Characteristic Classification and Correlation Analysis of Source-Level Vulnerabilities in the Linux Kernel*

Kwangsun Ko[1], Insook Jang[2], Yong-hyeog Kang[3], Jinseok Lee[2], and Young Ik Eom[1,**]

[1] School of Information and Communication Eng., Sungkyunkwan University, 300 Cheoncheon-dong, Jangan-gu, Suwon, Gyeonggi-do 440-746, Korea
{rilla91, yieom}@ece.skku.ac.kr
[2] National Security Research Institute, 161 Gajeong-dong, Yuseong-gu, Daejeon 305-700, Korea
{jis, jinslee}@etri.re.kr
[3] School of Business Administration, Far East University, 5 San Wangjang, Gamgok, Eumseong, Chungbuk 369-851, Korea
yhkang@mail.kdu.ac.kr

Abstract. Although studies regarding the classification and analysis of source-level vulnerabilities in operating systems are not direct and practical solutions to the exploits with which computer systems are attacked, it is important that these studies supply the elementary technology for the development of effective security mechanisms. Linux systems are widely used on the Internet and in intra-net environments. However, researches regarding the fundamental vulnerabilities in the Linux kernel have not been satisfactorily conducted. In this paper, characteristic classification and correlation analysis of source-level vulnerabilities in the Linux kernel, open to the public and listed on the SecurityFocus site for the 6 years from 1999 to 2004, are presented. This study will enable Linux kernel maintenance groups to understand the wide array of vulnerabilities, to analyze the characteristics of the attack abusing vulnerabilities, and to prioritize their development effort according to the impact of these vulnerabilities on the Linux systems.

1 Introduction

There have been ongoing studies conducted by academics and institutes on the classification and analysis of hardware and software vulnerabilities in computer systems since the 1970s. Although these these studies are not direct and practical solutions to the exploits with which computer systems are attacked, it is

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).
** Corresponding author.

Y. Hao et al. (Eds.): CIS 2005, Part II, LNAI 3802, pp. 1149–1156, 2005.

important that these studies supply the elementary technology for the development of effective security mechanisms. Software containing a minor security flaw can expose a secure environment and make a system vulnerable to attacks [1][2].

There are a variety of reasons in the existence of security vulnerabilities in computer systems: incorrect implementation, improper configuration and initialization, design errors, no verification of parameters, abusive system calls, and so on. Surely as the social concerns of security vulnerabilities increases, computer systems must be designed to be increasingly secure. In 1991, after the first version of the Linux kernel distributed out by Linus Torvalds, Linux systems are widely used on Internet and in intra-net environments. However, researches regarding the fundamental vulnerabilities inherent in the Linux kernel have not been thoroughly conducted. In this paper, characteristic classification and correlation analysis of 124 source-level vulnerabilities in the Linux kernel, open to the public and listed in the SecurityFocus site for the 6 years from 1999 to 2004, are presented according to Linux kernel versions.

The subsequent sections of this paper are organized as follows. Section 2 presents related studies that have been conducted, in order to classify and analyze vulnerabilities in computer systems. In Section 3 and 4, characteristic classification and correlation analysis of source-level vulnerabilities in the Linux kernel are detailed, respectively. Section 5 concludes this paper.

2 Related Works

Several studies exist regarding vulnerabilities in computer systems: the Research In Secured Operating Systems (RISOS) [2], Security taxonomy [3], Chillarege's Orthogonal Defect Classification [4][5], Spafford's taxonomy [6], Landwehr's taxonomy [6], Bishop's taxonomy [7], Du and Mathur's taxonomy [8], and Jiwnani's taxonomy [2]. In addition, there are many Internet sites that have classified or analyzed the vulnerabilities of computer systems: the SecurityFocus site [9], the Common Vulnerabilities and Exposures (CVE) site [10], the LinuxSecurity site [11], and the iSEC Security Research site [12].

In this paper, necessary information regarding characteristic classification and correlation analysis is obtained from the SecurityFocus site for the 6 years from 1999 to 2004, where source-level vulnerabilities in the Linux kernel are very well defined and categorized among others.

3 Characteristic Classification

The characteristic classification of source-level vulnerabilities in the Linux kernel are done as follows:

- ***Location in the Linux kernel.*** This consists of 8 criteria: 'network', 'device', 'memory', 'system call', 'file', 'process', 'hardware', and 'etc.' according to where vulnerabilities are detected. (Software functionalities of the Linux kernel, 'location' of [6], and 'location of flaws in the system' of [2] are referred to in this paper.)

- ***Impact on the system.*** This consists of 6 criteria: 'privilege escalation', 'memory disclosure', 'denial of service', 'system crash', 'information leakage', and 'etc.' according to how vulnerabilities impact on the system. ('effect domain' of [7] and 'impact of flaws on the system' of [2] are referred to in this paper.)
- ***Existence of exploits.*** This consists of 4 criteria: 'O (there are exploits)', 'X (there is not a exploit)', 'proof of concept (just showing how an exploit code processes)', and 'no exploit is required (e.g. sending abnormal network packets is enough to exploit a vulnerability)' according to the existence of exploits. ([9] is referred to in this paper.)

There are 132 vulnerabilities from the SecurityFocus site for the 6 years from 1999 to 2004: 18 vulnerabilities in 1999; 5 in 2000; 12 in 2001; 16 in 2002; 12 in 2003; and 69 in 2004. Some are, however, also excluded for accuracy. Vulnerabilities like a 'Multiple Local Linux Kernel Vulnerabilities (Aug. 27, 2004)' can not be accepted as raw data of our classification and analysis because of ambiguousness. There are 3 and 5 vulnerabilities in Linux 2.4 kernel and 2.6, respectively. Therefore, 124 vulnerabilities are used for characteristic classification in this paper. For the reference, the reason that the number of vulnerabilities in 1999 are higher than all other years except 2004 is because 11 pre-existing vulnerabilities were simultaneously made public on Jun. 1, 1999.

Based on the SecurityFocus' taxonomy, the numbers and percentages of vulnerabilities are presented in Table 1.

Table 1. The numbers and percentages of vulnerabilities (SecurityFocus' taxonomy)

Criteria	Number	Percentage
Race condition error	6	4.8
Boundary condition error	18	14.5
Access validation error	9	7.3
Serialization error	2	1.6
Failure to handle exceptional conditions	25	20.2
Environment error	2	1.6
Input validation error	3	2.4
Origin validation error	1	0.8
Design error	40	32.3
Unknown	18	14.5
Total	124	100

Table 1 shows that almost 80% of total vulnerabilities are related to 4 criteria: 'Design error' (32.3%), 'Failure to handle exceptional conditions' (20.2%), 'Boundary condition error' (14.5%), and 'Unknown' (14.5%). More than 50% are related to the criteria of 'Design error' and 'Failure to handle exceptional condition'. The reason that the criteria of 'Design error' and 'Unknown' occupy relative much portion of total vulnerabilities is because many functions and mechanisms are incorrectly implemented, supporting a variety of hardware

products and faculties. This fact means that as Linux systems play important roles in IT, Linux kernel maintenance groups intensively fix any 'Design error' and 'Unknown' issues as quickly as possible.

Additionally, there are two considerations. One is the division of vulnerabilities according to Linux kernel versions. Vulnerabilities found prior to the public date of a specific Linux X kernel are associated with the prior version of X. For example, vulnerabilities from Jan. 4, 2000 to Dec. 17, 2003 are associated with Linux 2.4 kernel; Linux 2.6 kernel was released, Dec. 18, 2003. The other is that the sums of vulnerabilities according to Linux kernel versions are accumulated because a number of vulnerabilities do not belong to a only specific Linux kernel version.

3.1 Location in the Linux Kernel

Based on the 'location in the Linux kernel', the numbers of vulnerabilities are presented in Table 2. Vulnerabilities related to inner kernel functions are included in 'System call' criterion, and those related to 'Signal', 'Capability', and 'Structure' are included in 'Process' criterion.

Table 2. The numbers of vulnerabilities (location in the Linux kernel)

Version	Network	Device	Memory	System call	File	Process	Hardware	Etc.
2.2	9	-	2	-	3	4	-	-
2.4	27	4	4	8	8	6	-	2
2.6	39	14	10	22	15	16	4	4

Table 2 show that most vulnerabilities are included under 4 criteria in Linux 2.2 kernel: 'Network', 'Process', 'File', and 'Memory'. However in the posterior Linux kernel versions, those are spread over the entire Linux kernel. It is indirectly confirmed that many vulnerabilities related to the main functionalities in the Linux kernel have been fixed and that Linux systems are becoming increasingly stable in all major faculties. The 'Hardware' criterion firstly appears in Linux 2.6 kernel and will increase continuously because a number of hardware-dependent kernel codes supporting various hardware platforms may be incorrectly implemented. The vulnerabilities of 'System call' criterion which is 13.6% appear in Linux 2.4 kernel and increase more than 17.7% in 2.6. This fact means that no checking the arguments of system calls whether the arguments are correct until the completion of translation from linear to physical address [14] has problems. Additionally, we predict that the vulnerable probability of system calls becomes higher.

3.2 Impact on the System

Based on the 'impact on the system', the numbers of vulnerabilities are presented in Table 3. The criterion of 'etc.' include the 4 criteria: 'buffer overflow', 'hardware access', 'providing the wrong information', and 'implicit effects' criterion.

Table 3. The numbers of vulnerabilities (impact on the system)

Version	Privilege escalation	Memory disclosure	Denial of service	System crash	Information leakage	Etc.
2.2	3	1	7	6	-	5
2.4	12	1	24	10	7	12
2.6	27	4	48	18	18	21

Table 3 show that 'privilege escalation' and 'denial of service' are less than 20% and more than 30%, respectively. These criteria are mainly used when attackers exploit Linux systems. The percentage of 'memory disclosure' is low in all versions, and this may prove that the virtual memory management mechanism in the Linux kernel is very effective. The reason that the number of vulnerabilities in the 'information leakage' criterion increases in posterior to Linux 2.4 kernel is because tracing system calls (e.g. *ptrace()* system call) or the */proc* filesystem are frequently used in order to support many new mechanisms.

3.3 Existence of Exploits

Based on the 'existence of exploits', the numbers of vulnerabilities are presented in Table 4.

Table 4. The number of vulnerabilities (existence of exploits)

Version	O	X	Proof of concept	No exploit required
2.2	16	2	-	-
2.4	28	23	4	4
2.6	39	68	11	6

Table 4 show that the ratio of public vulnerabilities sharply increases while that of the corresponding exploit codes slowly increase. This fact presents that the exploit codes related to vulnerabilities take considerable time to be open. The advent of exploit codes also intentionally decreases in order to prevent unskilled people from easily abusing exploit codes. This is confirmed by the result of the 'proof of concept' criterion.

4 Correlation Analysis

In this section, the results of a correlation analysis are presented based on the characteristic classification above. There are two criteria: (location in the Linux kernel, impact on the system) and (existence of exploits, location in the Linux kernel), are briefly (location, impact), (existence, location), respectively.

4.1 Location and Impact

Based on the (location, impact), the numbers of vulnerabilities are presented in Table 5.

Table 5. The numbers of vulnerabilities in high-ranked criteria of (location, impact)

Version	Location	Impact	Number (duplication)
2.2	Network	Denial of service	3
	Network	System crash	4 (1)
	Process	Denial of service	2 (1)
2.4	Network	Denial of service	6
	Network	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

Table 5 shows that the number of (network, denial of service) and (device, privilege escalation) criteria are 16/46 (34.8%) and 9/46 (19.6%) in all Linux kernel versions, respectively. This shows that the impact of vulnerabilities corresponding to the location of the 'network' relates to 'denial of service' and that 'device' relates to 'privilege escalation'. That is, when attackers aim to make the services of a target system helpless against an attack, attackers usually abuse the vulnerabilities related to 'network' faculties. When attackers aim to be an abnormal administrator, attackers usually abuse vulnerabilities relating to 'device'. Needless to say, device drivers are not as susceptible to exploits as other parts of OS and that there seems to be a large lag between discovery of vulnerabilities and actual exploit code [15][16]. No regarding of the vulnerabilities corresponding to device drivers is not useful because device drivers in Linux run as kernel modules and the source code size of the */dev* directory is very larger than others.

4.2 Existence and Location

Based on the (existence, location), the numbers of vulnerabilities are presented in Table 6.

Table 6. The number of vulnerabilities in high-ranked criteria of (existence, location)

Version	Existence	Location	Number
2.2	O	Network	8
	O	File	3
	O	Process	3
2.4	O	System call	5
	X	Network	11
	X	Device	4
2.6	X	Device	10
	X	System call	10
	X	File	7
	X	Process	7

Table 6 shows that the ratio of 'O' clearly decreases from Linux 2.2 kernel to 2.6. The frequency of 'O' is 14/14 (100%) in Linux 2.2 kernel; 5/20 (25%) in 2.4; and 0/34 (0%) in 2.6. This means that the advent of exploit codes to public vulnerabilities take much time or the number of public exploit codes intentionally decreases in order to prevent unskilled people from easily abusing exploit codes. In all Linux kernel versions, about 14/49 (28.6%) of 'X' are in the 'device' criterion of location. This fact is somewhat different from the result of the characteristic classification of 'location in the Linux kernel'; the ratio of vulnerabilities is low. That is, there are a number of source-level vulnerabilities related to 'device' criterion, however, the exploitable probability of vulnerabilities is low. It is indirectly proved that when an attacker wants to exploit vulnerabilities of a target system, he or she has a problem with which source codes of 'device' criterion are chosen because the target system has many source codes in order to support a variety of hundreds of thousands of hardware products.

5 Conclusions

In this paper, characteristic classification and correlation analysis of source-level vulnerabilities in the Linux kernel, open to the public and listed on the SecurityFocus site for the 6 years from 1999 to 2004, are presented. According to characteristic classification, most vulnerabilities in Linux 2.2 kernel are detected in the criteria of 'Network', 'Process', 'File', and 'Memory', however, vulnerabilities in posterior versions are spread over the entire criteria of the Linux kernel. Based on 'impact on the system', the criteria of 'privilege escalation' and 'denial of service' occupy less than 20% and more than 30% of total vulnerabilities, respectively. Additionally, the ratio of public vulnerabilities sharply increases in whole Linux kernel versions, however that of exploit codes related to public vulnerabilities slowly increases. According to correlation analysis, when attackers aim to render target systems denial of service, they mainly exploit vulnerabilities related to 'network' faculties while when attackers want to be abnormal administrators, they mainly abuse vulnerabilities related to 'device'. There are a number of source-level vulnerabilities related to 'device' criterion, however, the exploitable probability of vulnerabilities is low.

This study will enable Linux kernel maintenance groups to understand the wide array of vulnerabilities, to analyze the characteristics of the attack abusing vulnerabilities, and to prioritize their development effort according to the impact of these vulnerabilities on the Linux systems.

References

1. B. Marick: A survey of software fault surveys, Technical Report UIUCDCS-R90-1651, University of Illinois at Urbana-Chamaign, Dec. (1990)
2. K. Jiwnani and M. Zelkowitz: Maintaining Software with a Security Perspective, International Conference on Software Maintenance (ICSM'02), Montreal, Quebec, Canada (2002)

3. Security Taxonomy, http://www.garlic.com/ lynn/secure.htm.
4. R. Chillarege: ODC for Process Measurement, Analysis and Control, Proc. of the Foruth International Conference on Software Quality, ASQC Software Division, McLean, VA, (1994)3-5
5. R. Chillarege, I. S. Bhandari, J. K. Chaar, M. J. Halliday, D. S. Moebus, B. K. Ray, and Man-Yuen Wong: Orthogonal Defect Classification - A Concept for In-Process Measurements, IEEE Transactions on Software Engineering, 18(1992)
6. C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi: A Taxonomy of Computer Program Security Flaws, ACM Computing Surveys, 26(1994)
7. M. Bishop: A Taxonomy of UNIX System and Network Vulnerabilities, Technical Report CSE-95-10, Purdue University (1995)
8. 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 (1998)
9. SecurityFocus, http://www.securityfocus.com.
10. Common Vulnerabilities and Exposures, the Standard for Information Security Vulnerability Names, http://www.cve.mitre.org.
11. Guardian Digital: Inc., http://www.linuxsecurity.com.
12. iSEC Security Research, http://www.isec.pl.
13. A. Rubini and J. Corbet, Linux Device Drivers 2nd Ed., O'REILLY (2001)
14. D. P. Bovet and M. Cesati, Understanding the Linux Kernel 2nd Ed., O'REILLY (2003)
15. E. Rescorla: Security Holes Who cares?, Proc. Of the 12the USENIX Security Symposium, Washington D.C. (2003)
16. H. Browne, W. Arbuagh, J. McHugh, and W. Fiothen: A Trend Analysis of Exploitations, In IEEE Symposium on Security and Privacy (2001)