情報保護學會論文誌
第15券 第3號, 2005. 6

소스 레벨 리눅스 커널 취약점에 대한 특성 분류 및 상관성 분석*

고 광 선,[1†] 장 인 숙,[2] 강 용 혁,[3] 이 진 석,[2] 엄 영 익[1‡]
[1]성균관대학교, [2]국가보안기술연구소, [3]극동대학교

Characteristic Classification and Correlational Analysis of Source-level Vulnerabilities in Linux Kernel*

Kwangsun Ko,[1†] In-Sook Jang,[2] Yong-hyeog Kang,[3] Jin-Seok Lee,[2] Young Ik Eom[1‡]
[1]Sungkyunkwan University, [2]National Security Research Institute, [3]Far East University,

요 약

컴퓨터 운영체제가 가지는 취약점을 분석하고 분류하는 연구는 취약점을 이용한 익스플로잇을 방어할 수 있는 직접적인 보안기술에는 해당하지 않지만, 운영체제의 보안성 향상을 위한 보안기술 개발에 우선순위를 부여할 수 있다는 점에서 매우 중요한 연구 분야로 볼 수 있다. 그러나 최근 리눅스 운영체제에 대한 활용도에 비하여 리눅스 커널이 가지고 있는 취약점에 대한 연구는 몇몇 커뮤니티에서 운영하는 인터넷 사이트에 단순한 취약점 정보 및 분류기준만 제공되고 있을 뿐, 리눅스 커널이 가지는 근본적인 취약점에 대한 자세한 분석 작업은 수행되고 있지 않다. 따라서 본 논문에서는 1999년부터 2004년까지 6년 동안 SecurityFocus 사이트에 공개된 124개의 리눅스 커널 취약점에 대하여 커널 버전별로 몇 가지 항목을 기준으로 특성 분류 및 상관성 분석을 실시하고자 한다. 이러한 연구결과는 리눅스 커널 취약점을 악용한 익스플로잇의 공격 특성을 예측하고 주요 취약점이 커널 내부의 어떤 영역에서 발견되는지를 확인하는데 이용할 수 있다.

ABSTRACT

Although the studies on the analysis and classification of source-level vulnerabilities in operating systems are not direct and positive solutions to the exploits with which the host systems are attacked, It is important in that those studies can give elementary technologies in the development of security mechanisms. But, whereas Linux systems are widely used in Internet and intra-net environments recently, the information on the basic and fundamental vulnerabilities inherent in Linux systems has not been studied enough. In this paper, we propose characteristic classification and correlational analyses on the source-level vulnerabilities in Linux kernel that are opened to the public and listed in the SecurityFocus site for 6 years from 1999 to 2004. This study may contribute to expect the types of attacks, analyze the characteristics of the attacks abusing vulnerabilities, and verify the modules of the kernel that have critical vulnerabilities.

Keywords : *source-level vulnerability, Linux kernel*

접수일 : 2005년 2월 27일 ; 채택일 : 2005년 5월 13일
* 본 연구는 정보통신부 대학 IT연구센터 육성, 지원사업의 연구결과로 수행되었습니다.
† 주저자 : rilla91@ece.skku.ac.kr
‡ 교신저자 : yieom@ece.skku.ac.kr

Ⅰ. 서 론

컴퓨터 시스템이 사회경제적으로 중요한 위치를 차지함에 따라 해당 시스템 내에서 관리되는 정보를 보호하기 위하여 다양한 보안 기술이 개발되어 왔으며, 컴퓨터 시스템과 관련된 취약점을 분석하고 보완하는 연구는 1970년대부터 국외 대학과 연구소를 중심으로 꾸준히 진행되어 왔다[1,2]. 이 중에서 컴퓨터 운영체제가 가지는 취약점을 분류하는 연구는 취약점을 이용한 익스플로잇을 방어할 수 있는 직접적인 보안 기술에 해당되지는 않지만, 운영체제의 보안성 향상을 위한 보안기술 개발에 우선순위를 부여한다는 점에서 매우 중요한 연구 분야이다.

리눅스 운영체제는 1991년 9월에 커널 버전 0.01 발표 이후, 1.0(1994.3), 2.0(1996.6), 2.2 (1999.1), 2.4(2001.1), 2.6(2003.12)을 발표하면서 계속 발전하며 IT 산업에 사용되는 비중이 점차적으로 확대되어 가고 있다. 그러나 리눅스 운영체제에 대한 활용도에 비하여 리눅스 커널이 가지고 있는 취약점에 대한 연구는 몇몇 커뮤니티에서 운영하는 인터넷 사이트에 단순한 취약점 정보 및 분류기준만 제공되고 있을 뿐, 리눅스 커널이 가지는 근본적인 취약점에 대한 자세한 분석 작업은 수행되고 있지 않다.

본 논문에서는 1999년부터 2004년까지 6년 동안 SecurityFocus 사이트에 공개된 124개 리눅스 커널 취약점에 대하여 커널 버전별로 몇 가지 항목을 기준으로 특성 분류 및 상관성 분석을 실시하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 취약점 분석에 대한 기존 연구와 SecurityFocus 사이트에서 제시하고 있는 취약점 분석에 대해서 살펴보고, 3장에서는 본 논문에서 보여주고자 하는 리눅스 커널 취약점에 대한 특성 분류 내용을 소개하고 4장에서는 리눅스 커널 취약점에 대한 상관성 분석 내용을 소개한다. 마지막으로 5장에서는 결론 및 향후 연구계획을 제시한다.

Ⅱ. 관련 연구

본 장에서는 기존에 연구논문으로 발표된 취약점 분석 내용과 본 연구에서 참고자료로 인용한 SecurityFocus 사이트에서 제공하고 있는 취약점 분석 내용에 대해서 알아본다. 기존의 연구논문에서 발표된 취약점 분석 내용은 리눅스 운영체제만 국한되지 않고 컴퓨터 시스템 전반에 나타나는 취약점 분석 내용이고, SecurityFocus 사이트에서 제공하는 취약점 분석 내용은 리눅스 커널에만 국한된 내용이다.

1. 기존 취약점 분류법

기존에 컴퓨터 시스템 전반에 나타나는 취약점 분석 연구로는 Protection Analysis(PA) 분류법[3], The Research in Secured Operating Systems (RISOS)[3], Security 분류법[4], Brian Marick Survey[5], Chillarege's Orthogonal Defect Classification[6,7], Spafford 분류법[8], Landwehr 분류법[8], Aslam 분류법[9][10], Bishop 분류법[11], Du and Mathur 분류법[12], Jiwnani 분류법[13]이 있다. 이 중에서 대표적인 취약점 분석 및 분류 연구로는 C. E. Landwehr가 제안한 Landwehr 분류법[8], M. Bishop이 제안한 Bishop 분류법[11], 그리고 K. Jiwnani가 제안한 Jiwnani 분류법[3]이 있다.

1.1 Landwehr 분류법[8]

C. E. Landwehr는 컴퓨터 시스템의 하드웨어와 소프트웨어에 나타나는 50여개의 보안 취약점을 '취약점 발생 원인 (flaws by genesis)', '취약점 발생 시기 (flaws by time of introduction)', 그리고 '취약점 발생 위치 (flaws by location)'라는 세 가지 기준으로 분류하였다. '발생 원인'이란 "취약점이 어떻게 시스템에 들어왔는가?"를 의미하며, '발생 시기'는 '취약점이 언제 시스템에 들어왔는가?'를 의미한다. 그리고 '발생 위치'는 '취약점이 시스템의 어디에서 발견되었는가?'를 의미한다. 이 분류 결과는 보안 취약점이 어떻게 발생하며, 그 특성이 무엇인지에 대한 정보를 제공한다. C. E. Landwehr는 이 정보를 이용하여 취약점에 대한 보안을 강화하는 방법을 제시하였다.

1.2 Bishop 분류법[11]

M. Bishop은 유닉스 운영체제에 나타나는 취약점들을 '취약점의 성향 (nature of the flaw)', '발생 시기 (time of introduction)', '공격의 범위 (exploitation domain)', '영향의 범위 (effect domain)', '취약점을 악용하여 공격할 때 필요한 최소 구성 요소의 수 (minimum number of components needed to exploit the vulner-

ability)', 그리고 '식별하기 위한 근원 (source of identification)'이라는 6가지 기준으로 분류하였다. 이 기준에 따른 분류 결과는 첫째로 시스템 보안을 강화하기 위하여 취약점 분석 작업을 어떻게 할 것인가에 대한 정보를 제공하고, 둘째로 보안 취약점들을 최소화하는 프로그램을 작성하기 위한 정보를 제공하였다.

2. SecurityFocus 사이트[13]

SecurityFocus 사이트는 업체 중립적인 사이트로서 보안 전문가, 네트워크 관리자, 보안 컨설턴트, IT 매니저, CIO(Chief Information Officer), CSO(Chief Security Officer) 등에게 포괄적이고 신뢰성 있는 인터넷상의 보안 정보를 제공하는 곳이다. 이 사이트에서는 운영체제 전반에 적용할 수 있는 취약점 분류기준으로 12가지 기준을 제시하고 있다. 각 분류 기준 및 기준에 대한 자세한 내용은 표 1과 같다.

본 연구에서는 리눅스 커널이 가지는 취약점에 대한 분석 및 분류 내용을 제시하는 것이기에 표 1의 12가지 분류 기준 중에서 리눅스 커널에만 적용되는 경쟁조건 오류를 포함한 10개의 분류 기준(음영으로 표시된 부분)을 참고자료로 인용한다.

Ⅲ. 취약점 특성 분류

본 장에서는 1999년부터 2004년까지 6년 동안 SecurityFocus 사이트에 공개된 124개 리눅스 커널 취약점에 대한 특성 분류 내용을 설명하고자 한

표 1. SecurityFocus 사이트에서 제시하는 취약점 분류 기준

분류 기준	설 명
경쟁조건 오류 (Race condition error)	· 두 연산 사이의 시간 차이로 인하여 발생하는 오류
경계조건 오류 (Boundary condition error)	· 정해진 경계 주소나 데이터 구조를 벗어나서 읽기 혹은 쓰기를 함으로써 발생하는 오류 · 시스템 자원이 고갈되어 발생하는 오류 · 고정된 크기의 자료구조를 오버플로우 시켜 발생하는 오류 (버퍼 오버플로우)
접근확인 오류 (Access validation error)	· 인증되지 않은 객체에 대해 읽기 혹은 쓰기를 하여 발생하는 오류 · 접근 영역을 벗어난 객체에 대한 연산을 수행하여 발생하는 오류 · 접근 영역을 벗어난 파일 또는 장치를 읽거나 씀으로써 발생하는 오류 · 객체가 인증되지 않은 주체로부터의 입력을 받아 발생하는 오류 · 시스템이 주체를 정확하게 인증하지 못하여 발생하는 오류
직렬화 오류 (Serialization error)	· 부적절하거나 적합하지 않은 명령어의 직렬화으로 인하여 발생하는 오류
예외조건 처리 오류(Failure to handle exceptional conditions)	· 기능 모듈이나 장치, 사용자 입력으로 인해 발생한 예외 조건을 처리하는데 실패하는 경우
환경 오류 (Environment error)	· 특정 환경에서 기능적으로 정확한 모듈간 상호작용으로 발생하는 오류 · 특정 기계가 특정 설정 하에서 프로그램이 실행되어 발생하는 오류 · 설계된 연산환경과 다른 환경에서 사용되어 발생하는 오류
입력확인 오류 (Input validation error)	· 제품의 설계나 구조에 관한 오류로서 문제가 되는 구조를 고쳐야 근본적인 해결이 되는 오류 · 프로그램이 문맥상 부정확한 입력을 인식하지 못함으로써 발생하는 오류 · 모듈이 추가적인 입력 필드를 받아들임으로써 발생하는 오류 · 모듈이 누락된 입력 필드에 대한 처리를 하지 못함으로써 발생하는 오류 · 필드 값간 상호관계의 잘못 때문에 발생하는 오류
확인 오류 (Origin validation error)	· 정당하지 않은 사용자, 사용 절차, 사용 방법 등으로 인하여 발생하는 오류
기타 (Unknown)	· 분류 기준에 해당하지 않거나 명확한 분류가 어려운 오류
설계 오류 (Design error)	· 제품의 설계나 구조에 관한 오류로써 문제가 되는 구조를 고쳐야 근본적인 해결이 되는 오류
원자 오류 (Atomicity error)	· 부분적으로 수정된 자료구조를 다른 프로세스가 확인할 수 있는 오류 · 원자적으로 처리되어야 하는 연산 도중에 부분적으로 수정된 자료를 가진 상태로 코드가 종료됨으로써 나타나는 오류
설정 오류 (Configuration error)	· 시스템 유틸리티가 잘못된 설정 인자값으로 설치되어 발생하는 오류 · 잘못된 경로에 설치된 시스템 유틸리티를 익스플로잇하여 발생하는 오류 · 유틸리티에 접근 권한이 잘못 설정되어 보안 정책을 위배하는 오류

다. 특성 분류를 위한 기준으로는 'SecurityFocus 사이트의 분류 기준', '취약점 발생 영역', '시스템에 끼치는 영향', '익스플로잇 유무'이며, 각 기준에 대한 자세한 내용은 아래와 같다.

- SecurityFocus 사이트의 분류 기준: 해당 사이트에서 제시하는 기준으로 각 항목을 구성한다.
- 취약점 발생 영역: 리눅스 커널을 구성하는 주요 요소들과 대표적인 취약점 연구논문에서 제시하는 요소들을 참조하여 각 항목을 구성한다. 각 항목들로는 '네트워크 (network)', '디바이스 (device)', '메모리 (memory)', '시스템 콜 (system call)', '파일 (file)', '프로세스 (process)', '하드웨어 (hardware)', '기타 (etc.)' 이상 8개이다. (Landwehr 분류법의 '취약점 발생위치', Jiwnani 분류법의 '취약점 발생위치')
- 시스템에 끼치는 영향: 대표적인 취약점 연구논문에서 제시하는 요소들을 참조하여 각 항목을 구성한다. 각 항목들로는 '권한 상승 (privilege escalation)', '메모리 공개 (memory disclosure)', '서비스거부 (Denial of Service)', '시스템 고장 (system crash)', '정보 공개 (information leakage)', '기타 (etc.)' 이상 6개이다. (Bishop 분류법의 '영향의 범위', Jiwnani 분류법의 '취약점이 시스템에 미치는 영향')
- 익스플로잇 유무: 각 항목들로는 '존재 (O)', '존재하지 않음 (X)', '익스플로잇 필요 없음 (no exploit is required)', '개념 증명 (proof of concept)' 이상 4개이다.

위에서 명시한 항목들을 기준으로 취약점들의 특성을 분류하기 전에 2가지 조건을 전제로 한다. 첫 번째는 리눅스 커널 발표일을 기준으로 발표일 이전은 해당 리눅스 커널 버전에 해당하는 취약점이고, 발표일 이후는 다음 리눅스 커널 버전에 해당하는 취약점이다. 즉, 표 2에 보이는 리눅스 커널 버전별 발표일을 기준으로 2000년 1월 5일부터 2003년 12월 17일까지에 공개된 취약점들은 커널 버전 2.4에 해당하는 취약점으로 구분한다. 두 번째는 각 리눅스 커널 버전별로 구분된 취약점들은 누적된 취약점들을 의미하며, 누적된 취약점 수를 기반으로 분석 작업을 실시한다. 이는 취약점이 하나의 리눅스 커널 버전에만 국한되어 나타나지 않고 전체 버전에 모두 나타날 수 있으며, 새로운 커널 버전에 발표된 이후에도 이전 버전에 적용되는 취약점이 발견될 수 있기 때문이다.

표 2. 리눅스 커널 버전별 발표일

리눅스 커널 버전	발표일
2.6	2003.12.18
2.4	2000.1.4
2.2	1999.1.25
2.0	1996.6.9
1.0	1994.3.14

1. SecurityFocus 사이트의 분류 기준

SecurityFocus 사이트에서는 소프트웨어 관련 취약점 정보를 1995년부터 제공하고 있으며, 리눅스 커널에 해당하는 취약점 정보는 1999년부터 제공하고 있다. 리눅스 커널 취약점에 대해서는 표 3에서 보이는 바와 같이 132개를 제공하고 있으나 다수의 취약점들을 하나의 취약점으로 공개하는 취약점에 대해서는 제외할 필요가 있다. 이에 해당하는 취약점은 8개이며, 커널 버전 2.4에 3개(Linux Kernel 2.4.18 Security Issues-2002/08/21, Multiple Linux Kernel 2.4.18 Security Vulnerabilities-2002/10/17, Multiple Linux Kernel 2.2 Security Vulnerabilities-2002/10/22), 2.6에는 5개(Red Hat Linux 2.4 Kernel Multiple Potential Vulnerabilities-2003/12/23, Multiple Linux 2.4 Kernel Vulnerabilities-2004/01/15, Linux Kernel Multiple Unspecified Local Privilege Escalation Vulnerabilities-2004/07/22, Multiple Local Linux Kernel Vulnerabilities-2004/08/27, Linux Kernel Multiple Device Driver Vulnerabilities-2004/12/01)가 존재한다. 따라서 SecurityFocus 사이트에서 제공하는 132개 취약점 중에서 본 연구에 이용한 취약점 수는 124개이다.

표 3에서 주목할 만한 사항은 1999년에 공개된 취약점 수가 2004년을 제외한 다른 해보다 많다는 점과 2004년도에 공개된 취약점 수가 5배 이상 늘었다는 점이다. 이는 기존에 발견된 취약점 정보 11개를 1999년 6월 1일에 한꺼번에 등록하였기 때문이고, 최근에 보안 관련 종사자들이 리눅스 커널 취

표 3. SecurityFocus 사이트에서 제공하는 각 연도별 취약점 수

연도	개수	비고
2004	69	21개 취약점 공개 (2005.1.1 ~ 1.20)
2003	12	
2002	16	
2001	12	
2000	5	
1999	18	11개 취약점 공개 (1999.6.1)
합 계	132	

약점에 대한 관심이 커지고 있음을 의미한다. 또한 2005년 1월 20일까지 20일 동안 21개의 취약점 정보가 새로이 제공되고 있다는 사실로 미루어 보아 발견되는 리눅스 커널 취약점 수가 급격히 증가하고 있음을 알 수 있다. 표 3의 내용을 바탕으로 Security Focus 사이트에서 제시하는 취약점 분류 기준별 취약점 수 및 비율은 표 4와 같다.

표 4에서 보이는 바와 같이 전체 취약점의 약 80% 이상을 차지하는 분류 기준은 '설계 오류 (32.3%)', '예외조건 처리 오류 (20.2%)', '경계조건 오류 (14.5%)', 그리고 '기타 (14.5%)'이며, 이 중에서 약 50% 이상을 '설계 오류'와 '예외조건 처리 오류' 항목이 차지하고 있다. '설계 오류 (32.3%)'와 '기타 (14.5%)'가 높은 비율을 차지하는 이유는 리눅스가 다양한 종류의 하드웨어와 다수의 메커니즘을 지원하기 위하여 구현된 부분이 정확히 구현되지 않았음을 의미한다.

표 4. SecurityFocus 사이트에서 제시하는 취약점 분류 기준별 취약점 수 및 비율

분류 기준	취약점 수	비율 (%)
경쟁조건 오류	6	4.8
경계조건 오류	18	14.5
접근확인 오류	9	7.3
직렬화 오류	2	1.6
예외조건 처리 오류	25	20.2
환경 오류	2	1.6
입력확인 오류	3	2.4
확인 오류	1	0.8
설계 오류	40	32.3
기 타	18	14.5
합 계	124	100

2. 취약점 발생 영역

본 절에서는 '취약점 발생 영역' 기준으로 리눅스 커널 취약점을 분석한다. '취약점 발생 영역'이란 리눅스 커널을 구성하는 기본 요소인 '프로세스', '메모리', '파일', '디바이스', '네트워킹'[14]과 '시스템 콜', '하드웨어', '기타' 추가요소를 합쳐서 총 8가지 항목으로 구성한다. '시스템 콜' 항목을 추가한 이유는 리눅스 커널이 가지는 취약점을 악용하는 익스플로잇 코드가 실행되기 위해서는 사용자의 입력을 받아야 하고, 사용자 입력에 의해서 리눅스 커널이 영향을 받기 위해서는 즉, 사용자 영역 메모리에서 실행되고 있는 프로세스가 커널 영역 메모리에 악영향을 주기 위해서는 시스템 콜을 이용하기 때문이다. 또

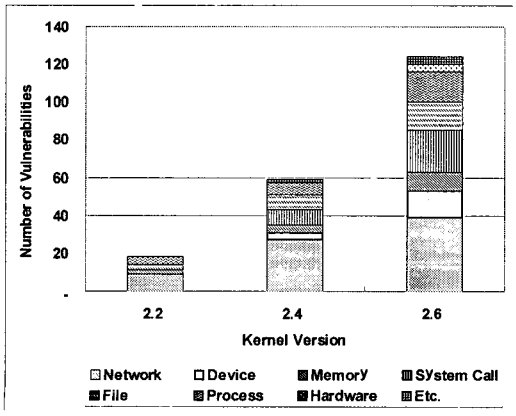

그림 1. 리눅스 커널 버전마다 발견된 '취약점 발생 영역'별 취약점 수

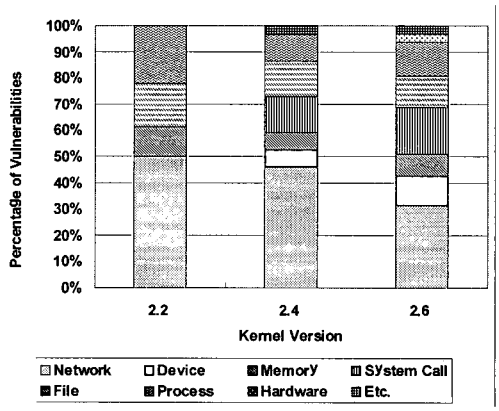

그림 2. 리눅스 커널 버전마다 '취약점 발생 영역'의 각 항목들이 차지하는 비율

한 '하드웨어' 항목을 추가한 이유는 최근에 리눅스 커널이 다양한 하드웨어를 지원하기 위해 구현된 하드웨어 의존적인 코드에서 리눅스 커널 버전에서 취약점이 발견되었기 때문이다. 또한 커널 내부 함수에서 나타나는 취약점은 시스템 콜 영역에 포함하였으며, '시그널', '특질 (capability)', 그리고 '자료구조'는 '프로세스' 영역에 포함하였다. 이러한 내용을 기준으로 각 리눅스 커널 버전마다 발견된 '취약점 발생 영역'별 취약점 수를 그림 1에서 보이고, '취약점 발생 영역'의 각 항목들이 차지하는 비율은 그림 2에서 보인다.

그림 1과 2에서 보이는 바와 같이 리눅스 커널 버전 2.2가 발표된 시점에는 대부분의 취약점이 '네트워크', '프로세스', '파일', 그리고 '메모리'에서 나타났으나, 이후 버전에서는 전체 취약점 수와 각 발생 영역별로 발생하는 취약점 수는 증가하면서 리눅스 커널 전반에 걸쳐서 취약점이 발생하는 것을 확인할 수 있다. '하드웨어' 항목의 경우 커널 버전 2.6에서 처음으로 취약점이 나타났으나 이후에 계속해서 증가할 것으로 보인다. 그 이유는 리눅스가 구현한 모든 하드웨어 의존적인 코드가 정확히 구현되었으리라고는 보지 않기 때문이다. 또한 리눅스 커널 버전 2.4에서 처음으로 '시스템 콜' 항목에 해당하는 취약점(15%)이 나타났으며, 커널 버전 2.6에서 취약점 수가 20%로 좀 더 증가하였음을 알 수 있다. 이는 현재 리눅스에서는 커널 버전 2.2부터 시스템 콜 매개 변수에 대한 검사를 가능한 마지막 순간까지, 즉 페이징 유닛이 선형 주소를 물리 주소로 바꾸는 순간까지 미루는 방식으로 동작하기에[15] 시스템 콜이 호출하는 커널 내부 함수가 가지는 취약점들이 발견되기 때문이다.

3. 시스템에 끼치는 영향

본 절에서는 리눅스 커널이 가지는 취약점을 이용하여 악의적인 사용자가 공격하였을 경우 '시스템에 끼치는 영향'을 기준으로 리눅스 커널 취약점을 분석한다. 시스템에 끼치는 영향은 '권한 상승 (privilege escalation)', '메모리 공개 (memory disclosure)', '서비스거부 (denial of service)', '시스템 고장 (system crash)', '정보 공개 (information leakage)', 그리고 '기타 (etc.)'와 같이 총 6개 항목으로 구분한다. '시스템 고장' 항목이 '서비스거부'에 포함되거나 '메모리 공개'와 '정보 공개'가 하나의 항목으로 합쳐질 수도 있지만, 보다 구체적으로 시스템에 끼치는 영향을 확인하기 위하여 구분하였다. 이러한 내용을 기준으로 각 리눅스 커널 버전마다 발견된 '시스템에 끼치는 영향'별 취약점 수를 그림 3에서 보이고, '시스템에 끼치는 영향'의 각 항목들이 차지하는 비율은 그림 4에서 보인다.

그림 3과 4에서 보이는 바와 같이 '권한 상승'과 '서비스거부' 항목은 커널 버전에 상관없이 전체 취약점 중에서 각각 20% 이내와 30% 이상을 차지하고 있으며, 리눅스 시스템에서 가장 많이 예상되는 공격 유형이다. 리눅스 커널 전반적으로 '메모리 공개' 항목에 해당하는 취약점이 낮은 비율을 차지하는 이유로 리눅스 커널이 가지는 프로세스 독립적인 가상메모리 관리 메커니즘에 의해서 효율적인 프로세

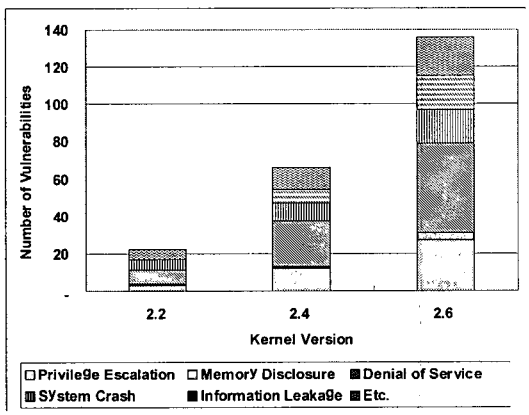

그림 3. 리눅스 커널 버전마다 발견된 '시스템에 끼치는 영향'별 취약점 수 (중복 허용)

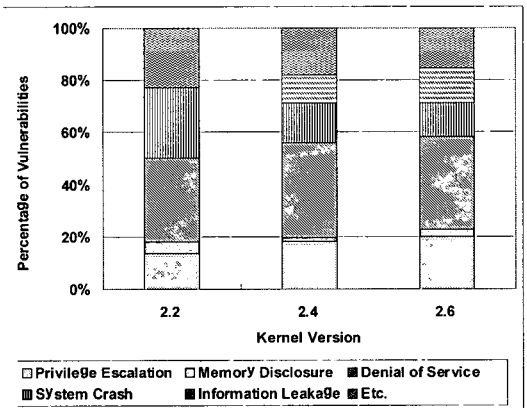

그림 4. 리눅스 커널 버전마다 '취약점 발생 영역'의 각 항목들이 차지하는 비율 (중복 허용)

스 메모리 관리가 실시되고 있다고 볼 수 있다. 그러나 커널 버전 2.4 이후 '정보 공개' 항목이 증가하는 것으로 보아 ptrace() 시스템 콜과 같은 메커니즘의 취약점을 이용하는 경우가 증가하고 있다고 볼 수 있다. '기타' 항목은 '버퍼 오버플로우', '하드웨어 접근', '잘못된 정보 제공', 그리고 '명확하지 않는 영향'에 해당하는 취약점을 포함한다.

4. 익스플로잇 유무

본 절에서는 리눅스 커널 취약점을 이용하는 '익스플로잇 유무'를 기준으로 취약점을 분석한다. '익스플로잇 유무'는 '존재' (O), '존재하지 않음' (X), '필요 없음' (no exploit is required), 그리고 '개념 증명' (proof of concept)와 같이 총 4개 항목으로 구분한다. '개념 증명' 항목에 해당하는 익스플로잇들은 취약점을 악용하여 시스템에 영향을 끼칠 수 있는 실질적인 익스플로잇 코드는 존재하지 않고 어떠한 방식으로 취약점을 공격할 수 있는지에 대한 개념 설명을 위한 익스플로잇 코드가 존재한다는 뜻이다. '필요 없음' 항목에 해당하는 취약점들은 대표적인 예로 비정상적인 네트워크 패킷을 이용하여 해당 취약점을 악용할 수 있는 경우와 같이 별도의 익스플로잇 코드가 필요 없는 경우를 의미한다. 이러한 내용을 기준으로 각 리눅스 커널 버전마다 발견된 '익스플로잇 유무'별 취약점 수를 그림 5에서 보이고, '익스플로잇 유무'의 각 항목들이 차지하는 비율은 그림 6에서 보인다.

그림 5와 6에서 보이는 바와 같이 공개되는 취약점 수는 상당한 증가율을 보이는 반면, 이러한 취약점을 악용하는 익스플로잇 코드는 다소 완만한 증가율을 보이는 것으로 보인다. 이러한 경향은 2가지 내용을 보여주고 있다. 첫 번째로 리눅스 커널이 다양한 기능을 제공하기 위하여 구현된 리눅스 커널의 많은 부분에서 취약점이 발견되고 있으나 이러한 취약점을 악용하는 익스플로잇 코드가 나오기까지에는 상당한 시간이 소요되고 있음을 보여주고 있으며, 두 번째로는 보안 커뮤니티를 중심으로 익스플로잇 코드가 제시되고 있으나 수많은 미숙련자들에 의해 악용되는 것을 방지하기 위하여 공개되는 익스플로잇 코드 수가 의도적으로 줄어들었다는 것을 보여주고 있다. 이러한 내용은 익스플로잇 코드의 개념만을 설명해 주는 '개념 증명' 항목이 커널 버전 2.4부터 나타나기 시작하는 것으로도 확인할 수 있다.

Ⅳ. 취약점들간의 상관성 분석

본 장에서는 취약점 특성에 대한 분류 결과를 기반으로 각각의 분석 기준으로 사용한 항목들 간의 연관 관계를 분석함으로써 리눅스 커널 취약점에 대한 상관성 분석 내용을 보인다. 취약점들간의 상관성 분석에 사용되는 기준은 앞에서 언급한 '취약점 발생 영역', '시스템에 끼치는 영향', 그리고 '익스플로잇 유무'라는 3가지 기준들을 이용한 {취약점 발생 영역, 시스템에 끼치는 영향}과 {익스플로잇 유무, 취약점 발생 영역} 쌍을 기반으로 분석한다.

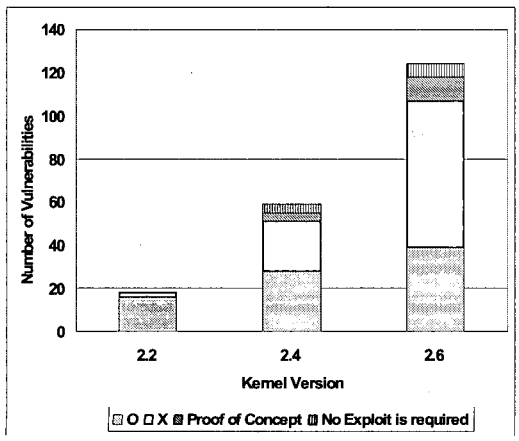

그림 5. 리눅스 커널 버전마다 발견된 '익스플로잇 유무'별 취약점 수

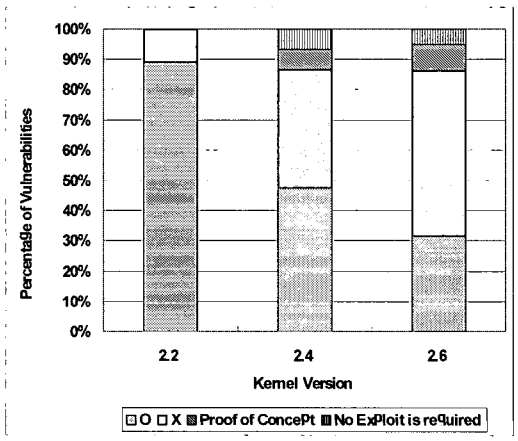

그림 6. 리눅스 커널 버전마다 '익스플로잇 유무'의 각 항목들이 차지하는 비율

1. '취약점 발생 영역'과 '시스템에 끼치는 영향'

본 절에서는 리눅스 커널 버전별로 {취약점 발생 영역, 시스템에 끼치는 영향}을 기준으로 취약점을 분석한다. 이러한 기준으로 분석함으로써 예상되는 내용은 권한 상승 공격이나 서비스거부 공격이 발생하였을 경우 가장 중점적으로 나타나는 취약 영역이 어디인지 확인하는 것이다. 리눅스 커널 버전마다 발견된 {취약점 발생 영역, 시스템에 끼치는 영향} 별 취약점 중에서 상위 3위 안에 해당하는 항목은 리눅스 커널 버전 2.2에서는 {네트워크, 서비스거부}, {네트워크 시스템 고장}, 그리고 {프로세스, 서비스거부} 항목, 2.4에서는 {네트워크, 서비스거부}, {네트워크, 정보 공개}, {디바이스, 권한 상승}, 그리고 {시스템 콜, 권한 상승} 항목, 2.6에서는 {네트워크, 서비스거부}, {디바이스, 권한 상승}, 그리고 {시스템 콜, 서비스거부} 항목이며, 이에 대한 자세한 내용은 표 5에서 보인다. 단, 리눅스 커널 버전 2.4에서는 {네트워크, 정보 공개}와 {시스템 콜, 권한 상승} 항목이 동일하게 4개의 취약점을 가지기 때문에 총 10개의 항목을 보인다.

표 5에서 보이는 바와 같이 리눅스 커널 버전에 상관없이 {네트워크, 서비스거부} 항목이 전체 10개 중 3개에 해당하며, 46개 취약점 중 16개로 34.8%를 차지한다. 또한 {디바이스, 권한 상승} 항목이 전체 10개 중에서 2개에 해당하며, 46개 취약점 중에서 9개로 19.6%를 차지한다. 이는 네트워크 영역에 나타나는 취약점들이 시스템에 끼치는 영향은 주로 서비스거부와 관련되어 있고, 디바이스 영역에 나타나는 취약점들이 시스템에 끼치는 영향은 주로 권한 상승과 관련되어 있음을 알 수 있다.

표 5. 리눅스 커널 버전마다 발견된 {취약점 발생 영역, 시스템에 끼치는 영향}별 취약점 중에서 상위 3위 안에 해당하는 항목들에 해당하는 취약점 수 (중복 허용)

버전	취약점 발생 영역	시스템에 끼치는 영향	취약점 수 (중복된 수)
2.2	Network	Denial of Service	3
		System Crash	4 (1)
	Process	Denial of Service	2 (1)
2.4	Network	Denial of Service	6
		Information Leakage	4
	Device	Privilege Escalation	5 (3)
	System Call	Privilege Escalation	4
2.6	Network	Denial of Service	7
	Device	Privilege Escalation	4 (1)
	System Call	Denial of Service	7

2. '익스플로잇 유무'와 '취약점 발생 영역'

본 절에서는 리눅스 커널 버전별로 {익스플로잇 유무, 취약점 발생 영역}을 기준으로 취약점을 분석한다. 이러한 기준으로 분석함으로써 예상되는 내용은 취약점을 악용하는 익스플로잇 코드가 어떠한 영역에 분포하고 있는지를 확인하는 것이다. 리눅스 커널 버전마다 발견된 {익스플로잇 유무, 취약점 발생 영역}별 취약점 중에서 상위 3위 안에 해당하는 항목은 리눅스 커널 버전 2.2에서는 {존재, 네트워크}, {존재, 파일}, 그리고 {존재, 프로세스} 항목, 2.4에서는 {존재, 시스템 콜}, {존재하지 않음, 네트워크}, 그리고 {존재하지 않음, 디바이스} 항목, 2.6에서는 {존재하지 않음, 디바이스}, {존재하지 않음, 시스템 콜}, {존재하지 않음, 파일}, 그리고 {존재하지 않음, 프로세스} 항목이다. 이에 대한 자세한 내용은 표 6에서 보인다. 단, 리눅스 커널 버전 2.6에서는 {존재하지 않음, 파일}과 {존재하지 않음, 프로세스} 항목이 동일하게 7개의 취약점을 가지기 때문에 총 10개의 항목을 보인다.

표 6. 리눅스 커널 버전마다 발견된 {익스플로잇 유무, 취약점 발생 영역}별 취약점 중에서 상위 3위 안에 해당하는 항목들에 해당하는 취약점 수

버전	익스플로잇 유무	취약점 발생 영역	취약점 수
2.2	O	Network	8
		File	3
		Process	3
2.4	O	System Call	5
	X	Network	11
		Device	4
2.6	X	Device	10
		System Call	10
		File	7
		Process	7

표 6에서 보이는 바와 같이 리눅스 커널 버전이 2.6으로 갈수록 취약점을 악용하는 익스플로잇이 존재하는 비율이 낮아진다. 커널 버전 2.2에서는 14건, 100%로 익스플로잇이 존재하지만, 커널 버전 2.4에서는 20건 중에서 5건, 25%만 익스플로잇이 존재하고, 커널 버전 2.6에서는 34건 중에서는 모든 취약점에 익스플로잇이 존재하지 않는다. 이러한 경향은 취약점 특성에 대한 분류에서도 설명한 것과 같이 익스플로잇 코드가 나오기까지에 상당한 시간이 소요되고 있으며, 많은 수의 미숙련자들에 의해 악용되는 것을 방지하기 위하여 공개되는 익스플로잇 코드 수가 의도적으로 줄어들고 있다는 내용을 보여주고 있다. 또한 익스플로잇 코드가 존재하지 않는 취약점들 49개 중 15개(약 28.6%)가 리눅스 커널의 '디바이스' 영역을 차지하는 것으로 보아 디바이스 관련 리눅스 커널 소스의 취약점 발생 비율이 커널의 다른 영역보다 높음을 간접적으로 확인할 수 있다.

V. 결 론

본 논문에서는 SecurityFocus 사이트에서 1999년부터 2004년까지 6년 동안 제공하고 있는 124개의 리눅스 커널 취약점에 대해서 특성 분류와 상관성 분석을 실시하였다. 취약점 특성에 대한 분류에서는 '취약점 발생 영역', '시스템에 끼치는 영향', 그리고 '익스플로잇 유무' 기준을 분석하였으며, 취약점들간의 상관성 분석에서는 {취약점 발생 영역, 시스템에 끼치는 영향}과 {익스플로잇 유무, 취약점 발생 영역} 기준을 이용하였다. '취약점 발생 영역' 기준 특성 분류에서는 커널 버전이 높아질 수록 점차적으로 리눅스 커널 전반에 걸쳐 취약점이 발생하는 것을 확인할 수 있었고, '시스템에 끼치는 영향' 기준 특성 분류에서는 '서비스거부' 항목에 해당하는 취약점들이 커널 버전에 상관없이 30% 이상을 차지하고 있음을 알 수 있었다. 또한 커널 버전 2.4 이후 '정보 공개' 항목이 증가하는 것으로 보아 ptrace() 시스템 콜과 같은 메커니즘의 취약점을 이용하는 경우가 증가하고 있다고 볼 수 있다. 마지막으로 '익스플로잇 유무' 기준 특성 분류에서는 리눅스 커널 버전이 증가하면서 취약점 발견 건수가 증가하고 있으나 이러한 취약점을 악용하는 익스플로잇 코드가 나오기까지는 상당한 시간이 소요되고 보안 커뮤니티에서 의도적으로 익스플로잇 코드를 공개하지 않다는 것을 알 수 있었다. {취약점 발생 영역, 시스템에 끼치는 영향} 기준 상관성 분석에서는 네트워크 영역에 나타나는 취약점들이 시스템에 끼치는 영향은 주로 서비스거부와 관련되어 있고, 디바이스 영역에 나타나는 취약점들이 시스템에 끼치는 영향은 주로 권한 상승과 관련되어 있음을 알 수 있다. 그리고 {익스플로잇 유무, 취약점 발생 영역} 기준 상관성 분석에서는 리눅스 시스템의 활용도가 증가함에 따라 리눅스 커널에서 발견되는 취약점의 수도 증가하지만, 취약점을 악용할 수 있는 익스플로잇의 수는 발견되는 취약점 증가율만큼 증가하지 않는다는 것을 알 수 있으며, 익스플로잇 코드가 존재하지 않는 취약점들 중 약 28.6%가 리눅스 커널의 '디바이스' 영역에서 나타나고 있음을 알 수 있다.

이와 같은 연구는 리눅스 커널의 취약점을 악용하는 익스플로잇 코드의 공격 특성과 주요 취약점이 커널의 어떠한 영역에서 발견되는지를 확인하는데 이용할 수 있으며, 익스플로잇 코드 존재 유무에 따른 리눅스 커널 취약점 발생 영역을 분석함으로써 커널 보안 패치 및 보안성 향상 메커니즘 개발을 위한 연구에 기여할 수 있을 것이다.

참 고 문 헌

[1] B. Marick, "A survey of software fault surveys," Technical Report UIUCDCS- R90-1651, University of Illinois at Urbana-Chamaign, Dec. 1990.

[2] 박태규, 임연호, "커널 기반의 보안 리눅스 운영체제 구현," 한국정보보호학회, 정보보호학회 논문지, Vol. 11, No. 4, Aug. 2001.

[3] K. Jiwnani and M. Zelkowitz, "Maintaining Software with a Security Perspective," *International Conference on Software Maintenance (ICSM'02)*, Montreal, Quebec, Canada, Oct. 03-06, 2002.

[4] http://www.garlic.com/~lynn/secure.htm.

[5] B. Marick, "A survey of software fault surveys", Technical Report UIUCDCS -R- 90-1651, University of Illinois at Urbana-Champaign, Dec. 1990.

[6] R. Chillarege, "ODC for Process Measurement, Analysis and Control," *Proc. of the Fourth International Conference on Software Quality*, ASQC Software Division, McLean, VA, USA, Oct. 3-5, 1994.

[7] R. Chillarege, I. S. Bhandari, J. K. Chaar, M. J. Halliday, D. S. Moebus, B. K. Ray, Man-Yuen Wong, "Orthogonal Defect Classification - A Concept for In-Process Measurements," *IEEE Transactions on Software Engineering*, Vol. 18, No. 11, Nov. 1992.

[8] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi, "A Taxonomy of Computer Program Security Flaws," *ACM Computing Surveys*, Vol. 26, No. 3, 1994.

[9] T. Aslam, "A taxonomy of Security Faults in the Unix Operating System," M.S. Thesis, Purdue University, 1995.

[10] T. Aslam, "Use of a taxonomy of Security Faults," Technical Report 96-05, COAST Laboratory, Department of Computer Science, Purdue University, Mar. 1996.

[11] M. Bishop, "A Taxonomy of UNIX System and Network Vulnerabilities," Technical Report CSE-95-10, Purdue University, May 1995.

[12] W. Du and A. P. Mathur, "Categorization of Software Errors that led to Security Breaches," *Proc. of the 21st National Information Systems Security Conference (NISSC'98)*, Crystal City, VA, USA, 1998.

[13] http://www.securityfocus.com

[14] A. Rubini and J. Corbet, *Linux Device Drivers 2nd Ed.*, O'REILLY.

[15] D. P. Bovet and M. Cesati, *Understanding the Linux Kernel 2nd Ed.*, O'REILLY.

〈著者紹介〉

고 광 선 (Kwangsun Ko) 학생회원
1998년 2월: 성균관대학교 정보공학과 졸업
2004년 8월: 성균관대학교 전기전자및컴퓨터공학부 석사
2004년 9월~현재: 성균관대학교 컴퓨터공학과 박사과정
〈관심분야〉 정보보호, 리눅스, 네트워크

장 인 숙 (In-Sook Jang)
1998년 2월: 경북대학교 문헌정보학과 졸업
2001년 2월: 경북대학교 컴퓨터과학과 석사
2001년 3월~현재: 한국전자통신연구원 부설 국가보안기술연구소 연구원
〈관심분야〉 시스템 보안, 운영체제, 정보보호, 데이터베이스

강 용 혁 (Yong-hyeog Kang)
1996년 2월: 성균관대학교 정보공학과 졸업
1998년 2월: 성균관대학교 전기전자및컴퓨터공학과 석사
2004년 8월: 성균관대학교 전기전자및컴퓨터공학과 박사
2004년 3월~현재: 극동대학교 경영학부 전자상거래학과 전임강사
〈관심분야〉 전자상거래, 시스템 보안, 네트워크 보안, 리눅스

이 진 석 (Jin-Seok Lee)
1986년 2월: 대전산업대학교 전자계산학과
1990년 2월: 한남대학교 수학과 석사
2000년 8월: 한남대학교 컴퓨터공학과 박사
1986년 1월~1999년 12월: 한국전자통신연구원 선임연구원
2000년 1월~현재: 한국전자통신연구원 부설 국가보안기술연구소 책임연구원
〈관심분야〉 정보보호, 시스템 및 네트워크 보안, 정보전

엄 영 익 (Young Ik Eom)
1983년 2월: 서울대학교 계산통계학과 졸업
1985년 2월: 서울대학교 전산과학과 석사
1991년 8월: 서울대학교 전산과학과 박사
2000년 9월~2001년 8월: Dept. of Info. and Comm. Science at UCI 방문교수
1993년 3월~현재: 성균관대학교 정보통신공학부 교수
〈관심분야〉 분산 컴퓨팅, 이동 컴퓨팅, 이동 에이전트, 시스템 보안, 운영체제, 내장형 시스템