인덕대학교

운영체제 보안
- 우선순위 제어 및 암호화

컴퓨터소프트웨어학과

김병국 교수

학습목표

- 프로세스의 우선순위를 변경할 수 있다.
- 파일을 암호화 할 수 있다.
- 체크섬 툴을 활용할 수 있다.

목차

- 프로세스 우선순위
- 계정별 우선순위 제한
- 암호화
- 실습

1. 프로세스 우선순위 (1/4)

❑우선순위

- 프로세스는 고유의 우선순위를 가짐
- 값이 낮을 수록 높은 우선순위를 가짐

- 리눅스 운영체제 :
  - 새로운 프로세스를 생성시 부모의 우선순위를 적용
  - 값의 범위 : 0 ~ 39
  - 기본 값: 20

- 관련 명령: nice, renice

1. 프로세스 우선순위 (2/4)

❑우선순위 지정

- 명령어 : nice -n <오프셋> <명령어>
  - <명령어>가 실행될 때 20+< 오프셋>로 우선순위가 지정되어 실행됨
  - 오프셋이 -20보다 작으면 -20으로 처리됨
  - 오프셋이 19보다 크면, 19로 처리됨

❑우선 순위 변경

- 명령어 : renice <+/-오프셋> <프로세스ID>
  - 프로세스의 우선순위를 변경

2. 프로세스 우선순위 (3/4)

❑ 실습

①

```c
#include <stdio.h>
#include <string.h>
#include <time.h>
#include <unistd.h>

int ResultPrint(char *pBuffer);

```

②

```c
int main()
{
    int nPid;
    char buffer[BUFSIZ];

    memset(buffer, 0, BUFSIZ);

    nPid = fork();
    if (nPid > 0)
    {
        sprintf(buffer, "A (PID: %d)", getpid());
        ResultPrint(buffer);
    }
    else
    {
        sprintf(buffer, "\t\t\tB (PID: %d)", getpid());
        ResultPrint(buffer);
    }

    return 0;
}
```

[파일명: priority.c (1/2)]

2. 프로세스 우선순위 (4/4)

❑실습

```
30  int
31  ResultPrint(char *pBuffer)
32  {
33      int nCount = 0;
34
35      for (int i = 0;; i++)
36      {
37          for (int j = 0;; j++)
38          {
39              time(NULL);
40              if (j >= 0x0FFFFFFF)
41                  break;
42          }
43          nCount++;
44          printf("%s (%d)\n", pBuffer, nCount);
45      }
46
47      return 0;
48  }
```

3

[파일명: priority.c (2/2)]

3. 계정 별 제한

❑우선순위 지정

- 설정한 값이 기본값으로 적용되어 프로세스가 동작됨

```
#<domain>      <type>  <item>         <value>
#

test            soft    priority        19
test            hard    priority        19
```

[우선순위 지정 예]

❑우선순위 제한

- 변경될 수 있는 우선순위의 범위를 제한함

```
#<domain>      <type>  <item>         <value>
#

test            hard    nice            -20
test            soft    nice            -20
```

[우선순위 제한 예]

4. 암호화 (1/4)

❑ 관련 기본 용어

- 평문(Plain Text) : 본래의 메시지
- 암호문(Cipher Text) : 암호화된 메시지
- 암호화(Encryption) : 평문에 대하여 암호문으로 변환하는 작업
- 복호화(Decryption) : 암호문에서 평문으로 변환하는 작업
- 키(Key) : 암호화 또는 복호화를 위한 열쇠

- 해쉬(Hash) : 암호화의 일종(복호화 불가), 결과물(암호문)들의 비교를 통해 이들이 동일한 평문임을 입증하는 용도로 사용

4. 암호화 (2/4)

❑ OpenSSL

- 네트워크기반의 암호화된 데이터 통신을 위한 TLS(Transport Layer Security)와 SSL(Secure Sockets Layer) 프로토콜을 위한 오픈소스
- Eric A. Young과 Tim Hudson이 개발
- C 언어로 구현된 라이브러리
- 암호화 관련 다양한 유틸리티 함수들을 제공
- 다양한 운영체제 지원 : 유닉스 계열 운영체제(솔라리스, 리눅스, BSD), 윈도우, Mac OS
- HTTPS 사이트를 포함한 다양한 인터넷 서버에서 폭넓게 사용 중

OpenSSL
Cryptography and SSL/TLS Toolkit

- 제공되는 암호화 알고리즘
  - 관련 명령어: openssl enc -help
  - 지원 암호화 방식: 200여개 (확인 명령: openssl enc -list)

❑명령 옵션 (1/2)

- -<ciphername> : 사용할 암호화 알고리즘 이름
- -in <filename> : 입력 파일명
- -out <filename> : 출력 파일명
- -salt / -nosalt : salt 사용(기본값) / 미사용

salt 용도

- 암호와 과정에서 주어진 KEY 값에 일부 첨가제(salt) 넣고 처리.
- Rainbow 사전 해킹을 방지하려는 목적.

4. 암호화 (4/4)

❑명령 옵션 (2/2)

- -e : 암호화 수행(기본값)
- -d : 복호화 수행(-e 옵션과 같이 쓸 수 없음)
- -K <KEY> : 키 값 설정
- -iv <벡터값> : 초기 벡터값
- -p : 생성된 KEY와 초기 벡터값을 출력

5. 암호화 실습 (1/3)

❑RC4 암호화/복호화

- 암호화

```
kali@kali:~/OperatingSystem/12_3$ openssl enc -rc4 -K 1234567890 -in passwd.txt
 -out passwd.txt.enc -p
hex string is too short, padding with zero bytes to length
salt=7A802E49CD7F0000
key=12345678900000000000000000000000
kali@kali:~/OperatingSystem/12_3$
```

- 복호화

```
kali@kali:~/OperatingSystem/12_3$ openssl enc -d -rc4 -K 1234567890 -in passwd.t
xt.enc -out passwd.txt.dec -p
hex string is too short, padding with zero bytes to length
salt=7A70C44C167F0000
key=12345678900000000000000000000000
kali@kali:~/OperatingSystem/12_3$
```

5. 암호화 실습 (2/3)

❑AES 암호화/복호화

- 암호화

```
kali@kali:~/OperatingSystem/12_3$ openssl enc -aes-128-cfb -K 1234567890 -iv 0 -
in passwd.txt -out passwd.txt.enc
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
kali@kali:~/OperatingSystem/12_3$
```

- 복호화

```
kali@kali:~/OperatingSystem/12_3$ openssl enc -d -aes-128-cfb -K 1234567890 -iv
0 -in passwd.txt.enc -out passwd.txt.dec
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
```

5. 암호화 실습 (3/3)

❑ 체크섬

- MD5
- SHA256
- SHA512

```
kali@kali:~/OperatingSystem/12_3$ md5sum passwd.txt
4d4eb8b3af703a067b81ce7add1754cd  passwd.txt
kali@kali:~/OperatingSystem/12_3$ sha256sum passwd.txt
fd5cdca71803925e8596229439c89ba023c457e345e0daa2bc6c138f88c09102  passwd.txt
kali@kali:~/OperatingSystem/12_3$ sha512sum passwd.txt
eba5cedbff97f07b364d5b5815423b02a66beee190e7c6af132335697f2421795f33b07a27baf98b
b929a274f1d9ca618300649700ee3ea6c8efe203b4061e5c  passwd.txt
kali@kali:~/OperatingSystem/12_3$
```

수고하셨습니다.