안티템퍼링 시나리오 기반의 VFS 조사 및 식별

최재민*, 허남정*, 박준영**, 장우현***, 김연재***, 허재원***, 박기웅†

*,**세종대학교 SysCore Lab. (대학원생*, 학부연구생**)

LIG넥스원 (연구원)

†세종대학교 정보보호학과 (교수†)

Anti-Tampering Scenario-Based VFS Investigation and Identification

Jae-Min Choi*, Nam-Jung Heo*, Jun-Young Park**, Woo-Hyun Jang***, Yeon-Jae Kim***, Jae-Won Huh***, Ki-Woong Park†

*,**SysCore Lab., Sejong University (Graduate Student*, Undergraduate Student**)

LIG Nex1 (Researcher)

†Dept. of Computer and Information Security, Sejong University (Professor†)

요 약

현대전이 고도화됨에 따라 더욱 복잡한 연산 및 제어 작업을 무기체계에서도 수행하기 위해서 리눅스 기반의 운영체제 시스템을 탑재하고 있다. 이러한 무기체계는 수출 대상국 또는 공격자가 역공학을 통해 내부의 중요한 군사 비밀 또는 설계도를 추출할 수 있다. 이에 대응하여 여러 안티-템퍼링 기술을 개발 및 공급하여 무기체계에 탑재하고 있다. 하지만, 기존의 안티-템퍼링 기술들은 시스템의 오버헤드를 증가시키는 문제점이 존재한다. 본 연구에서는 VFS 계층을 모니터링하여 템퍼링 상황을 인지하여 오버헤드를 감소시키는 방법을 제안한다. 또한 세 가지 템퍼링 시나리오에 대하여 이를 탐지할 수 있는 VFS 리스트를 생성하고 실증을 진행한다. 이를 통해 향후 연구에서는 템퍼링 상황을 감지하고 내부 구성 요소를 변경하여 효과적인 안티-템퍼링 기술을 구현할 수 있을 것이다.

I. 서론

현대전이 고도화됨에 따라 더욱 복잡한 연산 및 제어 작업을 무기체계에서도 수행하기 위해 리눅스/Unix 기반의 OS를 탑재하여 연산 작업에 활용하고 있다. 이러한 무기체계는 공격자로부터의 여러 템퍼링 기술에 대응하기 위해 각 사에서 여러 안티-템퍼링 기술을 개발 및 공급하여 무기체계에 탑재하고 있다 [1].

기존의 리눅스에서도 무결성 보호 메커니즘을 제공하는데, 해시값을 통해 시스템에서 실행되거나 접근되는 파일 및 바이너리 등의 무결성을 측정하는 아키텍처(IMA)와 파일의 메타데이터가 무단으로 변경되는 것을 보호하기 위한 확장 확인 모듈(EVM)이 있다.

이들 시스템은 사전에 정의된 정책을 기반으로 동작하지만 해당 정책을 변경하기 위해 수정하는 것은 쉽지 않으며 증가하는 측정 데이터는 시스템에 추가적인 오버헤드를 초래한다.

이와 같은 문제를 해결하고자 다음과 같은 Virtual Filesystem(VFS) 기반의 탬퍼링 탐지 기술을 제안하고자 한다.

† 교신저자: 박기웅 (세종대학교 정보보호학과 교수)

이 논문은 2022년 정부(방위사업청)의 재원으로 국방기술진흥연구소의 지원을 받아 수행된 연구임 (KRIT-CT-22-051)

VFS란 애플리케이션이 특정 파일시스템에 구애받지 않고 일관된 인터페이스를 통해 파일에 접근할 수 있도록 하는 계층을 의미한다. 이를 기반으로 한 템퍼링 탐지 기술은 VFS 계층에서의 모니터링 방식을 통해 시스템 내부에서 발생하는 비정상적인 행위를 탐지함에 있어 적은 오버헤드와 레이턴시로 탐지가 가능하다.

본 연구에서는 공격자의 관점에서 시스템 변경을 위한 여러 시나리오를 설계하고, 해당 시나리오를 바탕으로 VFS 계층을 기반으로 한 모니터링 방식이 시스템 내부에서 발생하는 비정상적인 행위를 탐지함에 있어 유효함을 실증적으로 검증한다.

본 논문의 구성으로는 2장에서 관련 연구를 소개하고 3장에서는 리눅스 VFS를 통한 템퍼링 탐지 시나리오와 탐지 가능 VFS 리스트를 서술하고, 5장에서는 결론을 서술한다.

II. 관련 연구

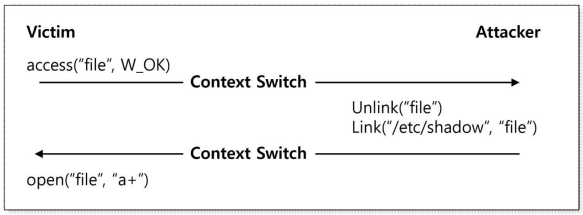
2.1. VFS 계층의 모니터닝 도구를 활용한 비정상적인 파일 접근 탐지 연구

S. Mehaz et al [2]는 VFS 계층의 I/O 모니터링 성능 도구인 blktrace를 사용하여 사용자의 파일 접근 세부 이벤트를 수집하고 모니터링을 수행하여 사용자의 비정상적인 시스템 접근을 식별하는 데 사용한다. 이러한 VFS 계층의 도구는 리눅스 운영체제 런타임 중에서 모니터링을 수행하므로 성능 오버헤드는 거의 미미하다.

2.2 VFS 기반 모니터링 기법 제안

E. Manenti et al [3]는 리눅스 커널의 VFS 계층에서 동작하는 모니터링 기법을 제안하였다. 특정 파일 연산이 발생할 때 해당 호출을 커널 수준에서 후킹하여 파일 접근을 실시간으로 감시한다. 이러한 모니터링 방식을 통해 사전에 정의된 보호 경로에 대한 쓰기 및 삭제 요청을 실시간으로 차단하며, 루트 권한을 가진 프로세스도 보호정책을 우회할 수 없도록 설계하였다.

또한, 지정된 보호 경로에는 시간 기반의 정책이 적용되어 해당 시간 동안은 파일의 속성 변경이나 데이터 수정이 불가능하도록 하여 파일 및 디렉터리의 무결성을 보장한다.



(그림 1) Race Condition Attack

III.안티템퍼링 시나리오 기반의 VFS 조사

3.1 시나리오 A: UART 기반의 접근 및 정보 유출

해당 시나리오를 수행함에 있어 각 단계를 나열하자면 다음과 같다. 공격자가 무기 체계에 들어가는 하드웨어를 탈취하고 UART 접근을 수행한다. 이를 통해 낮은 권한의 쉘을 획득한다. 공격자는 Root 권한을 얻기 위해서 Setuid가 설정된 레이스 컨디션에 취약한 애플리케이션을 찾아서 권한 상승 공격을 수행한다. Root 권한을 획득하면 UART를 통해서 내부 데이터를 추출한다.

해당 시나리오의 주요 탐지 VFS는 다음과 같다. 해당 하드웨어에 UART가 활성화되어 있는 경우, /sys/kernel/debug/gpio 및 /dev/<tty>에서 확인할 수 있다. UART를 통한 입력은 /proc/interrupts 및 /proc/tty/driver/<device>에서 UART 입력과 출력의 카운터 정보를 확인할 수 있다.

Time-Of-Check Time-Of-Use (TOCTOU) 기반의 Race Condition 취약점은 Root 권한의 프로그램이 특정 자원에 접근하기 전에 상태를 확인하는 과정과 파일 변경하는 과정 시간 사이에 공격자에 대상 자원에 링크 정보를 변경하는 방식이다. (그림 1)은 이러한 공격 방식을 도식화한 것이다.

레이스 컨디션 공격 과정에서 대량의 Link 및 Unlink가 발생하므로 /proc/slabinfo의 Dentry 필드의 크기가 빈번하게 바뀌게 된다. 해당 필드는 리눅스 파일시스템에 대한 파일 및 폴더의 캐시 정보를 제공한다.

이러한 일련의 과정을 연계하여, 디버깅 환경에서 정상 행위 또는 공격자의 비정상적인 행위를 식별하고 공격자의 공격 표면에 대하여

[표 1] 각 템퍼링 상황별 VFS List

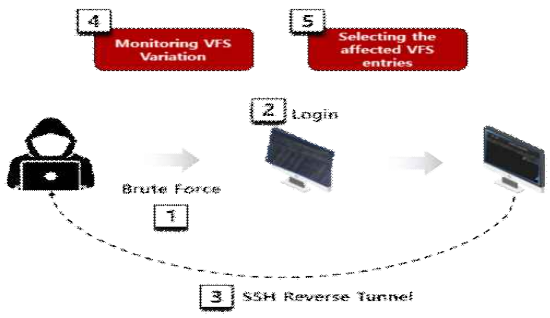
No.	Tempering Scenario	Main VFS	Side VFS
1	Serial Attach	/proc/interrupts, /proc/tty/driver/<device>	/proc/stat, /dev/<tty>, /sys/kernel/debug/gpio,
2	USB Attach	/sys/block, /sys/bus, /sys/block/<disk>/stat, /proc/diskstats , /dev/<disk>, /sys/block/<device>	
3	Data Backup	/proc/diskstats, /sys/block/<disk>/stat	/sys/block/<disk>/inflight
4	Network Connect	/proc/net/netstat, /proc/net/tcp, /proc/net/udp, /proc/net/wireless, /dev/pts	/sys/class/net, /sys/class/net/<interface>/addr_assign_type
5	Others	/proc/slabinfo	/proc/<pid>/stat

변화를 발생시켜 하나의 공격 표면을 보다 손쉽게 무력화 시키는 효과를 낼 수 있다.

3.2. 시나리오 B: SSH 리버스 터널링

해당 시나리오를 수행함에 있어 각 단계를 나열하자면 다음과 같다. 공격자는 원격에서도 시스템 내부의 셸에 지속적으로 접근하는 것이 가능하도록 시스템을 변조하기 위해, Brute force로 로그인 인증에 성공한 후 SSH 리버스 터널을 연결한다. 해당 시나리오의 주요 탐지 VFS는 다음과 같다.

공격자가 로그인 인증에 성공하여 해당 시스템에서 SSH 리버스 터널을 열고 그 후 원격 접속을 수행하면 정상적인 무기체계의 임무 수행 중 생성되지 않았어야 할 터미널이 생성된다. 터미널이 생성되면, VFS /dev/pts가 변하게 된다. 해당 VFS와 /run/sshd.pid에서 출력된 값과 /proc/<pid>/stat와 연계하여 프로세스의 상태, 부팅 후 시작된 시간, 사용자 등을 통하여 정상 행위 혹은 디버깅 환경이 아닌 누군가로부터 공격이 수행되고 있음을 식별할 수 있다. 아래의 (그림 2)는 해당 시나리오의 수행 과정을 나타낸다.



(그림 2) VFS 기반 템퍼링 탐지 시나리오

3.3. 시나리오 C: USB Attach를 통한 데이터 유출

재밍과 포획 등으로 장치를 탈취에 성공한 공격자는 시스템의 임무 혹은 암호와 같은 데이터를 획득하기 위해 USB 포트에 장치를 연결할 수 있다. 이럴 경우, /dev 경로에 블록 디바이스 파일인 /sd*와 /proc/mounts, /run/media[uer ID]/ 디렉터리 등이 변하게 된다. 다만 실제 사용자의 장치 연결인지 혹은 공격자의 장치 연결인지를 판단해야 하는데, 이는 장치의 일련번호를 나타내는 /sys/bus/usb/ devices/[bus no]-[port no]/serial 및 idProduct의 확인을 통해 장치 소유자를 식별할 수 있다. 이후 해당 장치를 통해 시스템에 있던 데이터를 수집할 경우 I/O 작업이 발생하여 /proc/diskstats 등이 변하게 된다.

3.4. 실험 결과

시나리오 A, B, C를 통해 식별한 VFS는 [표 1]과 같다. 해당 VFS 리스트는 각 탬퍼링 행위 Serial Attack, USB Attach, Data Backup, Network Connect 등 안티 탬퍼링 상황에서 발생하는 행위별로 VFS를 조사하였다. 이때, 디버깅 혹은 정상 행위를 판단하기 위하여 VFS를 단일 1:1 매핑하는 것이 아닌 여러 VFS 리스트를 조합하고, Side VFS를 선정하여 해당 VFS의 무결성 검증 및 정확도를 높이는데 활용한다.

IV. 결론 및 향후 연구

본 연구에서는 템퍼링 상황을 식별하기 위해서 VFS 계층을 기반으로 비인가자의 템퍼링 행위에 대한 모니터링을 수행한다. 제안된 기법에 대한 실증으로 세 가지의 템퍼링 상황을 설정하고 시나리오를 바탕으로 실험을 통해 시스템 내부의 탐지를 수행할 수 있는 Main VFS 와 Side VFS 리스트를 생성할 수 있다.

향후 연구에서 템퍼링 상황을 탐지될 경우 내부적인 알고리즘을 통하여 내부 구성 요소를 동적으로 변경하여 시스템을 재구성함으로써 템퍼링 상황에 대응할 수 있는 보호 기술로 확장하고자 한다.

[참고문헌]

[1] 주영진 et al, “무기체계 기술보호를 위한 안티탬퍼링 시험평가 방안”, 한국방위산업학회지, pp. 47-58, 2023.

[2] S. Mehnaz and E. Bertino, "A Fine-Grained Approach for Anomaly Detection in File System Accesses With Enhanced Temporal User Profiles", in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 6, pp. 2535-2550, 1 Nov.-Dec. 2021.

[3] E. Manenti, P. Caporaso, G. Bianchi, and F. Quaglia, “VFSMon: an Innovative Reference Monitor in linux”. CEUR Workshop Proceedings, vol. 3962, paper 46, 2025.

[4] 이석원, 김문회, and 오희국. "바이너리 분석을 통한 UNIX 커널 기반 File System 의 TOCTOU Race Condition 탐지." 정보보호학회논문지 31.4 (2021): 701-713.