

ネットワークセキュリティ演習

3回 利用者認証／認可とアクセス制御

暗号的ハッシュ関数の実行

nano エディタでメッセージを作成

```
1 | nano message
```

```
1 | hello Kindai people!  
2 |  
3 | [ 新しいファイル ]  
4 | ^G ヘルプ      ^O 書き込み  ^W 検索      ^K 切り取り  ^J 均等割付  ^C カーソル位置  
5 | ^X 終了        ^R 読み込み  ^\ 置換      ^U 貼り付け  ^T スペル確認 ^_ 行を指定
```

保存: コントロールO、 エンターキー

終了: コントロールX

SHA256によるsampleのハッシュ値の計算

```
1 | sha256sum sample
```

opensslコマンドを使う場合

```
1 | openssl sha256 sample
```

SHA512によるsampleのハッシュ値の計算

```
1 | sha512sum sample
```

opensslコマンドを使う場合

```
1 | openssl sha512 sample
```

パスワードファイルの確認

/etc/passwd の内容

```
1 | cat /etc/passwd
```

ユーザ名がkindaiのところを見つける

/etc/shadow の内容の確認

```
1 | sudo cat /etc/shadow
```

ユーザ名がkindaiのところを見つける

grep コマンドを使うとその行だけ見つけてくれる

パイプライン | を使うと、前のコマンドの実行結果を次のコマンドの入力にしてくれる

```
1 | sudo cat /etc/shadow |grep kindai
2
3 # 結果
4 kindai:$6$.3zGobdz$50uhb1Ntz1z.VIeuNuGCGzR9X0tjQ6d5AdurV/6PrCpGb2M..mod615dUtfLDBYx6e1
```

パスワードクラッカー

/etc/passwdのクラックツール「John The Ripper」を試してみる

John the Ripperをインストール

```
1 | sudo apt install john
```

クラック対象のユーザを作成する

わざと弱いパスワードをつけてみる

```
1 | sudo adduser userx
2 |
3 | # パスワードに hydrogen を入れてみる
```

/etc/shadow を解析可能にする

```
1 | sudo unshadow /etc/passwd /etc/shadow > ~/jpasswd
2 | sudo chmod 400 jpasswd
```

johnコマンドを実行

ubuntuの辞書ファイルを使ってパスワードを調べる

10分近くかかる

```
1 | john --users=userx --wordlist=/usr/share/dict/words jpasswd
```

発見したパスワードを表示する

```
1 | john --users=userx --show jpasswd
2 |
3 | userx:hydrogen:1003:1006:,,,:/home/userx:/bin/bash
4 |
5 | 1 password hash cracked, 0 left
```

Rainbow Table

レインボーテーブル無料配布サイトURL

```
1 | https://project-rainbowcrack.com/table.htm
```

キーロガー

linuxのキーロガーをインストールする

他人が自分のマシンを使った形跡を調べるのにも利用可能

logkeys のインストール

```
1 | sudo apt update
2 | sudo apt upgrade
3 | sudo apt install -y autoconf
4 | sudo apt install -y make
5 | sudo apt install -y gcc
6 | sudo apt install -y git
7 |
8 | git init
9 | git clone https://github.com/kernc/logkeys
10 |
11 | cd logkeys
12 | ./autogen.sh
13 | cd build
14 | ../configure
15 | make
16 | sudo make install
```

USBキーボードのイベント情報をしらべる

```
1 | cat /proc/bus/input/devices
```

出てくる結果の中で、Name= ** USB Keyboard ** というところを調べる

```
1 | I: Bus=0003 Vendor=1c4f Product=0027 Version=0110
2 | N: Name="SIGMACHIP USB Keyboard"
3 | P: Phys=usb-0000:00:14.0-4.2/input0
4 | S: Sysfs=/devices/pci0000:00/0000:00:14.0/usb1/1-4/1-4.2/1-4.2:1.0/0003:1C4F:0027.0002
5 | U: Uniq=
6 | H: Handlers=sysrq kbd event5 leds
7 | B: PROP=0
8 | B: EV=120013
9 | B: KEY=1000000000007 ff80000000007ff febeffdf3cffffff ffffffffef
10 | B: MSC=10
11 | B: LED=7
```

この中の H: のところ

```
1 | Handlers=sysrq kbd event5 leds
```

の場所をみると event5 になっている 各自のマシンで異なるeventの場合もある

キーロガーを起動する

最後の event5 は、上記のeventを指定している 各自で異なる場合がある

logtest.txt は、キーボードイベントの記録用ファイル

```
1 | sudo logkeys -s -o ~/logtest.txt -d /dev/input/event5
```

キーロガーを停止する

停止する前にキーボードを操作しておく

```
1 | sudo logkeys -k
```

キーのログを確認する

```
1 | sudo cat logtest.txt
```

シェルスクリプト

シェルスクリプトとはコマンドを自動実行するファイル

コマンドを実行してみる

```
1 | date
```

簡単なシェルスクリプトをnanoエディタで作成する

date コマンドを実行するシェルスクリプトを jikoku.sh という名前で作成する

```
1 | nano jikoku.sh
```

内容 1 行目は、かならず #!/bin/bash

```
1 | #!/bin/bash
2 | date
```

シェルスクリプトに実行権限を与える

```
1 | chmod 770 jikoku.sh
```

作成したシェルスクリプトを実行する

いまこのディレクトリ（カレントディレクトリ）を ./ と書く

```
1 | ./jikoku.sh
```

200人のユーザをつくる(1)

```
1 | nano add_students.sh
```

```
1 | #!/bin/bash
2 | i=1
3 | n=200
4 | while [ $i -le $n ];
5 | do
6 |     echo "gaksei$i"
7 |     i=`expr $i + 1`
8 | done
```

実行権限を与える

```
1 | chmod 770 add_students.sh
```

```
1 | ./add_students.sh
```

200人の学生ユーザをつくる(2)

```
1 | nano add_students.sh
```

★useradd -m を使う

```
1 | #!/bin/bash
2 | i=1
3 | n=200
4 | while [ $i -le $n ];
5 | do
6 |     echo "gaksei$i"
7 |     sudo useradd -m "gakusei$i"
8 |     i=`expr $i + 1`
9 | done
```

実行権限を与える

```
1 | chmod 770 add_students.sh
```

```
1 | sudo ./add_students.sh
```

200人の学生ユーザとディレクトリを削除する

```
1 | nano del_students.sh
```

```
1 | #!/bin/bash
2 | i=1
3 | n=200
4 | while [ $i -le $n ];
5 | do
6 |     echo "gaksei$i"
7 |     userdel -r "gakusei$i"
8 |     i=`expr $i + 1`
9 | done
```

実行権限を与える

```
1 | chmod 770 del_students.sh
```

```
1 | sudo ./del_students.sh
```

学生のプライマリグループをstudentにする

再度ユーザ作成

```
1 | ./add_students.sh
```

```
1 | nano mod_students.sh
```

```
1 | #!/bin/bash
2 | i=1
3 | n=200
4 | while [ $i -le $n ];
5 | do
6 |     echo "gakusei$i"
7 |     sudo usermod -g student "gakusei$i"
8 |     i=`expr $i + 1`
9 | done
```

```
1 | sudo chmod 770 mod_students.sh
2 | ./mod_students.sh
```

最初の100人の学生を情報学科グループに

```
1 | sudo groupadd joho
```

```
1 | nano joho_students.sh
```

```
1 | #!/bin/bash
2 | i=1
3 | n=100
4 | while [ $i -le $n ];
5 | do
6 |     echo "gakusei$i"
7 |     sudo usermod -G joho "gakusei$i"
8 |     i=`expr $i + 1`
9 | done
```

実行権限を与える

```
1 | chmod 770 joho_students.sh
```

```
1 | ./joho_students.sh
```


情報学科の20人を「情報エンジニア」の履修生に

```
1 | groupadd engineer
```

```
1 | nano eng_students.sh
```

```
1 | #!/bin/bash
2 | i=1
3 | n=20
4 | while [ $i -le $n ];
5 | do
6 |     sudo usermod -G joho, engineer "gakusei$i"
7 |     i=`expr $i + 1`
8 | done
```

実行権限を与える

```
1 | chmod 770 eng_students.sh
```

```
1 | ./eng_students.sh
```

情報学科の学生と教員のみがアクセスできるディレクトリを作成

演習用ディレクトリ作成

```
1 | mkdir /tmp/enshu2/
```

情報学科用ディレクトリ作成

```
1 | mkdir /tmp/enshu2/joho
```

情報学科学生用ディレクトリ作成

```
1 | mkdir /tmp/enshu2/joho/students
```