Netfilter의 UAF(Use-After-Free) 취약점에 대한 분석 및 eBPF기반 방어 기법 제안

김현문[1], 남경화[1], 최유정[1], 민병준[2]
[1]인천대학교 컴퓨터공학부 학부생
[2]인천대학교 컴퓨터공학부 교수
kimhm1219@inu.ac.kr, ponyvirus@naver.com, lovehk2002@gmail.com, bjmin@inu.ac.kr

Analysis of Netfilter's Use-After-Free vulnerabilities and suggestion of eBPF-based Defense techniques

Hyeonmoon Kim[1], Kyeong-Hwa Nam[1], Yu-Jung Choi[1], Byoungjoon Min[2]
[1]Dept. of Computer Science, Incheon-National University(Undergraduate student)
[2]Dept. of Computer Science, Incheon-National University(Professor)

요 약

최근 Linux 커널의 Netfilter 프레임워크에서 발생한 취약점들 중, Use-After-Free(UAF) 및 Double Free를 유발하는 사례가 꾸준히 보고되고 있으며, 이는 메모리 조작을 통한 권한 상승 공격으로 이어질 수 있어 보안상 큰 위협이 되고 있다. 본 논문에서는 이러한 사례 중 하나인 CVE-2024-1086을 분석 대상으로 삼고, 해당 취약점이 nf_tables 구성 요소에서 skb 구조체를 대상으로 발생한다는 점에 주목하였다. 이를 바탕으로 취약점의 구조와 보안 위협을 분석하고, 이를 탐지 및 완화하기 위한 대응 방안으로 eBPF(extended BPF) 기반의 모니터링 프레임워크를 제안한다.

1. 서론

네트워크 트래픽의 필터링과 제어를 담당하는 Netfilter 프레임워크는 Linux 커널에서 중요한 위치를 차지하며, iptables, nftables 등의 방화벽 도구들이 Netfilter 위에서 작동한다. Netfilter는 패킷 필터링, 주소 변환, 포트 포워딩, 등의 다양한 기능을 제공하고, 클라우드 플랫폼의 보안 구성 및 네트워크 경계 방어에서도 핵심적인 역할을 수행한다.

그러나 이러한 구조적 복잡성과 상태 관리의 어려움으로 인해, Netfilter는 최근 수년간 반복적으로 심각한 보안 취약점, 특히 UAF 메모리 오류에 노출되어 왔다.[1] 하지만 기업에서는 시스템의 중단, 복잡한 의존성 검증, 전문가 부족, 패치 후의 부작용 가능성 등을 이유로 커널에서 취약점이 발생해도 쉽게 패치하지 못하는 경우가 많다.[2]

본 논문에서는 Netfilter에서 UAF로 인해 발생한 CVE들을 조사하고, 그중 하나인 CVE-2024-1086을 자세히 분석하였다. 이를 탐지 및 완화하기 위해 CVE의 PoC 코드를 분석하고, Linux 커널의 패치 없이도 커널 함수의 모니터링 및 필터링이 가능한 eBPF로 프레임워크를 제작하여 커널의 취약점 탐지에 eBPF가 활용될 수 있음을 주장하고자 한다.

2. CVEs

2.1 CVE List

<표 1> 은 2023년부터 2025년 사이에 공개된 Netfilter UAF 관련 주요 CVE들을 정리한 것이다.

CVE ID	Year	Vulnerability	CVSS 3.x
CVE-2025-21714	2025	Linux Netfilter UAF	7.8
CVE-2024-1086	2024	Linux Netfilter UAF	7.8
CVE-2024-1085	2024	Linux Netfilter UAF	7.8
CVE-2024-50130	2024	Linux Netfilter UAF	7.8
CVE-2023-3610	2023	Linux Netfilter UAF	7.8
CVE-2023-31248	2023	Linux Netfilter UAF	7.8

<표 1> Netfilter CVE list

최근 수년간 Netfilter의 핵심 구성 요소인 'nf_tables'와 그 주변 모듈들에서 연달아 심각한 보안 취약점들이 발견되었으며, 최신 커널에서도 UAF를 악용한 높은 위험도를 지닌 CVE들이 보고되고 있다. 이는 Netfilter 모듈의 메모리 관리와 참조 처리에 구조적인 한계가 발생해 UAF 취약점에 반복적으로 노출됨을 의미한다.

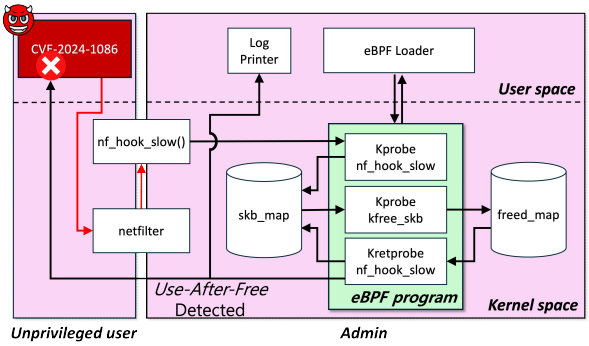
2.2 CVE-2024-1086

CVE-2024-1086은 Linux Kernel 내 Netfilter 서브 시스템 nf_tables의 nft_verdict_init 및 nf_hook_slow 내에서 발생하는 UAF 및 Double Free 취약

점으로, 사용자가 nft_verdict_init을 통해 설정한 verdict.code값의 상위 16비트를 검증하지 않아 발생한다. 공격자는 이를 악용해 verdict.code에 0xFFFF0000을 입력하여 nf_hook_slow에서 하위 16비트로 NF_DROP 처리를 유도하고 skb를 해제한다. 동시에 NF_DROP_GETERR에서 해당 값을 이용한 연산 결과가 오버플로우로 인해 1이 되어 NF_ACCEPT를 의미하는 값을 반환한다. 이로 인해 이미 해제된 skb를 재참조하게 되면서 UAF가 발생하고, 이후 해당 skb를 재해제하며 Double Free로 이어진다. 이를 악용하면 공격자는 메모리 조작이 가능하기 때문에 임의 코드를 실행하여 권한 상승이 가능해진다.

3. 설계

우리는 커널 함수의 호출 및 반환 시기 추적을 위해 eBPF의 Kprobe와 Kretprobe를 사용하였다. 추적의 결과를 충분한 크기의 eBPF map에 저장하여 eBPF 프로그램의 메모리 한계를 보완하고, map을 공유하여 서로 다른 hook에서도 정보를 활용할 수 있도록 설계했다.



(그림 1) eBPF 프로그램 아키텍처

비권한 사용자인 공격자가 (그림 1)의 상황처럼 루트 사용자 권한을 얻기 위해 CVE-2024-1086을 악용해 localhost로 악성 패킷을 보낸다. 이를 처리하기 위해 Netfilter가 취약 함수인 nf_hook_slow를 호출하면 해당 함수를 후킹 중이던 Kprobe가 함수의 시작을 감지하여 매개변수로 입력된 skb와 해당 프로세스의 thread group id(tgid)를 skb_map에 저장한다. nf_hook_slow는 switch문의 결과가 NF_DROP일 경우 kfree_skb를 호출하여 skb의 메모리 공간을 free 상태로 만든다. 이때, Kprobe로 kfree_skb를 추적해 free되는 skb와 tgid가 skb_map에 있는지 확인하고, 있다면 freed_map에 저장한다. Kretprobe를 이용해 nf_hook_slow가 return을 반환한 직후 tgid를 비교해 skb_map에서 skb를 찾아오고 해당 skb가 freed_map에 있으면서 return값이 NF_DROP이 아닐 경우 UAF의 발생으로 인식하고 해당 프로세스에 SIGKILL 명령을 보내 공격 프로세스를 종료시킨다.

4. 평가

본 실험은 Linux kernel 5.15.25 버전에서 진행되어, 오픈소스 PoC 코드[3]로 CVE를 재연하였다. 일반 사용자 계정에서 PoC 코드를 실행할 경우 손쉽게 루트 사용자 권한을 탈취할 수 있다.

추적을 위해 eBPF 프로그램 실행 시 취약 커널 함수를 후킹하고, 보안 위협 발생 시 log를 작성함과 동시에 프로세스에 SIGKILL 명령을 보내어 (그림 2)와 같이 공격 프로세스가 루트 사용자 권한을 취득하지 못하고 멈추는 것을 확인할 수 있다.

(그림 2) eBPF 프로그램 실행 후 PoC 실행 결과 및 log

5. 결론

본 논문에서는 Netfilter에서 발생한 CVE를 조사하고 그 중 하나인 CVE-2024-1086을 eBPF 기반 보안 프레임워크로 차단하여 Netfilter에 관련된 커널 함수의 모니터링 및 필터링에 eBPF가 활용할 수 있음을 보일 수 있었다. 이는 다른 커널 프로세스에서 발생한 취약점에도 충분히 활용될 여지가 있으므로 향후 추가적으로 eBPF를 이용한 연구를 진행할 예정이다.

참고문헌

[1] Omar Jarkas, Ryan Ko, Naipeng Dong, and Red owan Mahmud. 2025. A Container Security Surve y: Exploits, Attacks, and Defenses. ACM Comput. Surv. 57, 7, Article 170 (July 2025), 36 pages.

[2] V. A. Mehri, P. Arlos and E. Casalicchio, "Autom ated Patch Management: An Empirical Evaluation Study," 2023 IEEE International Conference on Cyber Security and Resilience (CSR), Venice, Italy, 2023, pp. 321-328

[3] Notselwyn."CVE-2024-1086".2024.[Online] Available:https://github.com/Notselwyn/CVE-2024-1086