Binary Diff 연구 및 개발

캡스톤디자인 2022 36팀

[중간 발표]

발표자: 윤형준

2022-04-08

목차

(1) 프로젝트 소개
(2) 진행 상황
(3) 향후 계획

프로젝트 소개

(1) 프로젝트 소개

프로젝트 팀 정보

- 산학분반 자유주제 36팀
- 기관: 한국과학기술원 사이버보안연구센터
- 국가 연구과제의 일환인 “바이너리 분석 플랫폼 B2R2 구현”에 참여하여 Bianry Diff 파트 단독 담당
- 기관 특성상 개발 및 서비스 제공보다는 **학술적 연구와 그에 따른 산출물**에 초점

프로젝트 팀 멤버

- 윤형준 학부생 (팀장)
  - 논문 조사 및 발표, 연구, 개발, 프로젝트 일정 및 산출물 관리
- 차상길 교수님 (멘토)
  - 논문 추천, 연구 방향성 제시, 코드 리뷰, 매주 온라인 미팅 및 피드백

(1) 프로젝트 소개

주제: Binary Diff 연구 및 개발

- Binary Diff: binary executable file(이하 binary)들 간의 차이를 확인하는 행위나 도구.

Binary

- 컴파일된 실행 가능한 프로그램 (CODE 영역 + DATA 영역)
  - 보안 전문가, 해커 등의 실무의 분석가는 분석 대상의 소스코드가 없을 확률이 매우 높다.
  - 심지어 소스코드를 확보하더라도 바이너리만이 가진 정보를 위해 바이너리 분석은 필수
  - 따라서 분석 대상은 소스코드가 아닌 바이너리 [1][2]

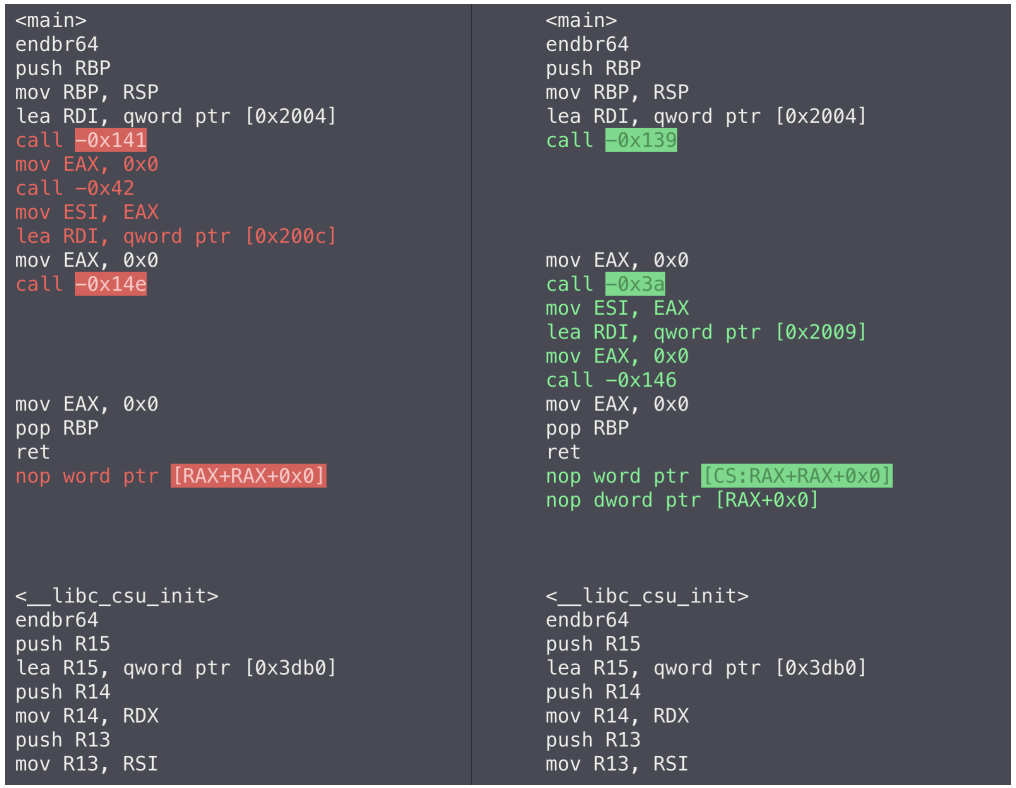
Diff

- 두 파일 간의 차이를 확인하는 행위나 이를 위한 도구
- 변경된 정보를 효율적이고 효과적으로 제공하는 데에 초점

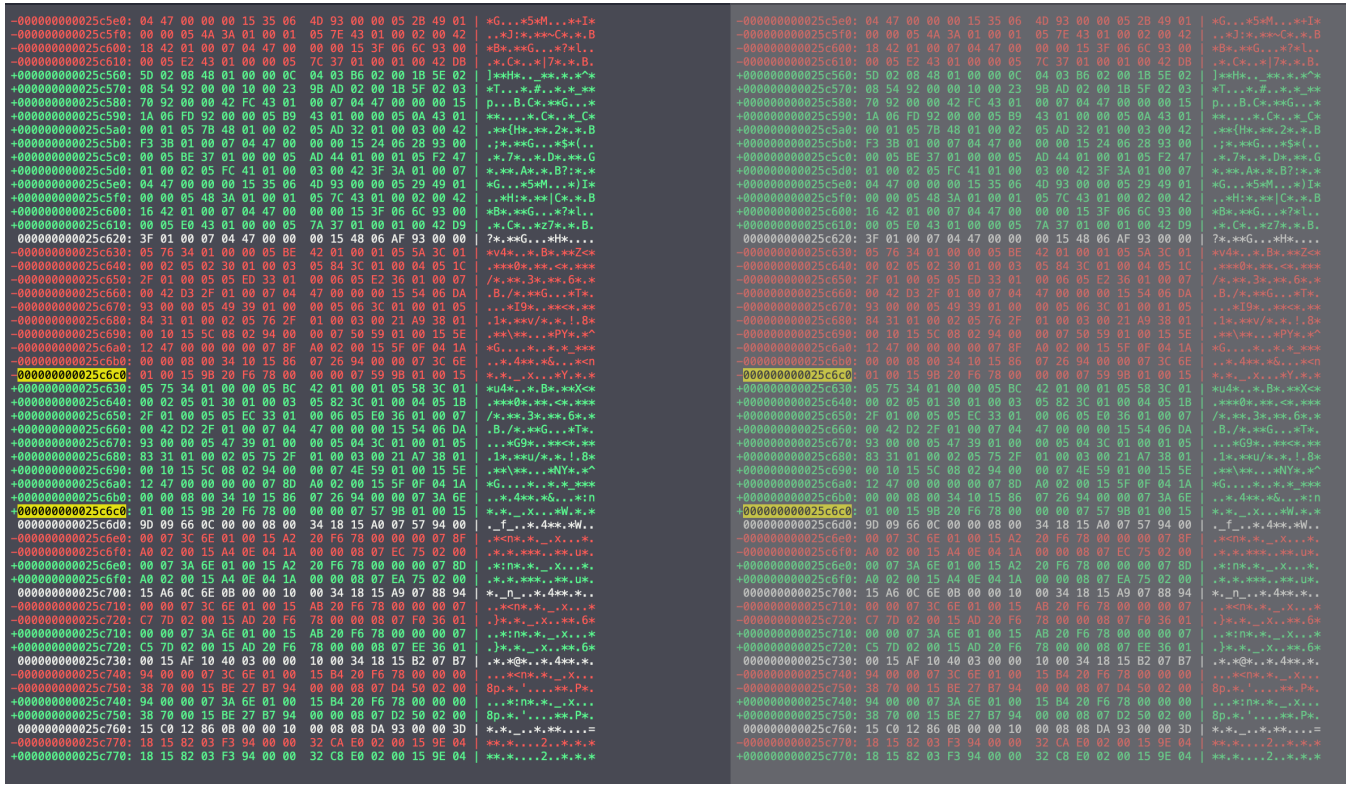
(1) 프로젝트 소개

Binary Diff

- 바이너리를 Section 별로 분석
- CODE 영역은 disassemble하여 assembly 언어로 표현하고, DATA 영역은 Hex 값으로 표현 가능



[CODE 영역 diff 예시 사진]



[DATA 영역 diff 예시 사진]

(1) 프로젝트 소개

Binary Diff 활용 분야

- Patch 생성 및 분석
- Malware detection 및 clustering(분류)
- 프로그램 버전 간 정보 porting
- 소프트웨어 도난 감지(소스코드 도난, 바이너리 코드 재사용, 라이센스 위반 등)
- 알려진 버그 또는 취약점 탐색 [3]

Binary Diff를 필요로 하는 사람

- 정보보안 전문가
- 화이트/블랙 햇 해커
- 개발자
- 해킹 대회(CTF) 참가자
- 등등 바이너리 분석을 필요로 하는 **실무의 모든 분석가**들

(1) 프로젝트 소개

프로젝트 Scope

Binary Diff 연구 및 구현 (기존 프로젝트 계획)

- Binary Diff 알고리즘을 구현하여 바이너리 분석 플랫폼인 B2R2에 탑재
- 널리 알려진 Text Diff 알고리즘을 이용해 Binary를 Diffing했을 때 나타는 통계적 양상을 연구
  - Git diff나 Unix diff 등에서 널리 사용되는 text diff 알고리즘 4가지 [4]
    - Minimal, Myers, Patience, Histogram

Binary Diff Dataset (프로젝트 진행 중 추가된 부분)

- Binary Diff를 평가할 수 있는 Dataset(Binary Pair들) 확보
  - 서로 다른 Binary Diff 알고리즘들 평가할 만한 공통적이고 양질의 Dataset이 없다는 문제점 존재
  - 특정 Git Repo의 모든 commit에 대해, commit별 binary pair들을 수집하는 도구 구현

(1) 프로젝트 소개

예상 산출물 및 기대효과

Binary Diff 연구 및 구현

- 국가 연구과제인 바이너리 분석 플랫폼 B2R2에 Binary Diff 기능 추가
  - 바이너리 분석 플랫폼 이용자 및 Binary Diff 도구 이용자가 사용 가능
  - 개발, 난독화 연구, 악성코드 연구, 패치 분석, 취약점 분석 등
- Text Diff 알고리즘을 Binary Diffing에 활용한 연구 결과
  - 특히 보안 관련 패치에 대한 Binary Diff 연구 결과

Binary Diff Dataset

- Binary Diff를 위한 Dataset 및 Dataset 확보를 위한 도구
  - Bianry Diff Dataset이 없다는 학계의 문제 해결
  - 다른 Bainry Diff 연구자가 연구에 사용거나 머신 러닝 연구자가 Dataset으로 사용 가능

진행 상황

(2) 진행 상황

Binary Diff 연구 및 구현

- Git의 네 가지 Diff 알고리즘 분석
  - **Minimal**: LCS와 SES Problem을 해결하는 알고리즘. 1986년 Myers의 Tech Paper에서 소개 [5]
  - **Myers**: Minimal 알고리즘 + Heuristic. Git Diff 소스코드 분석 [6]
  - **Patience**: 순수한 LCS가 아닌 Unique Lines를 매칭하는 알고리즘. Bram Cohen의 게시글 [7]
  - **Histogram**: Patience 알고리즘 계승. Git Diff 소스코드 분석 [8]
- 두 가지 Diff 알고리즘(Minimal, Histogram) 구현
- 바이너리 분석 플랫폼 B2R2에 탑재
  - B2R2 플랫폼은 .NET 프레임워크 기반
  - 함수형 프로그래밍 언어 F# Language로 구현

(2) 진행 상황

Binary Diff 연구 및 구현

- Diff 결과를 나란히(side-by-side) 출력
- line by line diff 뿐 아니라 binary component 정보를 이용한 **finer-granularity diff**도 가능

```
<main>
0000000000001192: endbr64
0000000000001196: push RBP
0000000000001197: mov RBP, RSP
000000000000119a: lea RDI, qword ptr [0x2004]
00000000000011a1: call -0x141
00000000000011a6: mov EAX, 0x0
00000000000011ab: call -0x42
00000000000011b0: mov ESI, EAX
00000000000011b2: lea RDI, qword ptr [0x200c]
00000000000011b9: mov EAX, 0x0
00000000000011be: call -0x14e
00000000000011c3: mov EAX, 0x0
00000000000011c8: pop RBP
00000000000011c9: ret
00000000000011ca: nop word ptr [RAX+RAX+0x0]
```

```
(.rodata)
01 00 02 00 41 41 41 41  42 42 42 00 25 64 0A 00 | *.*.AAAABBB.%d_.

(.eh_frame_hdr)
01 1B 03 3B 4C 00 00 00  08 00 00 00 10 F0 FF FF | ***;L...*...*...
80 00 00 00 40 F0 FF FF  A8 00 00 00 50 F0 FF FF | ....@.......P...
C0 00 00 00 70 F0 FF FF  68 00 00 00 59 F1 FF FF | ....p...h...Y...
D8 00 00 00 82 F1 FF FF  F8 00 00 00 C0 F1 FF FF | ................
18 01 00 00 30 F2 FF FF  60 01 00 00              | **..0...`*..
```

```
<main>
000000000000118a: endbr64
000000000000118e: push RBP
000000000000118f: mov RBP, RSP
0000000000001192: lea RDI, qword ptr [0x2004]
0000000000001199: call -0x139
000000000000119e: mov EAX, 0x0
00000000000011a3: call -0x3a
00000000000011a8: mov ESI, EAX
00000000000011aa: lea RDI, qword ptr [0x2009]
00000000000011b1: mov EAX, 0x0
00000000000011b6: call -0x146
00000000000011bb: mov EAX, 0x0
00000000000011c0: pop RBP
00000000000011c1: ret
00000000000011c2: nop word ptr [CS:RAX+RAX+0x0]
00000000000011cc: nop dword ptr [RAX+0x0]
```

```
(.rodata)
01 00 02 00 42 42 42 42  00 25 64 0A 00          | *.*.BBBB.%d_.

(.eh_frame_hdr)
01 1B 03 3B 4C 00 00 00  08 00 00 00 10 F0 FF FF | ***;L...*...*...
80 00 00 00 40 F0 FF FF  A8 00 00 00 50 F0 FF FF | ....@.......P...
C0 00 00 00 70 F0 FF FF  68 00 00 00 59 F1 FF FF | ....p...h...Y...
D8 00 00 00 7A F1 FF FF  F8 00 00 00 C0 F1 FF FF | ....z...........
18 01 00 00 30 F2 FF FF  60 01 00 00              | **..0...`*..
```

(2) 진행 상황

Binary Diff Dataset

- Binary Pair Dataset 수집을 위한 도구 구현
- GNU Binutils Repository를 대상으로 Dataset 수집 가능
  - 모든 commit에 대해, 해당 commit과 그 parent commit이 하나의 Pair.
  - GitPython Library와 shell script를 이용해 구현
  - 약 6개월 치 commit에 대한 정보들과 컴파일된 바이너리들(binary pair)를 수집

(2) 진행 상황

Binary Diff Dataset

- 한 pair를 bin-{datetime}-{commit_HASH} 디렉토리에 저장
- commit과 binary file들에 대한 정보를 각각 commit_info, files_info에 저장
- 실제 binary들을 parent와 patched 디렉토리 내부에 저장

```
› tree -C bin-20220308T223651-fb0e49d8e05e61ca2af9b5f60b01ad5fb6d274ff
bin-20220308T223651-fb0e49d8e05e61ca2af9b5f60b01ad5fb6d274ff
├── commit_info
├── files_info
├── parent
│   └── bin
│       └── as
└── patched
    └── bin
        └── as
```

향후 계획

(3) 향후 계획

Binary Diff 연구 및 구현

- Minimal 알고리즘과 Myers 알고리즘 Diff 결과 차이 확인
- Binary Diff 연구

Binary Diff Dataset

- 현재 구현은 한 commit과 그 parent commit을 pair로 정의
- Branch 단위의 Diff나 Merge case에 대한 처리 필요

프로젝트 산출물 관리

- 시나리오나 Use Case Diagram 등 개발에 대한 산출물 정리

발표 들어주셔서 감사합니다.

Reference

[1] [논문 리뷰] "WYSINWYX: What you see is not what you execute." - CPUU

[2] 바이너리 분석에 대한 오해와 진실 - CSRC

[3] A Survey of Binary Code Similarity - Arxiv

[4] How different are different diff algorithms in Git? - Springer

[5] An O(ND) difference algorithm and its variations - Springer

[6] Myers Diff Algorithm source code – Github Repository of Git

[7] Patience Diff Advantages - Bram Cohen

[8] Histogram Diff Algorithm source code – Github Repository of Git

Appendix A.