

안드로이드폰에서 타임스탬프 변경 이벤트의 실시간 탐지

Authors : 안균승, 김선재, 조성제

Presenter : 안균승

Affiliation : 단국대학교, 컴퓨터보안 및 OS 연구실

Email : staran1227@dankook.ac.kr

INDEX

01

서론

02

배경 지식 및 관련 연구

03

탐지 방법

04

구현, 실험 및 평가

05

결론 및 향후 연구



01

서론

서론 – 안티 포렌식

❖ 안티 포렌식(Anti-forensics) 정의

- 디지털포렌식 수사를 방해하기 위해 증거를 은닉, 변조, 파괴하는 기법 전반을 의미함
- **타임스탬프 변경**(timestamp tampering)는 파일이나 시스템 리소스의 생성, 수정, 접근 시간등의 메타데이터를 조작하여 실제 이벤트 발생 시점을 감추는, **안티 포렌식 기법**임

❖ 타임스탬프 변경 예시

파일 형식:	응용 프로그램 확장(.dll)
연결 프로그램:	알 수 없는 응용 프로그램 변경(C)...
위치:	N:\Windows\System32
크기:	2.54MB (2,671,616 바이트)
디스크 할당 크기:	2.55MB (2,674,688 바이트)
만든 날짜:	2015년 12월 14일 월요일, 오후 6:03:08
수정한 날짜:	2015년 7월 10일 금요일, 오전 2:13:49
액세스한 날짜:	2015년 12월 14일 월요일, 오후 6:03:08

라자루스 그룹이 사용한 타임스탬프 변경 예시

서론 – 포렌식 수사에서의 문제점 및 연구 목적

❖ 기존 대부분의 포렌식 수사는 사건이 발생한 사후에 진행 됨

- 타임스탬프 변조 행위가 발생한 이후에 조사가 시작 됨
- 용의자는 사건과 관련된 로그 파일, 메타데이터 등을 조작/삭제 가능

❖ 타임스탬프 변조로 인한 문제점

- 디지털 포렌식 분석가의 분석 시간 증가
- 범죄 이벤트 은닉 및 정보 수집 방해

❖ 연구 목적

- 이벤트 발생 이후에 포렌식 조사가 진행되는 경우의 문제들을 극복하고자 **타임스탬프 변조 이벤트를 실시간**으로 탐지

연구 대상: 안드로이드 스마트폰

❖안드로이드 스마트폰

- 국내 스마트폰 보급률은 99%, 그 중 안드로이드 스마트폰이 72% 사용
- 20, 30 세대의 55.6%는 스마트폰을 4시간 이상 사용

❖안드로이드 스마트폰이 일상생활에서 다양한 정보들을 많이 저장하므로 안드로이드 스마트폰을 대상으로 연구 수행

머니온 : 여론조사·통계

[여론조사on] 스마트폰 안 쓰는 성인, 이제 없다...보급률 99% 시대

이장원 기자

입력 2025.07.07 23:21 | 수정 2025.07.08 09:59

스마트폰 사용자 986명에게 현재 주로 사용하는 브랜드를 물었다(자유응답). 그 결과 삼성이 72%, 애플 24%, 이외 브랜드 2% 순으로 나타났다. 2%는 현재 사용 중인 스마트폰 브랜드를 정확히 모르거나 답하지 않았다. 삼성, 애플이 작년 대비 각각 3%p, 1%p 늘었다.

스마트폰 보급률과 국내 안드로이드 폰 점유율

하루에 스마트폰을 몇 시간 정도 사용하시나요?

※ 2024년 8월 9일부터 15일까지 메타데이터 구축자를 대상으로 한 온라인 설문조사 결과, 388명 대답



스마트폰 사용 시간 통계



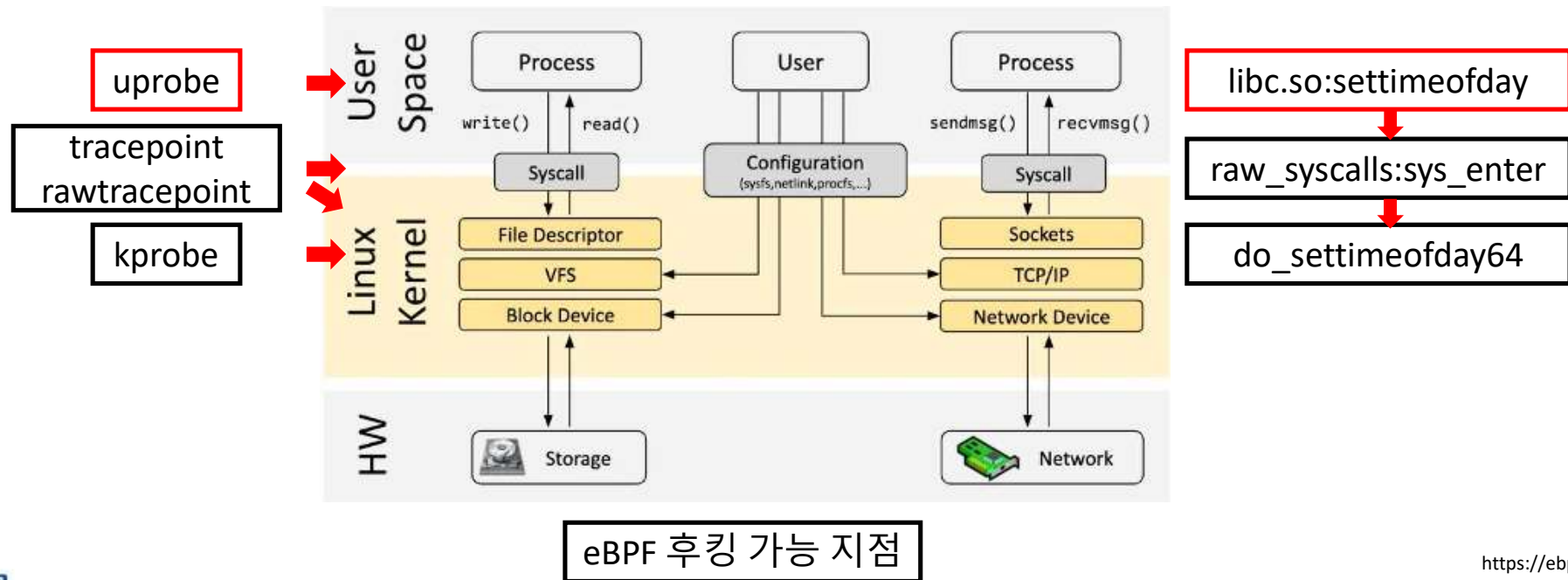
02

배경 지식 및 관련 연구

배경지식: eBPF

❖ eBPF

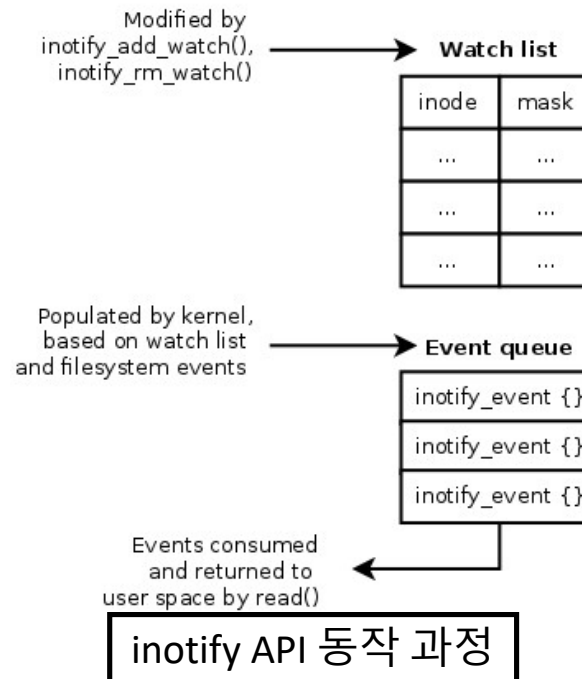
- 리눅스 커널 내부에서 안전하게 사용자 정의 코드를 실행할 수 있게 해 주는 기술
- uprobe, tracepoint, rawtracepoint, kprobe 지점을 후킹 가능
- 사용자가 라이브러리 수준에서 시간을 조작하더라도, **커널(or 시스템 콜 수준)에서 타임 관련 이벤트를 관찰·로깅하여 시스템 시간 변조여부를 탐지할 수 있음**



배경지식: inotify API

❖ inotify API

- 파일 시스템에서 시스템 콜을 통해 **특정 파일이나 디렉터리의 생성, 삭제, 메타데이터(권한·소유자·타임스탬프) 변경, 내용 쓰기** 등의 이벤트를 실시간으로 감지
- Inotify_init(), inotify_add_watch(), inotify_rm_watch()을 통해 감시 파일 등록, 삭제 및 감시 이벤트 종류(mask) 설정
- 즉, 특정 **파일·디렉터리**에 대한 **메타데이터 변경**(예, 타임스탬프 변경)을 실시간으로 모니터링 가능



관련 연구

- ❖ Palmbach 등[3]은 NTFS 파일시스템의 여러 아티팩트 로그 파일들을 활용하여 타임스탬프 조작을 탐지하는 방법을 제안함
- ❖ Oh 등[4]은 NTFS 파일시스템에서 다양한 로그·메타데이터 아티팩트를 통합 분석하여 타임스탬프 조작을 탐지하는 알고리즘을 제안함
- 👉 위 연구들은 NTFS 파일 시스템에서의 로그 파일 기반으로 타임스탬프 조작을 탐지하는 방법임.
 - 이들 연구들의 한계점: MS Windows 로그 파일의 무결성이 보장된다는 전제, 사후에 해당 이벤트를 탐지
- ❖ 본 논문은 안드로이드 시스템에서의 타임스탬프 조작을 실시간 탐지함

[3] Palmbach, David, and Frank Breiting. "Artifacts for detecting timestamp manipulation in NTFS on windows and their reliability." Forensic Science International: Digital Investigation, Vol. 32, 300920, 2020.

[4] Oh, Junghoon, Sangjin Lee, and Hyunuk Hwang. "Forensic detection of timestamp manipulation for digital forensic investigation." IEEE Access 12, pp.72544-72565, 2024.

관련 연구

- ❖ 이산 등[5]은 리눅스·안드로이드 환경에서 여러 시스템 로그를 활용해 과거 및 미래 시점의 시스템 시간 조작을 탐지하는 방법을 제시함
 - 반면 본 논문은 실시간으로 시스템 시간 뿐 아니라 **파일**에 대한 **시간 메타데이터** 변조도 탐지
- ❖ 안균승 등[6]은 eBPF를 활용하여 settimeofday 시스템 콜을 모니터링하여, 시스템 날짜 및 시간 설정, date, toybox date, hwclock 명령어를 통한 날짜 및 시간 변경 행위를 탐지함
 - 본 논문은 [6] 연구에 다음 기법을 추가 구현함
 - 추가적인 시스템 시간 변경 방법(네트워크 시간대로 설정, 지역 시간대로 설정) 고려
 - 파일의 시간 메타데이터 조작에 대한 탐지를 실시함

[5] 이산, 조민혁, 정지현, 조성제, “리눅스와 안드로이드 시스템에서 타임스탬프 조작식별을 위한 로그 분석 기법”, 한국차세대 컴퓨팅학회 논문지, 제20권, 제4호, pp. 34–45, 2024.

[6] 안균승, 김선재, 정연수, 김보겸, 조성제, “안드로이드에서 eBPF를 이용한 타임스탬프 변경 이벤트 감지”, 차세대 컴퓨팅 학회 하계 워크숍, 제주도, 2025, pp. 1-4



03

탐지 방법

탐지방법 - 개요

❖ 타임스탬프 변경 이벤트를 시스템 시간 변경과 파일 시간 메타데이터 변경 이벤트로 분류

1) 시스템 시간 변경

- 시스템 수준에서 설정 메뉴 또는 명령을 통한 시간 변경
- 탐지방법: `settimeofday` 시스템 콜 후킹을 통한 감시

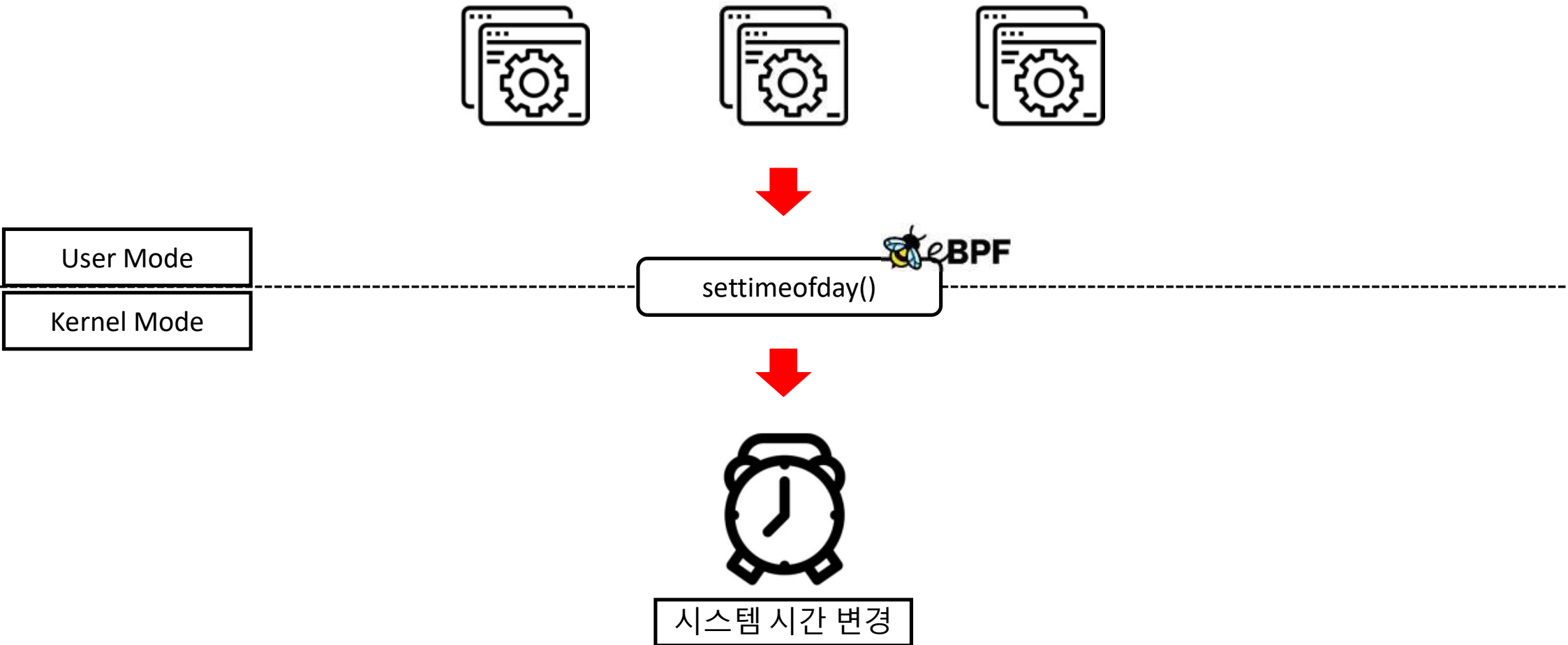


2) 파일 타임스탬프 관련 메타데이터 변경

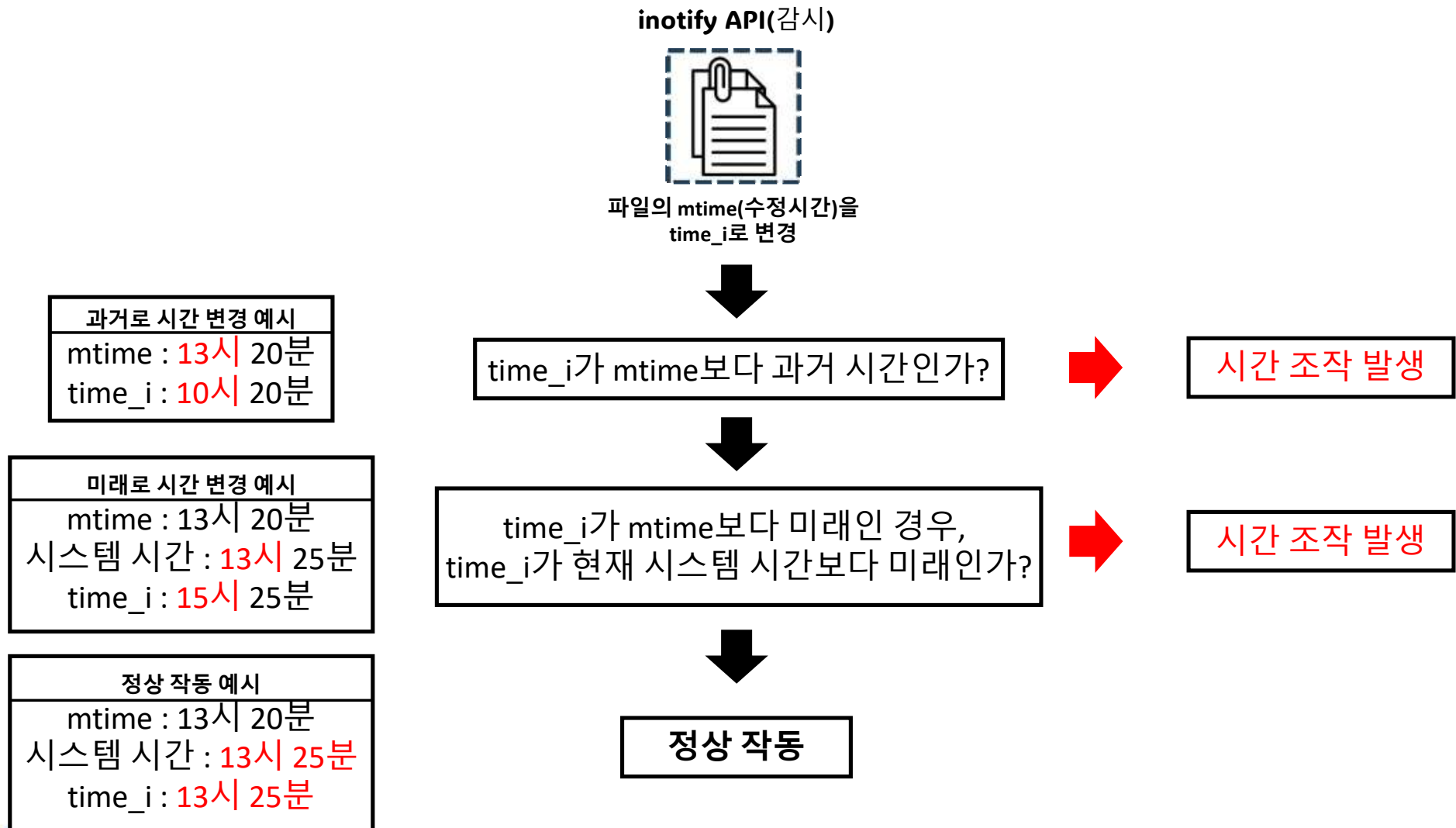
- 파일에 대한 `mtime`(파일의 내용이 마지막으로 수정된 시간) 변경
- 탐지방법: `inotify API`를 통한 감시와 `stat()`을 이용한 파일의 메타데이터 획득 후 비교



탐지방법 - 시스템 시간 변경 탐지



탐지방법 - 파일의 시간 메타데이터 변경 탐지



04

구현, 실험 및 평가

개발 환경



CI Failing IRC bpfftrace CodeQL Failing

bpfftrace는 Linux용 고급 추적 언어입니다. bpfftrace는 LLVM을 백엔드로 사용하여 **스크립트를 eBPF 바이트코드로 컴파일하고** libbpf와 bcc를 사용하여 Linux BPF 하위 시스템과 기존 Linux 추적 기능(커널 동적 추적(kprobes), 사용자 수준 동적 추적(uprobes), 추적점 등)과 상호 작용합니다. bpfftrace 언어는 awk, C 및 DTrace와 SystemTap과 같은 이전 추적 프로그램에서 영감을 받았습니다. bpfftrace는 Alastair Robertson이 만들었습니다.

확장된 안드로이드 도구(Extended Android Tools)는 우리 모두가 사랑하는 **안드로이드용 리눅스 도구들을 크로스 컴파일하는** 메이크파일과 빌드 환경 세트입니다. 모든 도구는 기본 빌드 시스템(autotools, cmake 등)과 안드로이드 NDK를 사용하여 빌드됩니다. 참조 빌드 환경은 Docker를 통해 제공됩니다.

실험기기	Galaxy A23N(pre-rooting)
운영체제	Android 14
커널버전	4.19.157
사용도구/사용 API	bpfftrace, ExtendedAndroidTools, inotify API
크로스 컴파일 환경	Ubuntu 22.04
디버깅도구	Android Debug Bridge 1.0.41

실험 방법 및 관련 명령어

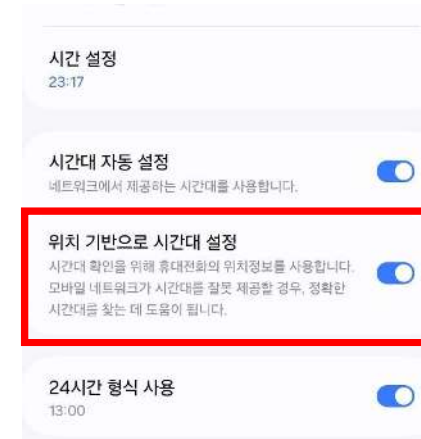
시간(or 타임스탬프) 변조 방법	설명
날짜 및 시간 수동 설정	시스템 시간 변경 (settimeofday 기반 탐지)
네트워크 시간대로 설정	
위치 기반 시간대로 설정	
hwclock -s	
date -s "YYYY-MM-DD HH:MM:SS"	
toybox date -s "YYYY-MM-DD HH:MM:SS"	
touch -d [-m, -d, -t] "YYYY-MM-DD" <file name>	파일의 수정시간 변경 (inotify API 기반 탐지)



시스템 설정 앱에서 시간 변경



네트워크 시간대로 시간 변경



위치 기반 시간대로 시간 변경

실험 결과 및 분석

```
=====
settimeofday() called
PID: 1318
COMM: binder:1318_7
TID: 2429
UID: 1000
Executable: system_server
END_OF_EVENT

=====
settimeofday() called
PID: 13065
COMM: date
TID: 13065
UID: 0
cat: /proc/13065/cmdline: No such file
Executable:
END_OF_EVENT

=====
settimeofday() called
PID: 17150
COMM: toybox
TID: 17150
UID: 0
cat: /proc/17150/cmdline: No such file
Executable:
END_OF_EVENT

=====
settimeofday() called
PID: 13061
COMM: hwclock
TID: 13061
UID: 0
cat: /proc/13061/cmdline: No such file
Executable:
END_OF_EVENT
```

시스템 앱 시간 변경 탐지 결과

date -s 탐지 결과

toybox -s 탐지 결과

hwclock -s 탐지 결과

```
a23:/data/local/tmp # ./probe
Initial mtime for /data/local/tmp/test.txt: 1758345541
Watching file: /data/local/tmp/test.txt
EVENT: mask=0x4 file=/data/local/tmp/test.txt prev=1758345541 new=1758172454 now=1758345577 diff(new-now)=-173123
[ALERT] /data/local/tmp/test.txt mtime moved to past: prev=1758345541 new=1758172454
EVENT: mask=0x4 file=/data/local/tmp/test.txt prev=1758172454 new=1758431654 now=1758345582 diff(new-now)=+86072
[ALERT-FUTURE] /data/local/tmp/test.txt mtime set to FUTURE: new=1758431654 now=1758345582 diff=86072s
```

touch를 이용한 과거/미래로 변경 탐지 결과

	시스템 설정 앱	date	toybox date	hwclock
PID	1318	13065	17150	13061
Com	binder:1318_[1-7]	date	toybox	hwclock
TID	2429	13065	17150	13061
UID	1000	0	0	0
Cmdline	system_server	X	X	X

	과거	미래
기존 mtime	1758345541	1758345582
변경 time_i	1758 172454	1758 431654

settimeofday 감시 결과

inotify API를 통한 감시 결과



05

결론 및 향후 연구

결론 및 향후 연구

❖ 결론

- 안드로이드 폰에서 안티포렌식 행위인 타임스탬프 변경 이벤트 탐지 연구를 수행함
 - settimeofday 후킹을 통해 시스템 시간 조작을 탐지함
 - inotify API, stat()를 이용하여 파일의 타임스탬프 메타데이터 변경을 실시간 탐지 및 비교 함
- 다양한 방법으로 실험하여 제안한 탐지 기법을 검증

❖ 향후 연구

- 제안 기법을 파일 삭제나 파일 이름 조작과 같은 다른 안티포렌식 행위에 이용
- 타임스탬프 변경 행위가 정말 악성행위인지 판별 방법 추가

Acknowledgement

본 연구는 2024년 과학기술정보통신부 및 정보통신기획평가원 의 SW중심대학사업 지원을 받아
수행되었음(2024-0-00035)

또한

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-대학ICT연구센터(ITRC)
의 지원을 받아 수행된 연구임(IITP-2025-RS-2023-00259967)

Thank you

Q&A
