JANT 한국항행학회논문지
Journal of Advanced Navigation Technology

기반 기술

J. Adv. Navig. Technol. 20(1): 72-78, Feb. 2016

리눅스에 적용된 해시 및 암호화 알고리즘 분석

Analysis of the Hashing and Encryption Algorithms Applied to the Linux

배 유 미[1] · **정 성 재**[1] · **소 우 영**[2*]
[1](주)엔버 기업부설연구소
[2]한남대학교 컴퓨터공학과

Yu-Mi Bae[1] · **Sung-Jae Jung**[1] · **Wooyoung Soh**[2*]
[1]Enber Co., Ltd., Seoul 05029, Korea
[2]Department of Computer Engineering, Hannam University, Daejeon 34430, Korea

[요 약]

리눅스는 초창기부터 사용자 패스워드의 암호화를 위해 해시 알고리즘인 MD-5를 사용해 왔다. 최근 보안성이 강화된 패스워드 관리가 요구되면서 엔터프라이즈 리눅스 시스템에서는 MD-5보다 더욱 높은 신뢰성을 보이는 SHA-512 알고리즘을 사용하고 있다. 본 논문에서는 해시 및 암호화 알고리즘의 특징에 대해 비교 분석하고, 리눅스 사용자 정보의 관리 체제에 대해 알아본다. 이러한 분석을 기반으로 사용자 패스워드에 적용된 해시 알고리즘의 보안성에 대해 분석하고, 추가적으로 Apache, PHP, MySQL과 같은 공개 소프트웨어 파일 검증에 사용되는 해시 알고리즘 적용 사례를 분석한다. 마지막으로 관련 보안 도구인 John The Ripper를 분석하여 사용자 패스워드 관리를 통한 시스템 보안 강화 방법을 제시한다.

[Abstract]

MD-5 has been the hash algorithm to encrypt the user's password on Linux from the beginning. Recently the more reliable password management was demanded and SHA-512 algorithm became the hash algorithm on the recent Enterprise Linux which is more reliable than MD-5. This paper researching the characteristics of the hashing and encryption algorithms and find out about Linux User information management. Based on this analysis, and analysis of the security of the hashing algorithm is applied to the user password. In addition, analyzes the cases used hash algorithm applied to the validation of Open Source Software file, such as Apache, PHP, MySQL. Finally, by analyzing the security tool John The Ripper this paper suggests the enhanced security with the administrative management of passwords.

Key word : Hash algorithm, Encryption algorithm, MD algorithm, SHA algorithm, Linux security.

http://dx.doi.org/10.12673/jant.2016.20.1.72

Received 18 January 2016; **Revised** 5 February 2016
Accepted (Publication) 11 February 2016 (28 February 2016)

***Corresponding Author; Wooyoung Soh**

Tel: +82-42-629-7657
Email: wsoh@hnu.kr

Ⅰ. 서 론

운영체제의 주요 기능 중에 한 가지는 사용자 인증을 통해 허가된 사용자인지를 검증하여 적절한 권한을 부여하는 것인데, 권한 부여를 위해 전통적으로 아용되는 방법이 아이디 (ID; identity)와 패스워드(password)를 이용한 사용자 인증이다. 운영체제 입장에서는 시스템의 보안을 위해 보편적으로 공개되는 아이디보다는 패스워드 관리가 더욱 중요하다. 이러한 패스워드 관리의 핵심은 사용자가 텍스트(text)로 입력하는 패스워드의 내용을 숨기는 것이라고 할 수 있고, 이 때 사용되는 방법이 해시(hash) 및 암호화(encryption) 알고리즘이다[1].

본 논문에서는 현재 가장 많이 사용되고 있는 엔터프라이즈 리눅스인 CentOS 6.7 버전을 기반으로 적용된 해시 및 암호화 알고리즘에 대해 알아보고, 리눅스 사용자 정보 체제에 대해 분석한다. 아울러 사용자 정보 체제 이외에 사용된 사례 및 관련 도구에 대해 알아보고 결론을 맺는다.

Ⅱ. 해시 및 암호화 알고리즘

2-1 해시 (hash)

해시는 임의의 데이터부터 일종의 짧은 '전자 지문'을 만들어 내는 방법을 말한다. 보통 해시 함수를 이용하여 데이터를 자르고 치환하거나 위치를 바꾸는 등의 방법을 사용해서 결과를 만들고 이 결과값을 해시 값 (hash value)라고 한다. 해시 알고리즘에서 가장 중요한 것은 해시 값에서 원래의 데이터를 구하는 것이 불가능해야 한다는 것이다[2].

해시의 원리에 대해 살펴보면 여러 가지 방법이 사용이 있지만 가장 보편적인 방법은 나눗셈을 이용하는 것이다. 예를 들면 123456789라는 수와 한 자리 수만 다른 123486789라는 수가 있다고 가정했을 때 두 수를 그림 1과 같이 가운데를 기준으로 둘로 분할하여 큰 수를 작은 수로 나눈다. 그 후 앞 6자리의 숫자를 버리고 7자리를 해시 값으로 취하는 것이다. 두 해시 값만으로 해시 전의 원래 수를 알아내는 것은 불가능에 가깝고, 로직을 알고 있을지라도 버려진 앞 6자리 수를 알 수 없기 때문에 원래 값을 추출하는 것은 어렵다. 특히 값이 조그만 다르더라도 결과값이 무척 상이하게 생성된다.

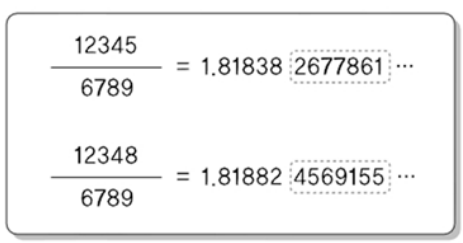

그림 1. 해시 알고리즘의 예
Fig. 1. Examples of Hash algorithm.

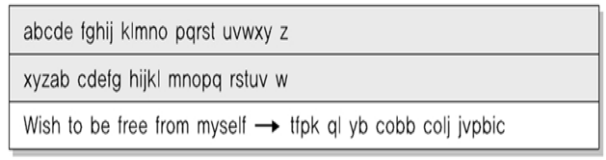

그림 2. 암호화 알고리즘의 예
Fig. 2. Examples of encryption algorithm.

2-2 암호화(encryption)

암호화는 특별한 지식을 소유한 사람을 제외하고 일반적인 사람들은 읽어도 데이터의 의미를 알 수 없도록 만드는 것이다. 암호화된 정보는 복호화(decryption) 작업을 통해 원래의 데이터를 읽을 수 있어야 한다.

암호화의 원리를 이해하기 위해 가장 쉬운 예는 로마시대에 사용된 치환(substitution)방식이 있다. 알파벳 순서를 3자씩 밀어내 대응되는 글자로 치환시키는 방법으로 알파벳을 3자씩 밀어서 대응시켰다는 내용을 알면 복호화가 가능하다. 여기서 '알파벳을 밀어서 대등되는 글자로 치환'하는 것이 암호화 알고리즘에 해당하고, 3이라는 숫자는 암호화 키(key)가 된다. 예를 들면 'Wish to be free from myself'라는 문구를 암호화하면 그림 2와 같다.

2-3 해시와 암호화 알고리즘의 비교

해시와 암호화 알고리즘 모두 원래의 데이터를 숨기는 기능을 제공한다. 그러나 해시 알고리즘은 해시 값에서 원래의 데이터를 구하는 것이 불가능해야 하고, 암호화 알고리즘은 복호화 작업을 통해 원래의 데이터를 읽을 수 있도록 해야 한다는 것이다. 해시와 암호화 알고리즘을 운영체제의 사용자 패스워드 관리에 적용하려면 추가로 주목해야할 특징이 있는데, 특정 값에서 나오는 해시 값이나 암호화된 값은 일정하다는 점이다.

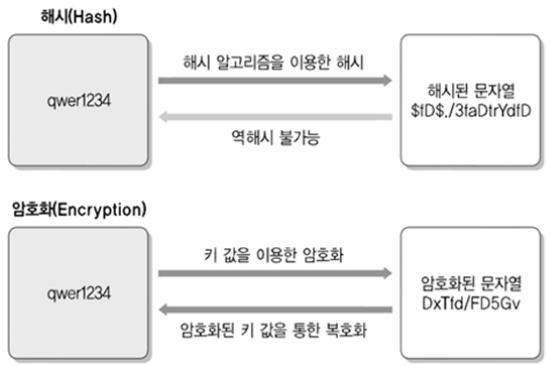

그림 3. 해시와 암호화 알고리즘의 차이점
Fig. 3. The difference between Hash and encryption algorithms.

Ⅲ. 리눅스 사용자 정보 체제와 알고리즘

3-1 개요

리눅스는 유닉스에 파생된 운영체제로 대부분의 정보를 텍스트 파일에 저장해서 관리한다. 사용자 계정 정보도 마찬가지로 /etc/passwd라는 텍스트 파일로 저장한다. /etc/passwd는 텍스트 파일이기 때문에 cat, head, tail 등과 같은 텍스트 관련 명령어로 확인이 가능하고, 권한이 있을 경우에는 vi나 emacs 등과 같은 편집기를 이용해 수정이 가능하다. /etc/passwd의 가장 큰 문제점은 모든 사용자가 파일의 내용을 볼 수 있다는 점이다. 이러한 문제점을 보완하기 위해 현재 대부분의 리눅스 배포판에서는 /etc/shadow 파일에 사용자의 패스워드를 별도로 관리하고, 관리자 계정인 root 이외의 접근을 불허하고 있다 [3]-[4].

3-2 관련 파일

1) /etc/passwd

/etc/passwd는 사용자 계정 정보를 저장하고 있는 파일로 콜론(:)을 구분자로 하여 'useranme:password:UID:GID:fullname: home-directory:shell' 과 같이 7개의 기본적인 정보를 기록한다 [5].

그림 4. /etc/passwd 파일의 예
Fig. 4. Examples of /etc/passwd.

표 1. /etc/passwd의 항목 설명
Table 1. Field description of /etc/passwd.

Field	Descriptions
posein	Account, the name of the user on system. ID called a one people
x	The encrypted user password. Currently managed separately from the /etc/shadow
500	The numerical user ID
500	The numerical primary group ID for this user
System Engineer	This filed is optional and only used fro informational purposes.
/home/posein	The user's Home Directory
/bin/bash	The program to run at login

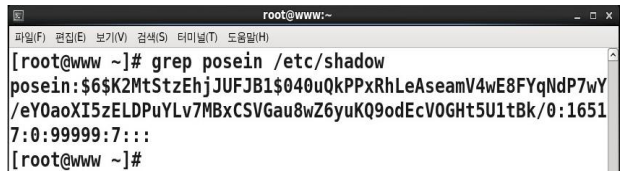

그림 5. /etc/shadow 파일의 예
Fig. 5. Examples of /etc/shadow.

표 2. /etc/shadow의 항목 설명
Table 2. Field description of /etc/shadow.

Field	Descriptions
posein	Login name.
6.....	Encrypted password. Initially, the hash algorithm is MD5 that was used, in recent years, the hash algorithm is used that SHA512.
15917	Date of last password change, expressed as the number of since Jan 1, 1970
0	Minimum password age, An empty filed and value 0 mean that there are no minimum password age.
99999	Maximum password age.
7	Password warning period
3	Password inactivity period
16070	Account expiration date
reserved field	This field is reserved for future use.

2) /etc/shadow

/etc/passwd는 사용자의 아이디 및 패스워드 등의 정보를 담고 있는 중요한 파일이나 모든 사용자가 이 파일의 내용을 볼 수 있도록 접근 권한이 설정되어 있다. 사용자의 패스워드의 경우에는 보통 사람들이 알아볼 수 없는 형태로 암호화하여 저장되기는 하나 잠재적으로 노출될 위험이 존재하였다. /etc/shadow는 /etc/passwd의 두 번째 필드인 패스워드 부분을 암호화여 관리하는 파일로 root 사용자 이외에는 접근이 불가능하도록 설정되어 있다. 이 파일은 'username:password:last: may:must: warn:expire:disable:reserved' 와 같이 9개의 필드로 구성되어 있고 그림 5 및 표 2와 같다.

3-3 알고리즘 적용 사례 분석

사용자의 패스워드가 저장되는 /etc/passwd는 모든 사용자가 접근이 가능한 파일이므로 해시 나 암호화 알고리즘을 이용한 패스워드 숨기기가 필수적으로 요구된다. 리눅스에서는 사용자의 패스워드를 숨기기 위해서 해시 알고리즘인 MD 알고리즘과 SHA 알고리즘을 이용한다.

1) MD (message digest) 알고리즘

MD 알고리즘은 암호화 알고리즘인 RSA (rivest shamir adleman)를 개발한 미국 MIT의 로널드 리베스트 교수가 함께

그림 6. MD5 알고리즘 적용 예
Fig. 6. Examples of MD5 algorithm.

개발한 해시 알고리즘이다. MD2, MD4, MD5, MD6와 같이 네 가지 종류가 존재한다. MD2는 1989년에 개발되었고, 8비트 컴퓨터에 최적화된 알고리즘이다. MD4는 1990년 개발되었고, 128비트(16byte)의 해시 값을 갖는다. MD5는 MD4의 확장판으로 1992년에 개발되었고, 신뢰성을 향상시켰다. MD6는 2009년에 개발되었고, 512비트의 해시 값을 갖는데 속도 문제 관련한 이슈가 존재한다. 리눅스는 초창기부터 MD5 알고리즘을 이용해서 사용자의 패스워드 정보를 숨겼으며 레드햇 계열 리눅스 기준으로 RHEL(red hat enterprise linux) 5 버전까지 사용되었다.

2) SHA (secure hash algorithm) 알고리즘

SHA 알고리즘은 1993년 미국 NIST(national institute of standard and technology)에서 SHA-0으로 개발되었지만 비공개된 중요한 결함으로 인해 SHA-1으로 대체되었다[6]. SHA-1은 미국 국가안보국인 NSA(national security agency)에 의해 개발된 160비트 해시 알고리즘이다. NSA에서는 추가로 SHA-2를 개발하였는데 대표적인 알고리즘에는 SHA-256과 SHA-512가 있다. SHA-256은 32 비트 워드(8*32=256비트)의 해시 함수를 사용하고, SHA-512는 64비트 워드(8*64=512비트)의 해시 함수를 사용한다[6]. 2015년 8월에 NIST에서 SHA-3을 개발하였는데, 1600비트의 해시 함수를 사용한다. 최근 대부분의 리눅스 배포판에서는 사용자의 패스워드를 감추는데 SHA-512 해시 알고리즘을 사용하고 있다. 서버 분야에서 많이 사용되는 레드햇 계열 리눅스에서는 RHEL 6 버전부터 가장 최근 버전인 RHEL 7까지 SHA-512를 기본적으로 적용하고 있다.

3) 사용자 패스워드에 적용된 해시 알고리즘 분석

사용자 패스워드를 더욱 안전하게 보관하기 위해서는 원래의 정보를 숨기는 기능 이외에 사용자들이 같은 패스워드를 설정하더라도 암호화된 값은 달라야 한다. 그러나 해시 알고리즘을 사용하든 암호화 알고리즘을 사용하든 기본적인 알고리즘 체제에서는 불가능하다. 현재 서버 분야에서 가장 많이 사용되

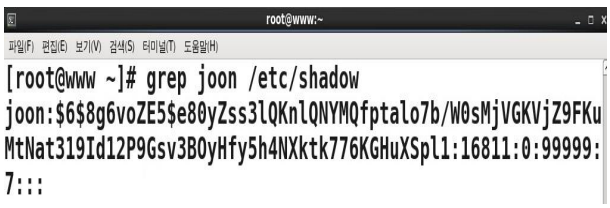

그림 7. SHA-512 알고리즘 적용 예
Fig. 7. Examples of SHA-512 algorithm.

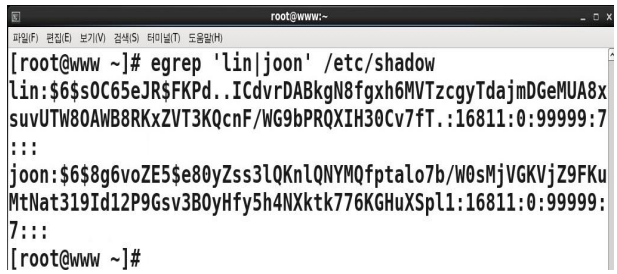

그림 8. 해시 값 비교
Fig. 8. Comparing a Hash value.

고 있는 RHEL 6 버전 계열 리눅스에서 2개의 계정인 lin과 joon을 생성한 뒤에 동일하게 1234라고 패스워드를 부여한 뒤에 /etc/shadow를 비교해보면 그림 8과 같다.

SHA-512 해시 알고리즘이 적용된 후 기록되는 /etc/shadow의 2번째 필드를 비교하면 값이 다른 것을 확인할 수 있다. 어떠한 알고리즘을 사용한다고 해도 동일한 정보값에서 결과로 나오는 암호화된 값은 동일하게 된다. 그렇다면 리눅스 운영체제에서는 이러한 문제를 어떻게 해결한 것인지 의문이 들 수도 있다. 보통 /etc/shadow의 2번째 필드에서 $6으로 시작하는 문자열 전부를 해시 값이라고 생각하는 경우가 많은데, 실제 구성은 $를 필드 구분자로 해서 3부분으로 나뉜다. 즉, '$해시알고리즘 표기값$솔트값$해시값'으로 구성된다. 해시 알고리즘 표기값은 어떠한 해시 알고리즘을 사용했는지를 나타내는 영역으로 리눅스에서는 보통 1 또는 6으로 표기된다. 1은 MD-5 알고리즘을 뜻하고, 6은 SHA-512 알고리즘을 나타낸다. 솔트값(Salt Value)은 음식에 첨부하는 소금처럼 동일한 정보에 첨가하여 서로 다른 해시 값을 갖도록 하는 역할을 한다. 해시값은 사용자가 입력한 패스워드와 두 번째 필드에 명기된 솔트값을 첨가하여 생성된다. 그림 8에서 보면 lin과 joon이라는 계정 모두 1234라는 패스워드를 설정했지만 서로 다른 솔트값인 kR090A1t와 QHGixfNw를 사용해서 서로 다른 해시 값을 만들어낸 것을 알 수 있다.

3-4 리눅스 환경에서 해시 알고리즘 사용 분석

1) MD 5 알고리즘

리눅스 초창기부터 사용되었던 MD-5 알고리즘을 이용한 해시 값을 알아보려면 기본적으로 설치된 openssl 명령을 사용하면 된다. 명령행에서 'openssl passwd -1'을 이용하면 되는데, 자동적으로 솔트값이 적용되어 생성된다. 만약 임의적으로 솔트값을 첨가하여 해시 값을 생성하려면 'openssl passwd −1 -salt 솔트값' 형식으로 실행하면 된다.

그림 9는 리눅스 명령행에서 1234라는 값에 대한 해시 값을 출력하는 과정이다. 임의의 솔트값을 첨가해서 서로 다른 해시 값이 나오는 것을 알 수 있다.

2) SHA-512 알고리즘 사용 분석

최근 대부분의 리눅스 배포판에서 사용자 패스워드 관리에

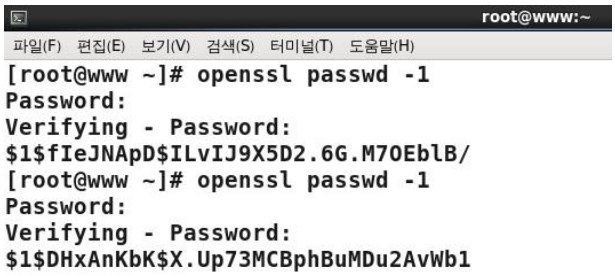

그림 9. openssl 명령 사용 예
Fig. 9. Examples of openssl command.

기본적으로 적용되는 SHA-512 알고리즘은 암호화 관련 함수인 crypt()을 이용해서 확인할 수 있다. 명령행에서는 perl 명령을 이용해서 확인할 수 있다. 사용법은 perl –e 'print crypt("패스워드값", "\$6\$솔트값");' 형태로 검증이 가능하다. 그림 10은 두 명의 사용자인 lin 및 joon 사용자의 패스워드를 1234로 설정하고 perl 명령과 crypt()함수를 통해 검증하는 예이다.

Ⅳ. 해시 알고리즘 적용 사례 및 관련 도구

4-1 공개 소프트웨어의 소스(Source) 파일 검증

대표적인 공개 소프트웨어인 아파치(apache), PHP, MySQL 등 수 많은 프로그램들이 배포되는 소스 파일 검증을 위해 MD-5 및 SHA 해시 알고리즘을 사용한다. 그림 11에서 그림 12를 살펴보면 소스 파일 옆에 관련 해시 값을 기재하거나 클릭을 통해 확인할 수 있도록 제공하고 있다[8]-[10].

리눅스에서는 관련 소스 파일 다운로드 후 명령행에서 md5sum, sha1sum, sha256sum, sha512sum 등과 같은 명령으로 검증할 수 있고, 그림 14는 PHP 소스 파일을 검증하는 과정이다. 이렇듯 공개 소프트웨어들은 해시 알고리즘을 이용해서 공인된 파일임을 증명하고 있다.

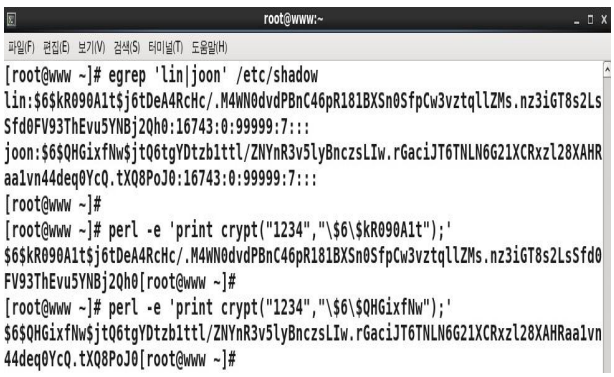

그림 10. crypt 함수를 이용한 검증 예
Fig. 10. Examples of verification using the crypt function.

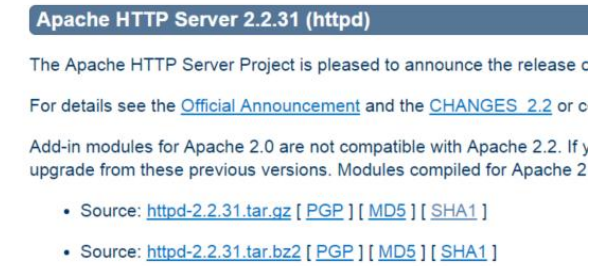

그림 11. 아파치 웹 서버 소스 파일 예
Fig. 11. Examples of Apache web server's source file.

그림 12. PHP 소스 파일 예
Fig. 12. Examples of PHP source file.

그림 13. MySQL 소스 파일 예
Fig. 13. Examples of MySQL source file.

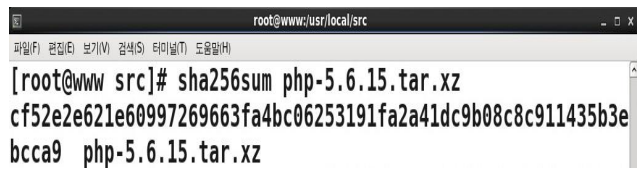

그림 14. PHP 소스 파일 검증
Fig. 14. Verification of PHP source file.

표 3. 주요 공개 소프트웨어의 검증 비교
Table 3. Verification comparison of leading open source software.

	Algorithm	Command
Apache	MD5, SHA-1	md5sum, sha1sum
PHP	MD5, SHA-256	md5sum, sha256sum
MySQL	MD5	md5sum

4-2 John The Ripper

John The Ripper는 Solar Designer가 개발한 유닉스 계열 패스워드 크랙 도구 (password crack tool)이다. 현재는 유닉스 계열 이외에도 Windows 계열, DOS, BeOS, OpenVMS 등 다양한

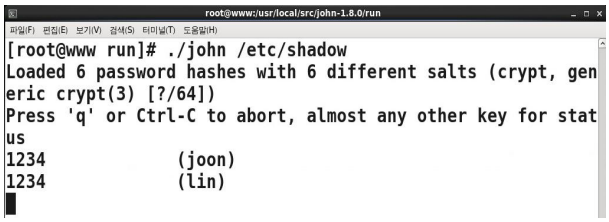

그림 15. John the Ripper 사용 예
Fig. 15. Examples of John the Ripper.

플랫폼도 지원한다[11]. 처리 속도를 높이기 위해 CPU의 특수 기능들을 이용한 최적화된 코드도 삽입하였다. 기본적인 원리는 사용자 패스워드의 해시 값 생성에 사용되는 crypt()함수를 이용하여 사전 파일(dictionary file)에 등록된 문자열을 조합한 뒤 /etc/shadow에 기록된 해시 값을 비교해서 사용자의 패스워드를 알아낸다. 그림 15는 John The Ripper를 이용해서 단순한 패스워드를 등록한 lin 및 joon 사용자의 관련 정보를 찾아내는 과정이다.

Ⅴ. 결 론

리눅스는 초창기부터 사용자 패스워드의 암호화를 위해 해시 알고리즘인 MD-5를 사용하였다. 현재 사용되는 대부분의 엔터프라이즈 리눅스는 MD-5보다 더욱 높은 신뢰성을 보이는 SHA-512 해시 알고리즘을 사용해서 사용자의 패스워드를 보호하고 있다. 공개 소프트웨어이고 무료라는 이미지가 강한 리눅스 운영체제이지만 공공기관이나 기업의 사용이 증가하면서, 엔터프라이즈 리눅스에서는 더욱 성능 좋은 해시 알고리즘 적용을 통해 보안을 강화하고 있다. 또한, 공개 소프트웨어 파일 검증에도 이용되고 있고, 관련 보안 도구인 John The Ripper도 해시 알고리즘을 이용하고 있다.

사용자 패스워드 유출로 인해 해당 시스템이 전체가 영향을 받는 사례가 무척이나 빈번하다. 이러한 문제점을 최소화하기 위해 더욱 신뢰성이 높은 해시 알고리즘이 사용되고 있고, 관리자측면에서는 주기적으로 John The Ripper와 같은 보안 도구를 이용해서 유추하기 쉽거나 사전 파일에 등록된 사용자 패스워드를 걸어낼 수도 있다. 그러나 가장 중요한 것은 각각의 사용자들이 유추하기 힘든 패스워드를 설정하도록 지속적으로 교육하고 감시하는 것이 가장 중요하리라고 판단된다.

감사의 글

이 논문은 2105년도 한남대학교 학술연구조성비 지원에 의하여 연구되었음.

참고 문헌

[1] D. I. Yang, *Introduction to Information Security and Training: System Hacking and Security*, Seoul, Korea: Hanbit Academy, 2013.

[2] D. I. Yang, *Introduction to Information Security,* Seoul, Korea: Hanbit Academy, 2013.

[3] S. J. Jung, and Y. M. Bae, *To conquer Linux Master First Class*, Seoul, Korea: Booksholic Publishing, 2015.

[4] S. J Jung, and K. Sung, "Management and security of user in linux server," *Journal of Advanced navigation Technology,* Vol. 19, No. 6, pp. 587-594, Dec. 2015.

[5] Y. M. Bae, and S. J. Jung, *To conquer Linux Master Second Class*, Seoul, Korea: Booksholic Publishing, 2014.

[6] National Institute of Standards and Technology [Internet]. Available: http://www.nist.gov/.

[7] National Security Agency [Internet]. Available: http://www.nsa.gov/.

[8] HTTP Server Project [Internet]. Available: http://httpd.apache.org/.

[9] The PHP Group [Internet]. Available: http://www.php.net/.

[10] Oracle, MySQL Database[Internet]. Available: http://www.mysql.com/.

[11] Openwall, John the Ripper password cracker [Internet]. Available: http://www.openwall.com/john/.

배 유 미 (Yu-Mi Bae)
2005년 2월 : 한남대학교 컴퓨터멀티미디어과 (공학사)
2007년 8월 : 한남대학교 정보기술과 (공학석사)
2013년 2월 : 한남대학교 컴퓨터공학과 (공학박사)
2013년 3월 ~ 현재: 한남대학교 컴퓨터공학과 겸임교수
2015년 11월 ~ 현재 : ㈜엔버 기업부설연구소 선임연구원
※ 관심분야 : 리눅스, 정보보안, 멀티미디어, 클라우드 컴퓨팅

정 성 재 (Sung-Jae Jung)
1998년 2월 : 한남대학교 컴퓨터공학과 (공학사)
2003년 8월 : 한남대학교 컴퓨터공학과 (공학석사)
2011년 2월 : 한남대학교 컴퓨터공학과 (공학박사)
2005년 3월 ~ 2010년 2월: 한남대학교 국제IT교육센터 전임강사
2015년 11월 ~ 현재 : ㈜엔버 기업부설연구소 소장
※ 관심분야 : 리눅스, 정보보안, 시스템보안, 클라우드 컴퓨팅, 서버 가상화

소 우 영 (Wooyoung Soh)
1979년 2월 : 중앙대학교 전자계산학과 (이학사)
1981년 2월 : 서울대학교 전자계산학과 (이학석사)
1991년 2월 : 메릴랜드대학교 전자계산학과 (이학박사)
1991년 9월 ~ 현재 : 한남대학교 컴퓨터공학과 교수
※ 관심분야 : 정보보호, 신경회로망