도구 개발 계획서

2024.11.07

모바일시스템공학과 조민혁

INDEX

01 Logcat 도구 개발 계획서

Logcat 도구 개발

❖Logcat의 문제점

✓ Logcat은 로그 데이터를 명령어를 통해 간편하게 볼 수 있다는 장점이 있지만,
전원 OFF 시 로그 데이터가 삭제된다는 특성이 있음

❖목적

✓ 휘발성 로그가 재부팅 또는 전원 꺼짐 감지 시 비휘발성 영역으로 옮겨져서 “안티-포렌식“ 키워드만 남기고, 삭제한다.

✓ 포렌식 수사 관점에서 개발

✓ 효율적인 포렌식 수사가 될 수 있도록 기여하는 측면에서 개발

개발 계획서

❖ Logcat 도구 개발 계획서 - 1

✓ 참여 인원

- 포렌식 팀: 조민혁, 이승민

- 머신러닝 팀: 정성원

✓ 개요

-> logcat은 시스템 전원 OFF 또는 재부팅하면 기존 로그가 사라지는 특성이 있다.

-> 따라서, 사전에 정의된 특정 안티 포렌식 행위가 감지되면 로그를 따로 옮겨 저장할 필요가 있다.

개발 계획서

❖ Logcat 도구 개발 계획서 - 2

✓ 대상 OS

-> Android 14, Galaxy 21

✓ 작동 방식

-> 도구니까 백그라운드 작동

-> UI 없음

개발 계획서

❖ Logcat 도구 개발 계획서 - 3

✓ 로직

STEP 1) 안티 포렌식 행위 감지

안티 포렌식 행위 종류 정리 및 해당 로그 메시지 확인 필요

(안티 포렌식 행위 종류는 인터넷으로 정리 가능, 로그 메시지는 logcat 실험 해보면서 확인 필요)

해당 안티 포렌식 행위 발생할 때 내부적으로 호출되는 함수 종류 찾기 필요

STEP 2) 내부적으로 logcat 작동 후 '포렌식 로그' 비휘발성 영역으로 옮긴다.

-> 초기 버전에서는 휴대폰 내 내부 저장소로 옮김

-> 비휘발성 영역 내부 저장소 종류 파악

STEP 3) 추후 조회 시 해당 로그 데이터를 보면서 안티 포렌식 행위 확신 가능

-> 초기 버전 제작된 후 교수님께서 말씀하신 것 + 교수님 피드백 얻으면서 추가하면서 확장

-> EX) NAS 서버로 옮기기 …

개발 계획서

❖ Logcat 도구 개발 계획서 - 4

✓ **목표 기간** -> 12월 전에는 끝내고 싶음

1) 학기 중에는 바쁘니까 .. 연구실 시간에 할 일 끝내는 게 목표

-> 연구실 시간(10:00 ~ 17:00)에 하기

2) 일단은 초기 버전 만드는 거 목표 -> 루팅되지 않은 휴대폰에서 작동

- 11월 4일 ~ 8일 : 관련 자료 찾기

(안티 포렌식 행위 종류, 해당 로그 메시지, 안티 포렌식 행위 발생 시 호출되는 함수 종류, 비휘발성 내부 저장소 종류 파악)

- 11월 11일 ~ 15일 : 구현 설계 (Android Studio)

- 11월 18일 ~ 29일 : 구현 및 테스트 (빠르게 구현할 수록 좋음)

02 관련 자료 찾기

관련 자료 찾기

❖ 11월4일 ~ 11월 8일 (관련 자료 찾기)

 # 안티 포렌식 행위 및 종류 -> {데이터 파괴, 데이터 암호화, 데이터 조작, 분석 시간 증가}

- 모바일 기기(Android)에서 가능한 안티 포렌식 행위 위주
- 로그 남는 거 위주

① Timestamp 조작
② Logcat –c를 통한 로그 버퍼 비우기
③ 전원 끄기 및 시스템 재부팅
④ 공장 초기화
⑤ 로그 기록 비활성화

-> 해당 부분 logcat에서 로그 어떻게 남는지 확인 필요

Timestamp 조작

❖ Timestamp 조작

- 날짜 및 시간 조작 방법 : '설정->일반->날짜 및 시간->조작'

- 이후 생성되는 로그 키워드

① timeChanged

② set system clock

③ Time is changed

④ TIME_CHANGED

⑤ onTimeChanged()

⑥ ACTION_TIME_CHANGED

Logcat –c를 통한 로그 버퍼 비우기

❖ Logcat –c를 통한 로그 버퍼 비우기

-> 링버퍼 (main, system, crash, kernel)를 비우는 명령어

-> 따로 로그 남지 않음

-> 비이상적 패턴으로 감지 필요

```
PS C:\Users\cgumg> adb logcat -g
main: ring buffer is 5 MiB (1 MiB consumed, 1 MiB readable), max entry is 5120 B, max payload is 4068 B
system: ring buffer is 2 MiB (512 KiB consumed, 85 KiB readable), max entry is 5120 B, max payload is 4068 B
crash: ring buffer is 512 KiB (0 B consumed, 0 B readable), max entry is 5120 B, max payload is 4068 B
kernel: ring buffer is 4 MiB (0 B consumed, 0 B readable), max entry is 5120 B, max payload is 4068 B
```

logcat –c 실행 전

```
PS C:\Users\cgumg> adb logcat -g
main: ring buffer is 5 MiB (1 MiB consumed, 2 KiB readable), max entry is 5120 B, max payload is 4068 B
system: ring buffer is 2 MiB (512 KiB consumed, 289 B readable), max entry is 5120 B, max payload is 4068 B
crash: ring buffer is 512 KiB (0 B consumed, 0 B readable), max entry is 5120 B, max payload is 4068 B
kernel: ring buffer is 4 MiB (0 B consumed, 0 B readable), max entry is 5120 B, max payload is 4068 B
```

logcat –c 실행 후

-> 버퍼가 꽉 차지 않았는 데 버퍼가 비워지는 것으로 안티 포렌식 행위 탐지 가능

-> 단, 어떻게 효율적으로 모니터링 해야할 지는 고민 필요 ..

감사합니다