.7-rc1	Multimedia processing	442.2k	2,267

[†] Non-comment, non-whitespace lines of code calculated using SLOCCount (<http://www.dwheeler.com/sloccount/>).

RQ1の検証

✧ 検証方法

- ディレクトリ内のファイルを1つずつ削除してビルドを繰り返して正解集合を作成
- 提案手法で抽出された使用ファイルを突きあわせ, その精度を評価する

✧ 検証結果

- **高い適合率(98%-100%), 再現率(88%-100%)をマークした**

Table 2: Results of RQ1: CBDGs provide very high recall and precision.

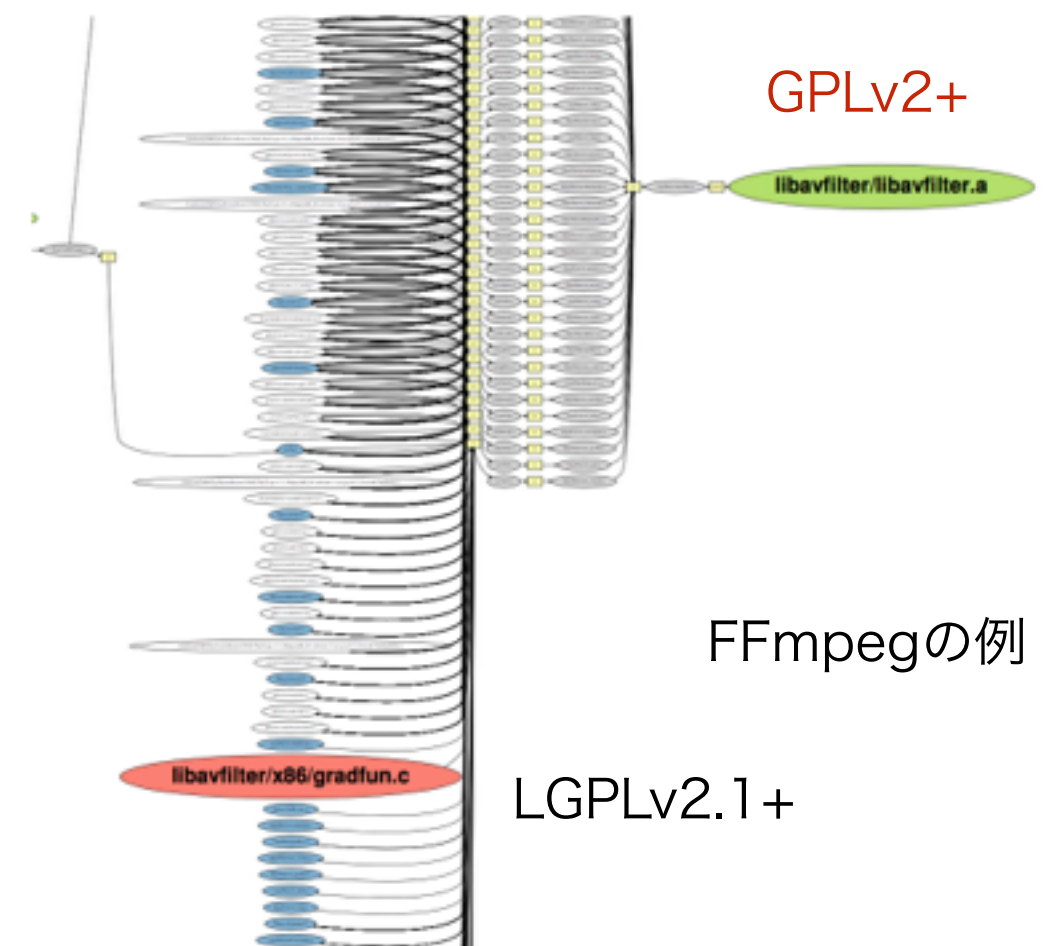
Package	Files					False Negatives	False Positives	Recall	Precision
	All	Source files	Excluded	Included	% Included				
Aterm	133	117	60	57	49%	1	0	98%	100%
Opkg	132	106	9	97	92%	1	3	99%	97%
Bash	1,111	1,086	806	280	26%	0	34	100%	88%
CUPS	1,398	1,079	213	866	80%	12	0	99%	100%
Xalan	1,334	1,334	379	955	72%	0	4	100%	99%
OpenSSL	2,105	2,027	1,180	847	42%	0	5	100%	99%
FFmpeg	2,267	2,239	986	1,253	56%	0	7	100%	99%

RQ2の検証

- ＊ 検証：CGBDにおけるコンポーネント関係から、ライセンス不整合が発生しているところを見つける
- ＊ 検証結果：2つのプロジェクトから不整合を見つけることができた

FFmpeg – The license of the FFmpeg deliverables varies depending on the configuration flags that are specified prior to executing the build. By default, its deliverables are released under the LGPLv2+ license. However, when configured using the appropriate flags, the deliverables are released under the GPLv2+ license.

CUPS – The CUPS deliverables are licensed under the permissive CUPS license. The studied version (1.4.6) contains three files (`backend/*scsi*`) that are released under the BSD-4 license, which is not compatible with the CUPS License. It is important to know whether any of these three files are being combined with CUPS licensed files during the build process.



まとめ

- ✧ ライセンスアセスメントを行うためのビルドトレース手法を提案した
- ✧ ケーススタディの結果
 - 98%-100%の適合率で、使用されるファイルを特定できた
 - FFmpeg, CUPSから直すべきライセンス不整合が発見された
- ✧ 今後の課題
 - 精度を再検証
 - ★ ビルドシステムの誤りについては考慮していない

所感

❖ いいところ

- やるべき研究, ストーリがシンプル
- OpenSSLのファイルは半分以上コンパイルされていない
 - ★ ダイナミックリンクが多い可能性あり

❖ 悪いところ

- RQ1の評価方法が不完全
- アイデア部分がないのでおもしろさに欠ける(考察もない)
- 動的リンクと静的リンク分けて分析して欲しかった