

BPF로 무엇을 할 수 있을까?

2025. 06. 14 대크톡
8기 김은빈

IT과학

SKT 해킹한 'BPF도어'...기업 보안 비상

BPF 활용, 네트워크 통신으로 위장...보안솔루션으로 탐지 어려워

이인애 기자

2025-05-01 13:46:39

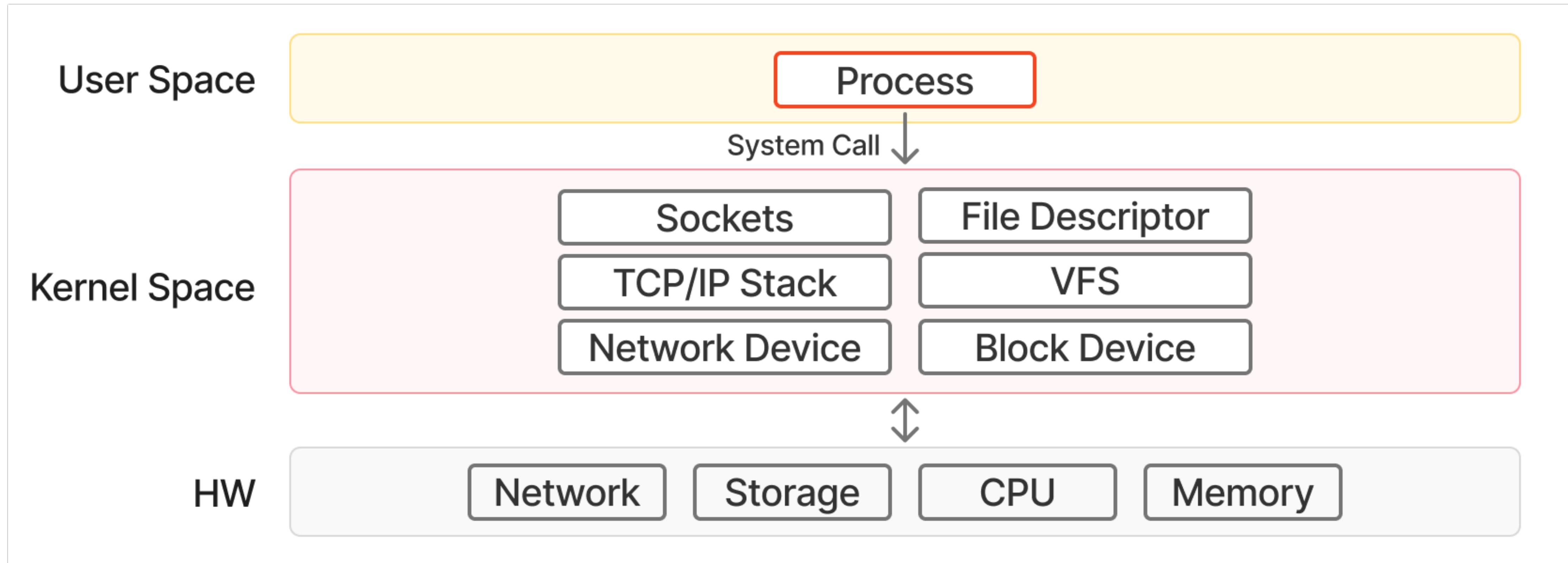
특히 조사단은 침투에 사용된 BPF 도어(Berkeley Packet Filter Door) 계열 악성 코드 4가지를 발견했다고 밝혔다. BPF 도어는 리눅스 운영체제(OS)에 내장된 네트워크 모니터링·필터 기능을 수행하는 BPF를 악용한 백도어(Backdoor)다. 은닉성이 높아 해커 통신 내역을 알아채기 어려운 특징이 있다.

<https://news.mtn.co.kr/news-detail/2025050113420834165>

<https://zdnet.co.kr/view/?no=20250428163730>

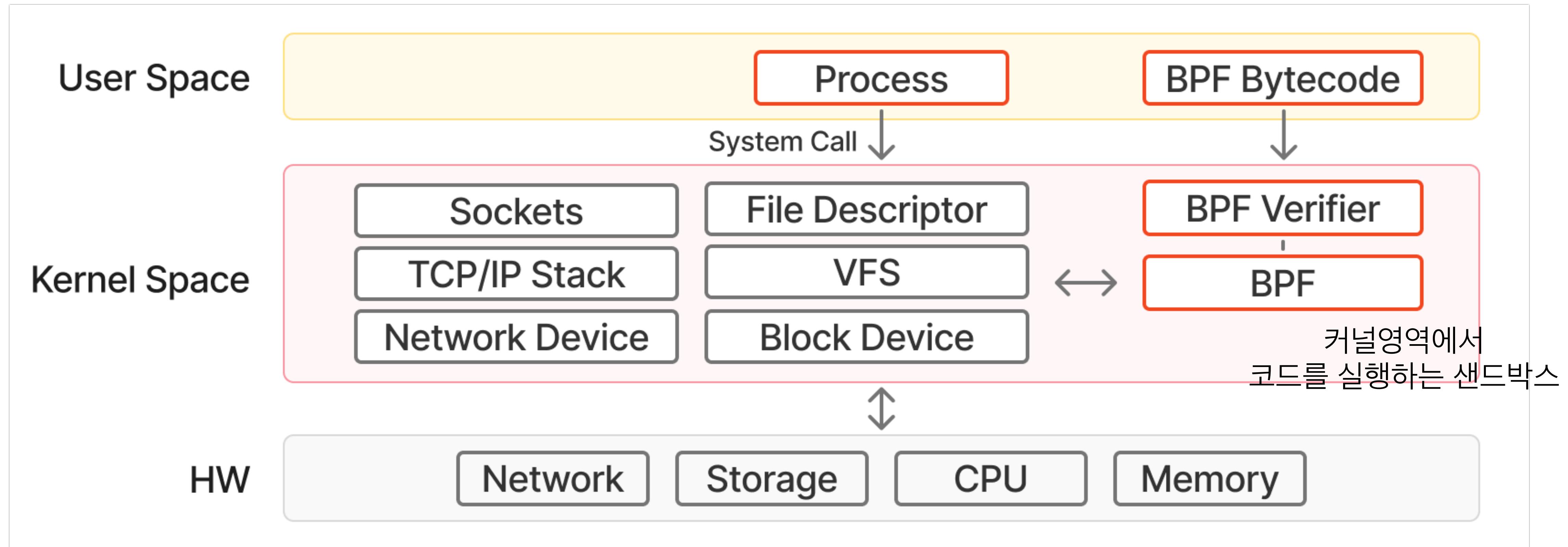
BPF란?

Kernel Space와 User Space



BPF란?

Kernel Space와 User Space



- BPF = Berkeley Packet Filter: 패킷 필터만
- eBPF = Enhanced BPF: 여러 용도로 확장 및 개선된 버전
- 서로 호환 X, 지금의 리눅스 커널은 둘 다 지원

BPF 활용 방식

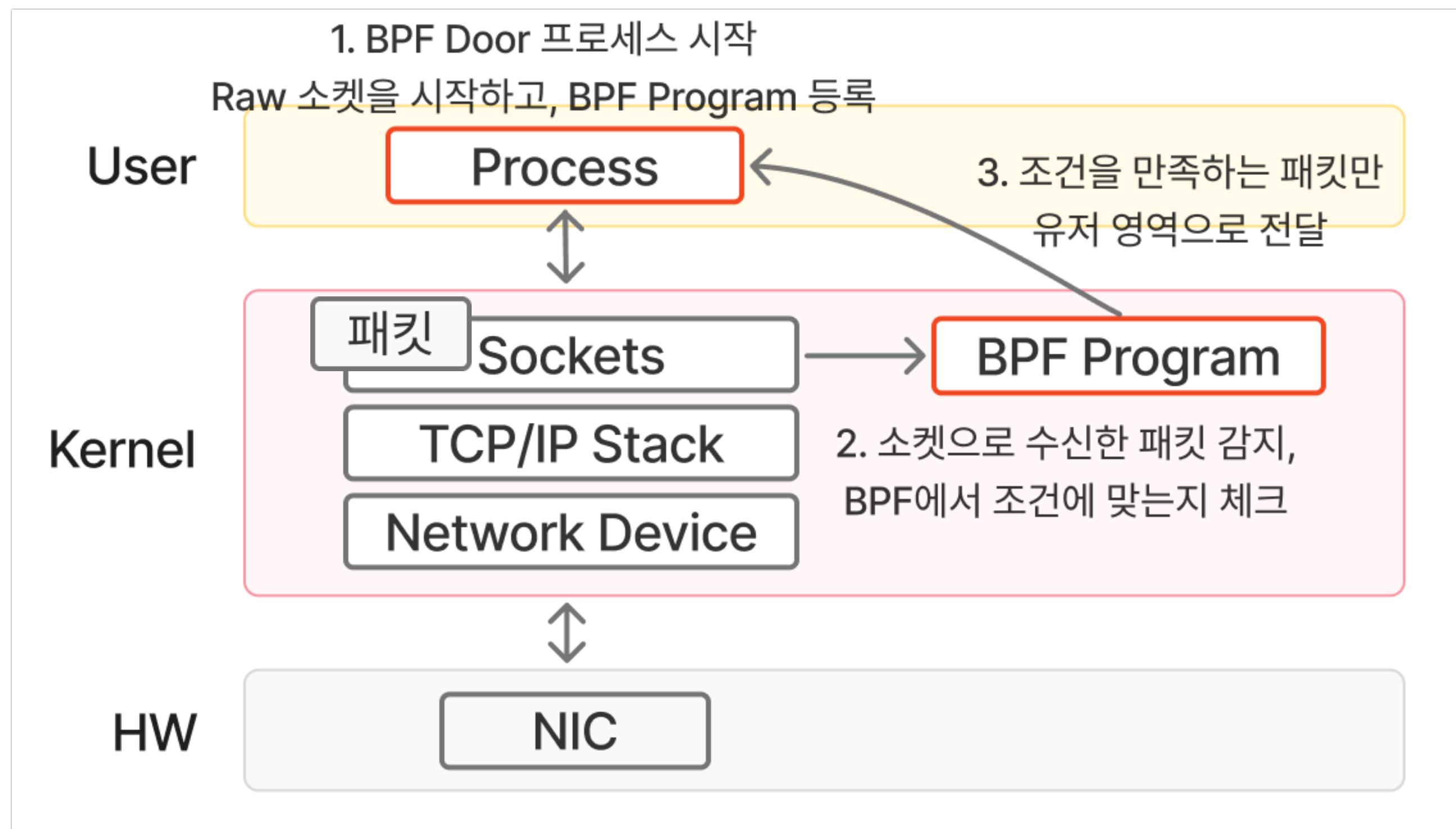
- BPF를 사용하면 커널에서 뭘 하는지 더 빠르게/많이 알 수 있다.
- 커널에서 무언가를 하면 = 이벤트가 발생하면
그 정보를 읽어서 활용해야 할 때 사용 = BPF 코드 실행 (Hook)

예시

- socket_filter: 소켓에서 패킷을 수신할 때
- perf_event: CPU 사용, 메모리 접근 등 성능 이벤트가 발생할 때
- kprobe/uprobe: Kernel/User Space의 함수가 실행 또는 반환될 때

활용 1: BPFDoor

cBPF 용도: 조용히 숨어있다가 나만 아는 신호(매직 패킷)를 감지



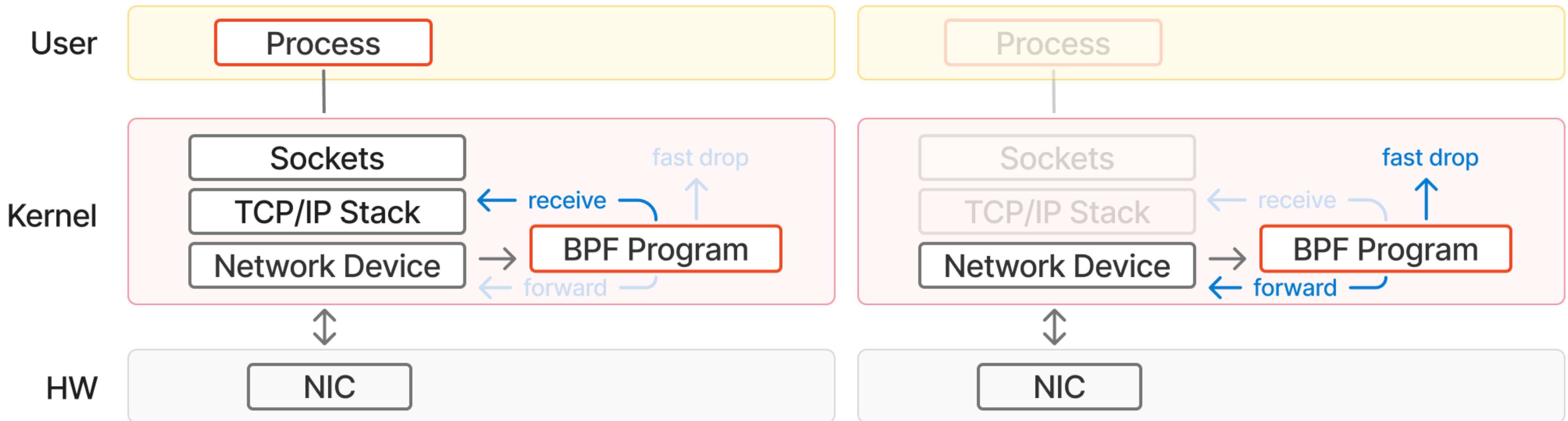
소켓을 여는 목적:
소켓으로 비밀 신호(패킷)을 받으면
문을 다시 열어줘!

Port를 열지 않는 Raw 소켓 사용
Raw 소켓은 모든 패킷을 받음

- BPFDoor = 백도어 = 나중에도 몰래 들어오기 쉽게 뚫은 뒷문
- 신호를 받으면, 공격을 위한 PTY 터미널 통신을 시작

활용 2: XDP(eXpress Data Path)

eBPF 용도: 낮은 계층에서 처리할 수 있는 패킷을 먼저 처리

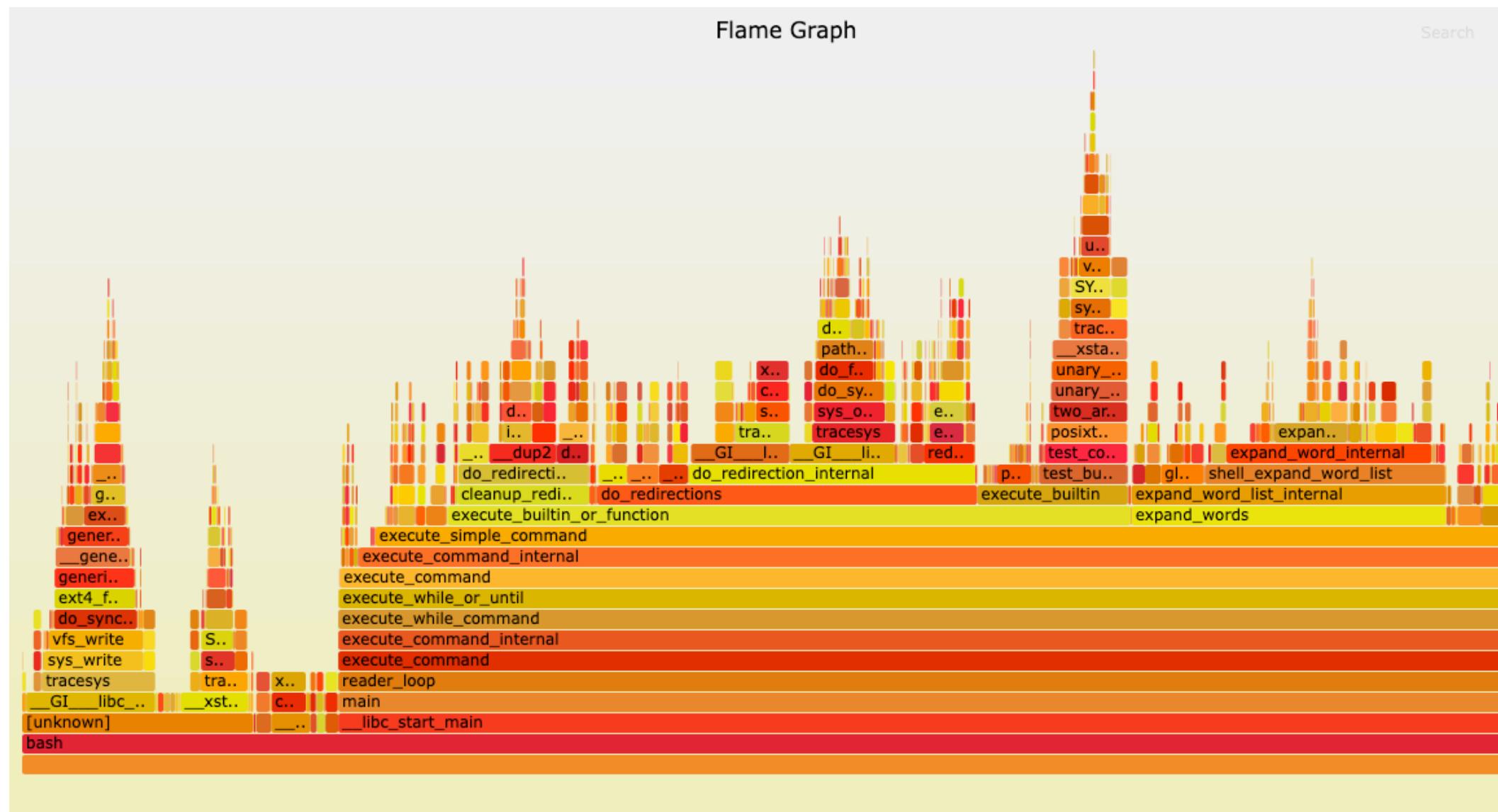


- NIC에서 들어온 L2 프레임에서 바로 drop/forward 처리
 - 공격 트래픽이 급증한 경우에도 안정적인 성능 유지 (e.g. DDoS)
- 커널 영역에서 동작하기 때문에, 나머지 프레임은 커널 TCP/IP 스택에 이어서 전달할 수 있다.

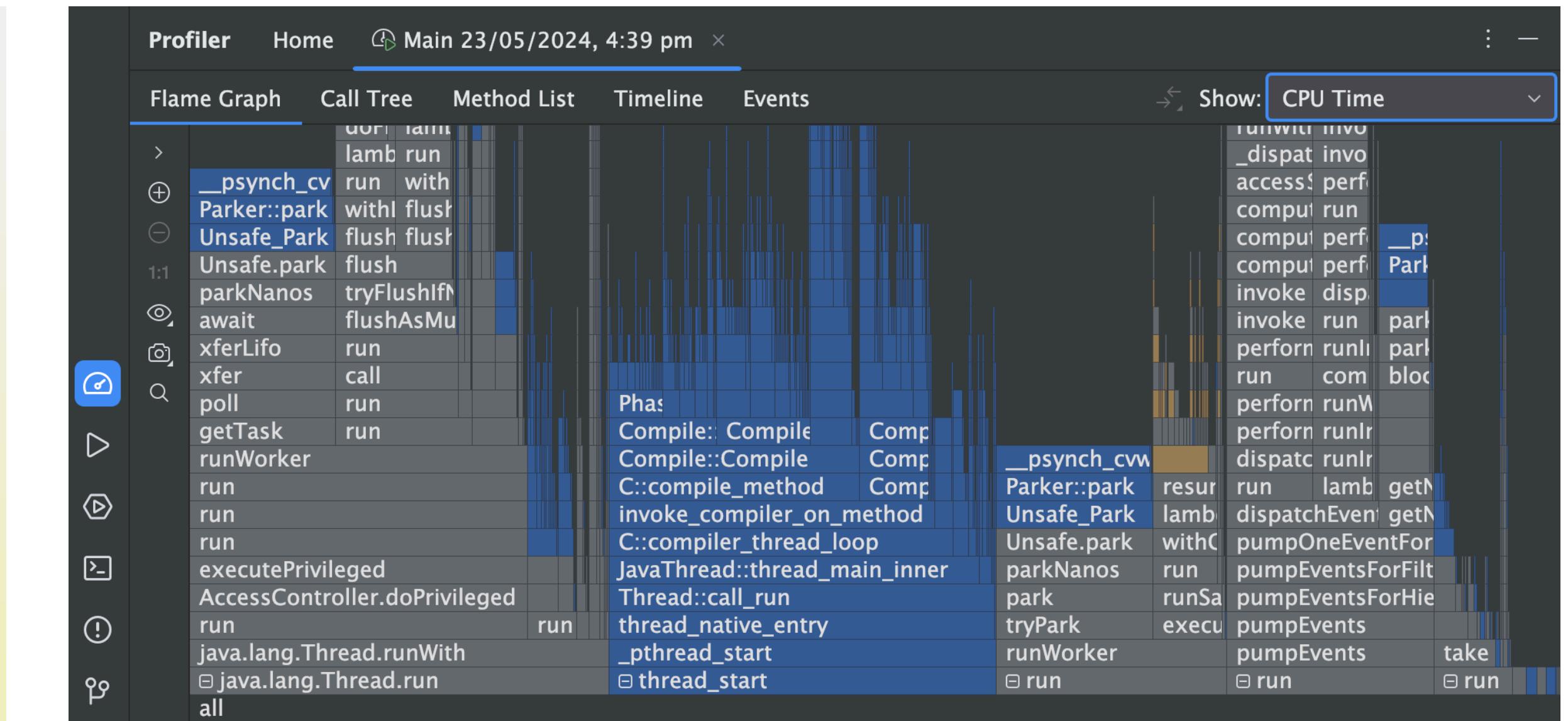
활용 3: Performance Profile

- CPU Profile: CPU가 어떻게 쓰이는지, 어떤 함수(명령어)가 CPU를 얼마나 사용하는지 추적하기
 - 고수준: 런타임 지원 기능 (JVM -XX:+PreserveFramePointer, node --cpu-prof 등)
 - 저수준: eBPF, Perf (linux 커널 트리에 포함된 툴)

CPU Profile 정보를 나타낸 FlameGraph



FlameGraph 툴로 생성

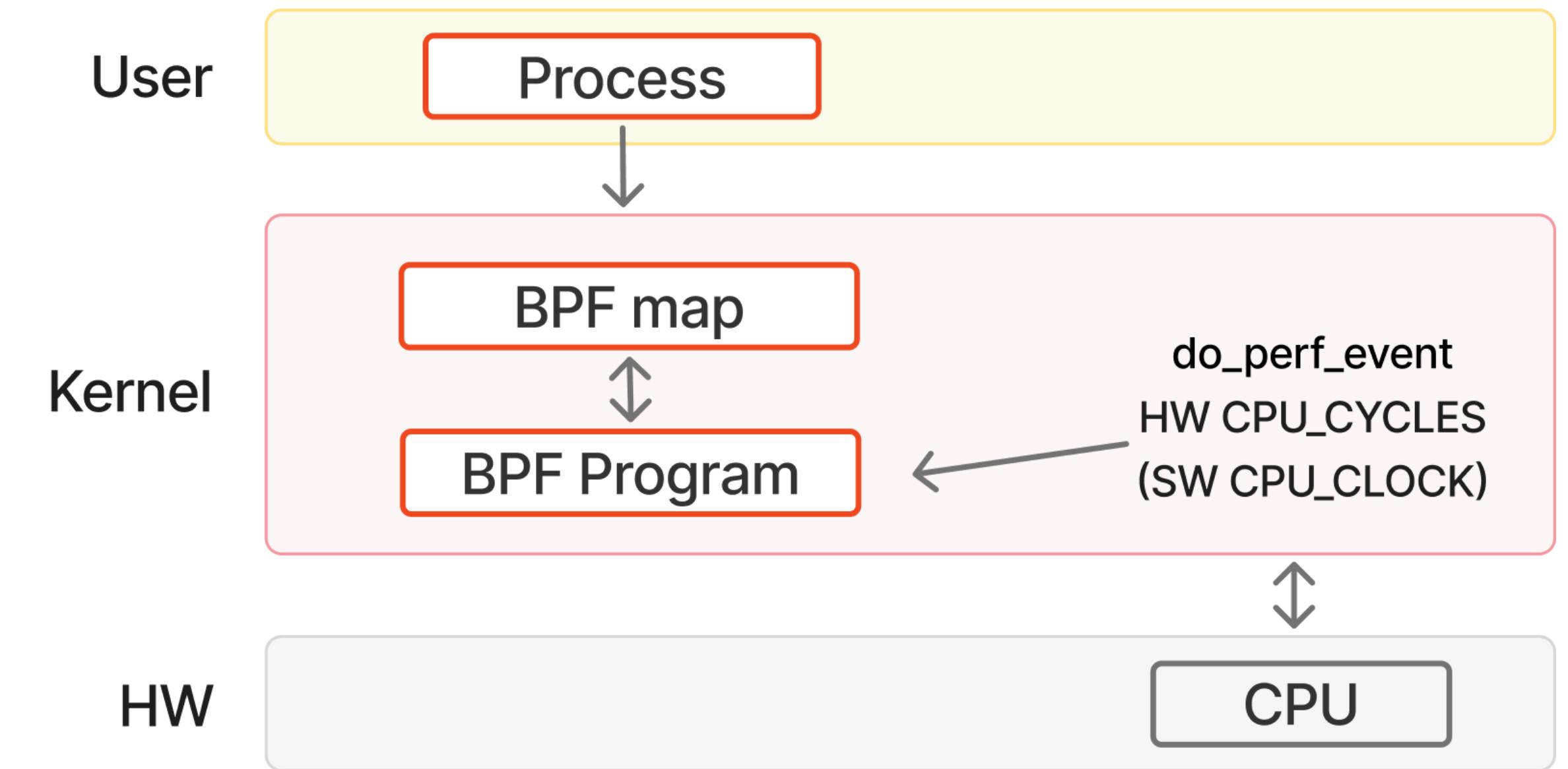


IntelliJ Profiling

활용 3: Performance Profile

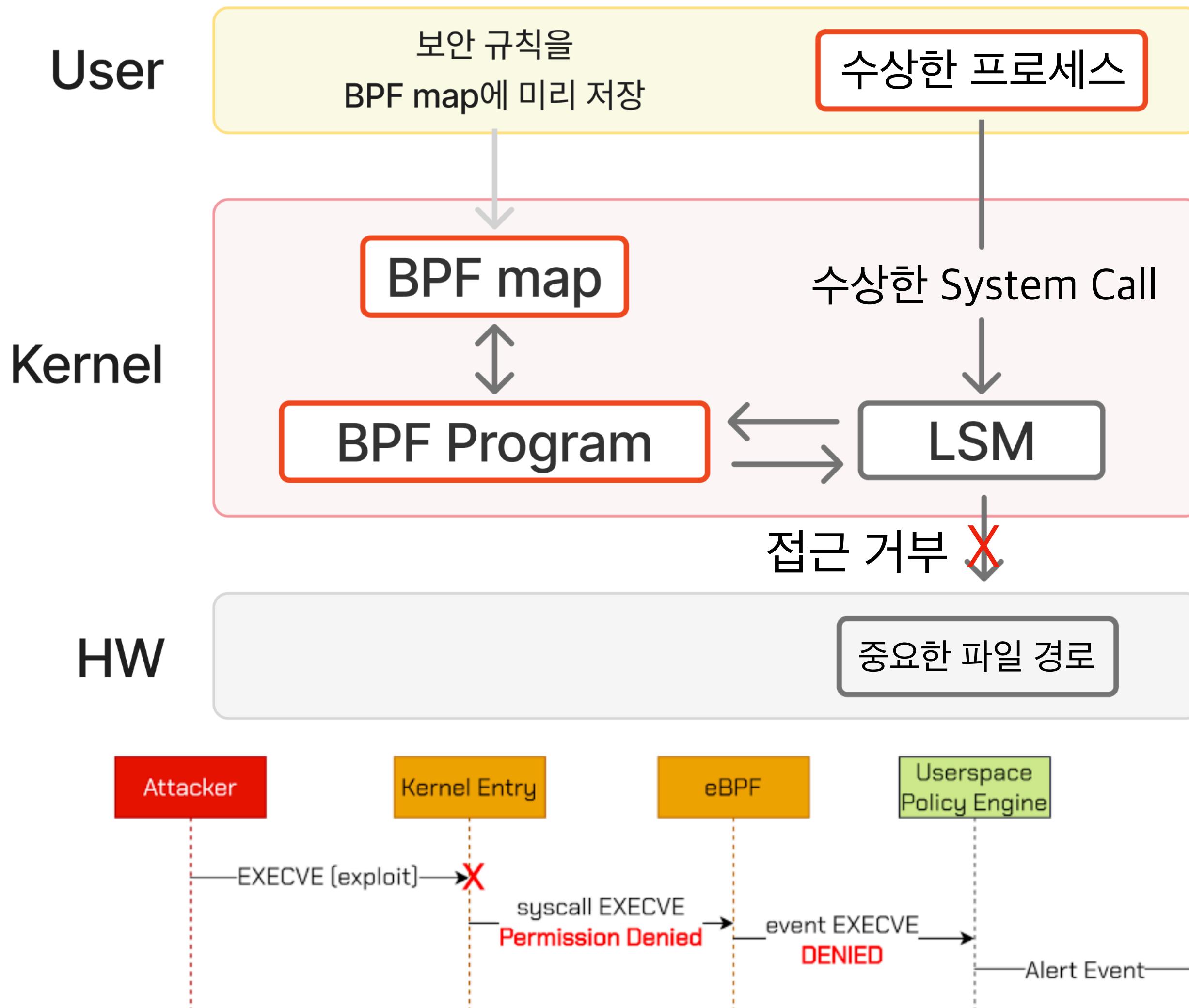
eBPF 용도: 커널에서 수집한 성능 메트릭을 BPF Map에 저장하여 부하 절감

- Perf vs ebpf
 - Perf도 커널 내부에 지정된 계측 지점으로 커널 관련 메트릭을 수집할 수 있음
 - CPU 뿐만 아니라 Memory, Network 등..
- ebpf를 사용할 때의 장점
 - 모든 event를 UserSpace에 전달하지 않고 BPF map에 압축된 수치만 저장
 - 많은 데이터를 Profile해도 더 적은 부하
- 고수준 애플리케이션 함수명 참조도 가능은 함, 하지만 JIT 고려 필요



활용 4: LSM Hook

eBPF 용도: 커스텀 로직에 따라 OS 수준 접근 제한



- LSM(Linux Security Module)와 통합
 - LSM으로 OS 수준 접근 제한
 - 중요한 System call, 파일 접근시 LSM 룰에 맞는지 검증함
- BPF를 LSM Hook에 연결하면 코드를 직접 작성하여 (Programable) 접근 제한 가능, 원하는 형식으로 로깅

예시

- 파일 접근 제한 (file_open event)
- 네트워크 소켓 생성 제한 (socket_create)
- 바이너리 실행 제한 (bprm_check_security)

정리

- BPF(Berkeley Packet Filter):
 - Kernel Space에서 코드 실행하는 안전한 공간.
 - socket_filter, perf_event, kprobe/uprobe 등 이벤트를 hook으로 실행
- eBPF(Enhanced BPF): 패킷 필터링 뿐만 아니라 여러 용도로 확장 및 개선된 버전

활용 사례

1. BPFDoor: 매직 패킷만 필터링하여 전달

2. XDP: 낮은 계층에서 패킷 처리하여 성능 향상

3. Profile: 커널에서 수집한 성능 메트릭을 BPF Map에 저장하여 부하 절감

4. LSM Hook: 커스텀 룰에 따라 OS 수준 접근 제한 (파일 접근, 명령어 실행..)

등 다양함

