

[리눅스 보안 시리즈] 1화 - 사례로 보는 리눅스 공격, 무엇부터 해야 하나?

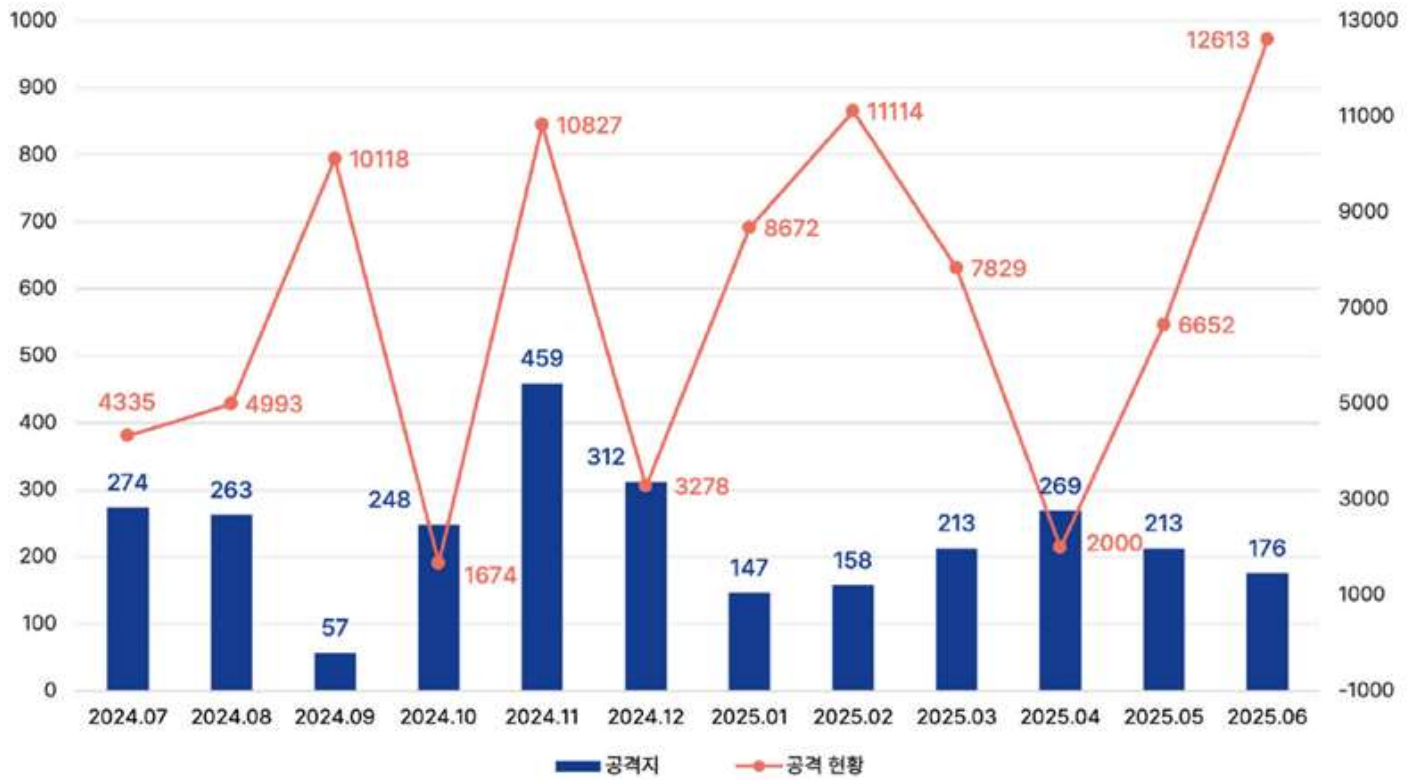
지난 4월, 국내 통신사의 리눅스 서버가 해킹 당하며 다수 고객 유심 정보가 유출되는 사고가 있었다. 해당 사건은 윈도우 기반 시스템 대비 관심에서 조금은 멀어져 있던 리눅스 서버 보안의 중요성을 다시 한 번 상기시켰다. 다만, 통신사 해킹 사례의 사회적 파급력이 컸던 것은 사실이나, 그 외에도 리눅스 서버를 향한 공격은 지속적으로 일어나고 있었다. 이제 기업들은 당사 중요 자산과 고객 정보를 다루는 리눅스 서버를 우선순위에 두고 보안을 적용해야 한다.

이번 글에서는 리눅스 서버 공격 통계와 사례, 그리고 리눅스 서버 보안을 위해 꼭 갖춰야 할 보안 요소들을 살펴본다.



1. 통계로 보는 리눅스 서버 공격

안랩의 위협 인텔리전스 조직인 ASEC(AhnLab Scurity intelligence Center)은 허니팟을 활용하여 부적절하게 관리되고 있는 리눅스 SSH(Secure Shell) 서버 대상 무차별 대입(Brute Forcing), 사전 공격(Dictionary Attack) 등을 탐지 및 분류하여 통계를 제공하고 있다. 여기서 부적절하게 관리된다는 것은 공격에 취약한 계정 정보가 설정되어 있는 환경을 의미한다.



[그림 1] 2025년 6월 기준 1년 간 리눅스 서버 공격 통계

[그림 1]은 2025년 6월을 기준으로 지난 1년 간 국내 리눅스 서버를 대상으로 감행되었던 공격 통계다.

각 항목을 살펴보면, ‘공격지’는 악성코드 또는 공격자에 의해 사용된 시스템의 수량으로, 실제 악성코드 설치 명령까지 수행된 이력이 확인되는 시스템이다. 만약, 공격자가 부적절하게 관리되고 있는 시스템에 관리자 계정으로 로그인에 성공한다면 해당 시스템에 대한 제어가 가능해진다.

‘공격 현황’은 악성코드 또는 공격자가 해당 시스템을 대상으로 공격을 수행한 횟수다. 리눅스 SSH 서버 공격은 스캐닝(scanning)부터 시작해 무차별 대입 혹은 사전 공격을 통해 계정 정보를 획득하거나 기본적인 정보를 수집하는 과정을 거친다. ‘공격 현황’은 이러한 과정을 수행한 뒤 실제 악성코드를 설치한 로그가 확인된 사례들이다.

리눅스 서버 공격에 사용된 악성코드 유형은 디도스 봇(DDoS Bot), 코인 마이너(CoinMiner), 백도어(Backdoor), 랜섬웨어(Ransomware) 등으로 다양하다. 다음은 각 악성코드 유형에 대한 간단한 설명이다.

디도스 봇: 공격자의 명령에 따라 감염 시스템을 제어하고 디도스 공격을 수행할 수 있도록 하는 악성코드. 추가 페이로드를 설치하거나 기타 명령을 수행할 수 있는 기능을 포함한다.

코인 마이너: 감염 시스템의 리소스를 활용해 가상 화폐를 채굴하는 악성코드.

백도어: 공격자가 시스템에 은밀하게 접속해 추가적인 악성 행위를 수행할 수 있도록 하는 악성코드.

랜섬웨어: 감염 시스템 내 파일 등을 암호화하고 공격자가 다양한 형태로 몸값(ransom)을 요구할 수 있도록 하는 악성코드.

[그림 1] 통계를 보면, 가장 최근인 6월에는 176개 시스템을 대상으로 무려 1만 2천 건 이상의 공격이 감행되었다. 지난 1년 동안의 흐름을 보면, 월 별로 차이는 있지만 대략적으로도 매달 100개 이상의 시스템에 수 천 건 많게는 1만 건 이상의 공격이 수행됐다. 서두에 언급한 바와 같이 리눅스 서버 공격은 대형 해킹 사건으로 인해 최근 관심도가 높아졌지만, 사실 공격 자체는 오랜 기간 활발하게 수행되고 있었다.

2. 리눅스 서버 공격 사례

다음으로, 실제 사례 분석을 통해 리눅스 서버 공격에 대해 알아보자. 이번 글에서는 두 가지 사례를 준비했다.

사례 1: BPFDoor(백도어)를 통한 정보 탈취

BPF(Berkeley Packet Filter)는 네트워크 패킷 필터링을 위해 개발된 메커니즘으로 커널 영역에 설치되어 외부로부터 전달받은 패킷을 사용자 영역으로 넘길 것인지 결정할 수 있다. BPFDoor는 이 BPF의 패킷 필터링 기능을 악용하는 리눅스 백도어 악성코드다. 패킷 필터링 규칙을 추가해 조작된 특정 패킷인 ‘매직 패킷’이 수신됐는지 검사한 후 악성 행위를 수행한다.

안랩은 지난해 10월, 자사 엔드포인트 탐지 & 대응 솔루션 AhnLab EDR을 활용해 BPFDoor 악성코드를 탐지 및 분석하고 해당 내용을 [ASEC 블로그](#)에 공개한 바 있다.



[그림 2] 2024년 10월 AhnLab EDR의 BPFDoor 탐지 내용 중 일부

분석 내용을 살펴보면, BPFDoor 최초 실행 시, 특정 명령을 통해 /dev/shm 경로에 kdmtmpflush라는 이름으로 자신을 복사한 후 자가 삭제한다. /dev/shm 경로는 리눅스의 메모리 기반 파일 시스템으로 애플리케이션이 임시 데이터를 저장 및 처리하는데 사용된다. 다만, 하지만 디스크에 기록되지 않고 메모리 상에서만 운영되는 특징이 있어 공격자들에 의해 자주 악용되곤 한다.

이후, [그림 3]의 문자열 중 하나를 선택해 이름을 변경하고 정상 프로세스로 위장한다. 이 때, prctl() 함수가 사용된다.

```
char *self[] = {
    "/sbin/udev -d",
    "/sbin/mingetty /dev/tty7",
    "/usr/sbin/console-kit-daemon --no-daemon",
    "hald-addon-acpi: listening on acpi kernel interface /proc/acpi/event",
    "dbus-daemon --system",
    "hald-runner",
    "pickup -l -t fifo -u",
    "avahi-daemon: chroot helper",
    "/sbin/auditd -n",
    "/usr/lib/systemd/systemd-journald"
};
```

[그림 3] 정상 프로세스 위장에 사용되는 문자열들

그리고 BPF 필터를 등록하고 대기한다. 이후 공격자가 매직 패킷이 포함된 명령을 보내면, BPF 필터로부터 이를 전달받아 공격을 위한 각각의 명령을 수행한다. 매직 패킷의 소스코드 기준, 비밀번호가 justforfun인 경우에는 매직 패킷에 포함된 IP/Port에 접속해 리버스 쉘(Reverse Shell)을 제공한다. 비밀번호가 socket인 경우에는 바인드 쉘(Bind Shell)을 제공해 새로운 포트를 열고 방화벽을 설정하여 공격자의 내부 연결을 확립한다. 마지막으로 비밀번호에 매칭되지 않

은 경우, 공격자에게 ‘1’을 응답한다. 이를 통해, 공격자는 악성코드의 성공적인 감염 여부를 판단할 수 있게 된다. 이러한 과정을 거쳐 공격자가 내부 환경과 연결되면 정보 유출 등의 악성 행위를 수행할 수 있다.

최근 통신사 공격과 유심 정보 유출에 사용되었던 악성코드도 바로 BPFDoor다. 해당 공격에 사용된 BPFDoor는 변종으로 위에 설명한 악성코드와 기능 측면에서 몇 가지 차이점이 있지만 큰 틀에서는 동일하게 작동한다. 이에 대한 자세한 내용은 안랩이 발간한 [BPFDoor 케이스스터디](#)를 통해 확인할 수 있다.

사례 2: 프록시 설치를 통한 시스템 무단 접속

2025년 2분기에는 리눅스 서버를 공격해 프록시를 설치하는 사례들이 확인되었다. 공격 사례들을 보면 프록시 도구 TinyProxy나 Sing-box를 설치했는데, 다른 공격 로그가 존재하지 않는 것으로 보아 공격자들의 목적은 감염 시스템을 프록시 노드로서 활용하기 위한 것으로 보인다.

공격자는 리눅스 서버 로그인에 성공한 뒤, Bash 악성코드를 다운로드해 프록시 도구 TinyProxy를 설치했다. 이후 TinyProxy의 설정 파일인 /etc/tinyproxy/tinyproxy.conf 또는 /etc/tinyproxy.conf에서 Allow와 Deny로 시작하는 접근 제어 규칙을 삭제하고 Allow 0.0.0.0/0 규칙을 추가했다. 해당 규칙이 적용되면 외부에서 제한 없이 접근이 가능하다. 공격자는 TinyProxy가 서비스하는 포트 8888번에 접근해 감염 시스템을 프록시로 악용할 수 있게 되었다.

```
200 #Allow 127.0.0.1
201 #Allow ::1
202 #Allow 192.168.0.0/16
203 #Allow 172.16.0.0/12
204 #Allow 10.0.0.0/8
205
325 #
326 #ReverseBaseURL "http://localhost:8888/"
327
328
329
330
331 # Added by script - WARNING: Allows all connections!
332 Allow 0.0.0.0/0
```

[그림 4] 주석 처리 및 삽입된 TinyProxy 설정

다음은 피해 시스템에 Sing-box라는 이름의 프록시 도구를 설치한 사례다. Sing-box는 다목적 프록시를 설치하는 도구로 vmess-argo, vless-reality, Hysteria2, TUICv5 프로토콜을 지원한다. 깃허브 설명에 따르면, 해당 도구를 설치하여

ChatGPT 및 Netflix의 차단을 해제할 수 있다. 본 사례에서는 공격자가 리눅스 시스템에 무단으로 접속하여 Sing-box를 설치했으며, 이를 통해 추가적인 불법 행위 혹은 금전적 수익을 노렸던 것으로 추정된다.

최근 리눅스 서버 공격 사례들을 보면, 프록시 기능을 담당하는 악성코드가 아닌 TinyProxy와 Sing-box처럼 정상적으로 사용할 수 있는 도구를 악용하는 추세다. 공격자는 감염 시스템을 프록시로 활용하여 또 다른 공격을 수행할 때 자신을 은폐할 수 있다. 또, 해당 프록시 노드에 대한 접근 권한을 판매하여 불법적인 수익을 올릴 수도 있다.

3. 리눅스 서버 보안을 위한 필수 솔루션은?

사이버 위협 고도화와 비즈니스 다변화가 빠르게 진행되는 가운데, 기업들이 운영하는 보안 솔루션도 많아지고 있다. 물론 많은 솔루션을 효과적으로 운영할 수 있다면 가장 좋지만, 현실적인 여력이 뒷받침되지 못하는 경우가 많다. 이 때, 성공적인 리눅스 서버 보안을 목표로 하는 기업들에게 꼭 필요한 솔루션을 꼽자면 안티바이러스, 샌드박스과 EDR, 그리고 MDR 서비스가 있다.

#1. 안티바이러스: 안티바이러스(Antivirus, AV)는 시그니처(Signature)와 행위 분석 등의 기술을 기반으로 악성코드 유입을 사전에 탐지해 차단한다. 엔드포인트 보안에 있어 가장 기초적이면서도 필수적인 솔루션이라 할 수 있다. AV 제품은 PC와 서버, 그리고 윈도우, 맥OS, 리눅스 등 각 환경에 맞춰 제공된다. 리눅스 서버 보안을 위해서는 해당 환경에 최적화된 AV 솔루션이 요구된다.

#2. 샌드박스: 샌드박스 솔루션은 여러 보안 구간에서 파일을 수집해 가상 환경(VM)에서 동적 분석을 수행한다. 예를 들면, 가상 환경에서 문서 파일을 실행하여 조작해 숨겨진 악성 행위 혹은 알려지지 않은 위협을 찾아낼 수 있다. 또, AV 솔루션을 우회하는 랜섬웨어도 진단한다. 시그니처를 기반으로 알려진 악성코드를 빠르게 진단하는 AV와는 상호보완적 관계를 갖는 것으로 이해하면 된다.

#3. EDR: EDR(Endpoint Detection & Response)은 엔드포인트에서 발생하는 모든 행위와 이벤트를 탐지 및 로깅(logging)하고 침해사고 조사에 필요한 정보를 상시적으로 수집하는 솔루션이다. 수집한 행위 정보를 바탕으로 위협을 능동적으로 추적·분석하며 장기적인 위협 대응 체계를 수립하는데 기여한다. 쉽게 비유하면 모든 것을 모니터링하는 CCTV와 같은 역할을 수행한다.

#4. MDR 서비스: MDR(Managed Detection & Response)은 보안 전문가들이 EDR을 활용한 위협 탐지, 분석 및 맞춤형 대응을 제공하는 서비스다. 이를 통해, 기업의 보안 운영 부담을 줄이고 탐지 & 대응 역량을 강화할 수 있다. 사이버 위협이 고도화되고 EDR 기반 탐지 및 분석에 상당한 전문성이 요구되는 만큼 MDR 서비스에 대한 수요와 관심도가 계속해서 높아지는 추세다.

이처럼, AV, 샌드박스, EDR을 올바르게 운영하고 MDR 서비스를 활용하면 다양한 위협에 대한 고도화된 대응 체계를 갖출 수 있어, 리눅스 서버 보안에 상당한 효과를 거둘 수 있다.

안랩은 지난 30년간 축적해온 기술력과 노하우를 바탕으로 위에 소개한 솔루션과 서비스가 유기적으로 시너지를 내는 업계 최고 수준의 오퍼링을 제공한다. 안티바이러스, 샌드박스, EDR, MDR 서비스를 아우르는 안랩의 오퍼링에 관한 자세한 내용은 다음화에서 살펴본다.

2편에서 계속

>> [\[리눅스 보안 시리즈\] 2화 - 안랩의 리눅스 서버 보안이 특별한 이유](#)

AhnLab

콘텐츠마케팅팀 신재만 과장