福井大学 リカレント 教育プログラム プログラミング応用

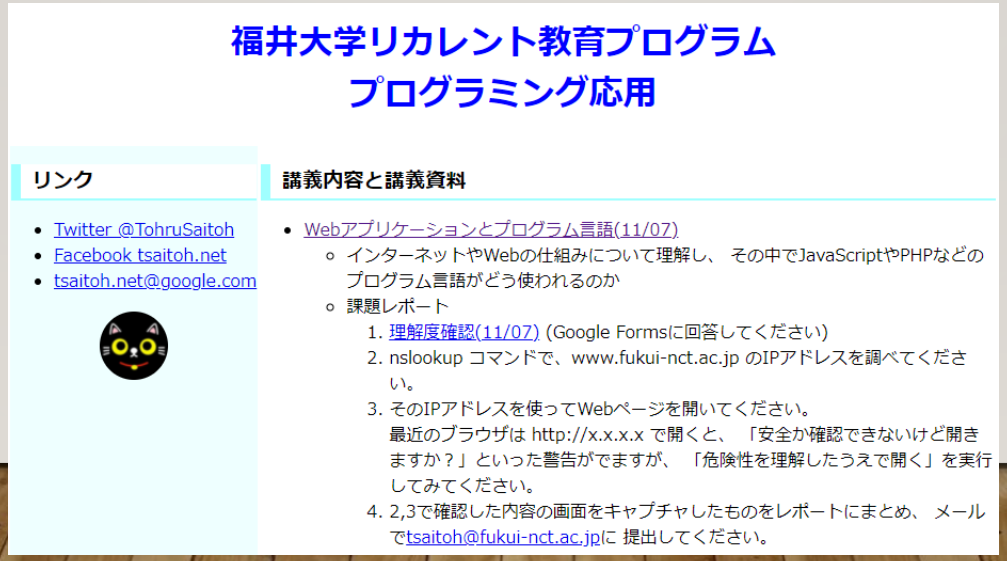
(4) WEBプログラミングと セキュリティ(11/28)

プログラミング応用
講義資料URL

https://tsaitoh.net/~t-saitoh/2021-11-recp/

login: guest

password: Guest



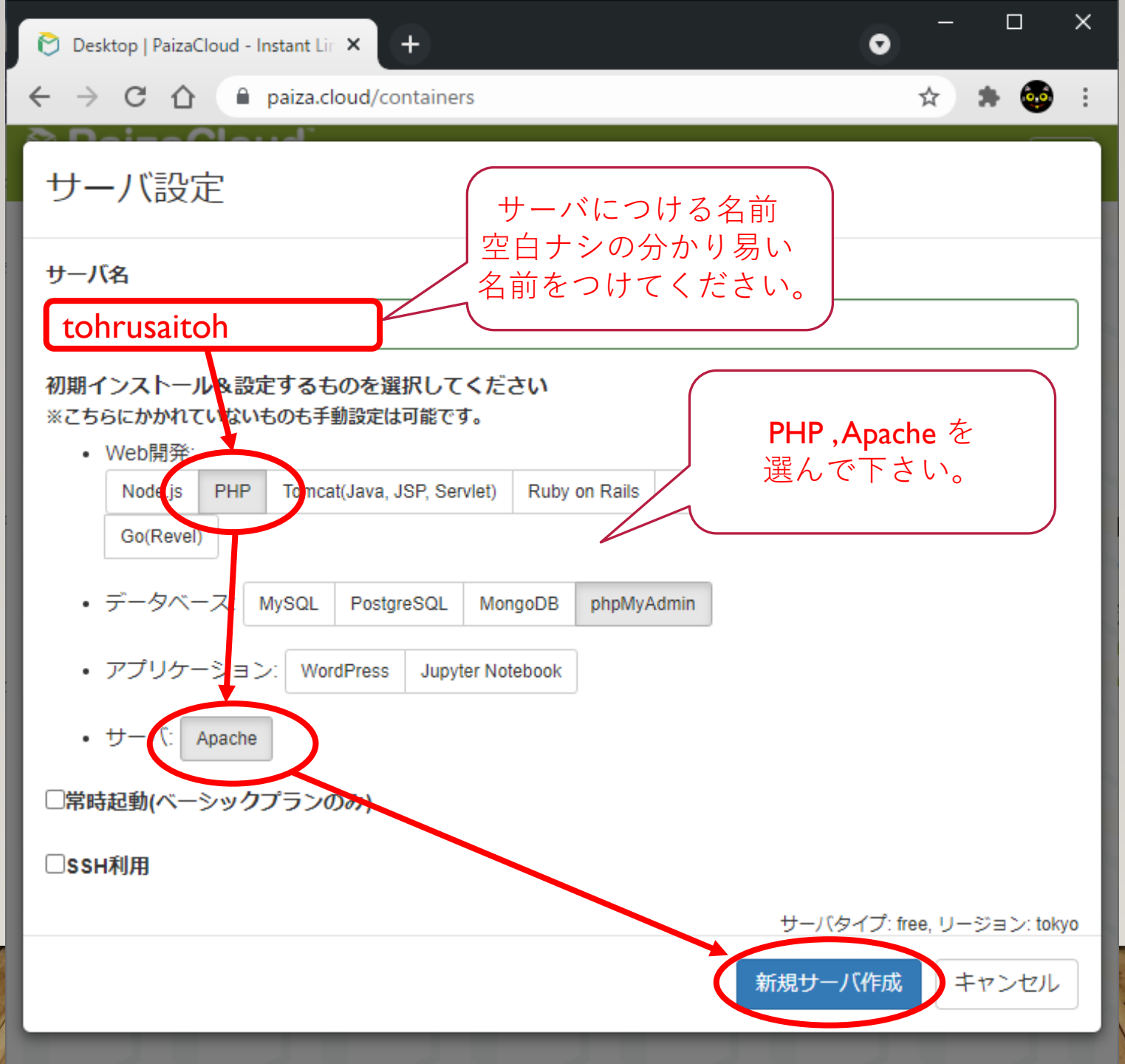
福井大学リカレント教育プログラム
プログラミング応用

リンク

- Twitter @TohruSaitoh
- Facebook tsaitoh.net
- tsaitoh.net@google.com

講義内容と講義資料

- Webアプリケーションとプログラム言語(11/07)
  - インターネットやWebの仕組みについて理解し、 その中でJavaScriptやPHPなどのプログラム言語がどう使われるのか
  - 課題レポート
    1. 理解度確認(11/07) (Google Formsに回答してください)
    2. nslookup コマンドで、www.fukui-nct.ac.jp のIPアドレスを調べてください。
    3. そのIPアドレスを使ってWebページを開いてください。
       最近のブラウザは http://x.x.x.x で開くと、「安全か確認できないけど開きますか？」といった警告がでますが、「危険性を理解したうえで開く」を実行してみてください。
    4. 2,3で確認した内容の画面をキャプチャしたものをレポートにまとめ、 メールでtsaitoh@fukui-nct.ac.jpに 提出してください。


Desktop | PaizaCloud - Instant Li
paiza.cloud/containers
サーバ設定
サーバにつける名前
空白ナシの分かり易い
名前をつけてください。
サーバ名
tohrusaitoh
初期インストール&設定するものを選択してください
※こちらにかかれていないものも手動設定は可能です。
PHP ,Apache を
選んで下さい。
Web開発:
Node.js
PHP
Tomcat(Java, JSP, Servlet)
Ruby on Rails
Go(Revel)
データベース:
MySQL
PostgreSQL
MongoDB
phpMyAdmin
アプリケーション:
WordPress
Jupyter Notebook
サーバ:
Apache
常時起動(ベーシックプランのみ)
SSH利用
サーバタイプ: free, リージョン: tokyo
新規サーバ作成
キャンセル

教材データのダウンロード

ターミナル画面では、サーバで動かす命令を入力
(先頭の$は入力しなくていい)

Webページ用フォルダに移動して
$ cd public_html

この講義のサーバ用資料をまとめてダウンロード
$ git clone https://github.com/tohrusaitoh/recp.git

WEBプログラミングとセキュリティ

- バックエンドサーバは、個人情報満載。
- いいかげんなプログラムは、個人漏洩となる。
- 実例と何が問題なのかを考えながら対応を考える。

WEBプログラミングとセキュリティ

- URLトラバーサル攻撃
- ユーザ認証のやり方
- ブルートフォース攻撃
- セッションハイジャック攻撃
- HTMLインジェクション攻撃
- SQLインジェクション攻撃
- インジェクション対策
- 情報セキュリティと倫理
- サニタイジングとは
- ネットワークとFireWall

情報セキュリティコンテストの CTF(Capture The Flag)を試す

Simple Capture The Flag

https://tsaitoh.net/~t-saitoh/ctf/

CTFとは

CTF(Capture The Flag)とは、情報技術・情報セキュリティに興味を持ってもらうために、 情報技術・セキュリティ技術を知っていれば解ける問題を、解けた問題数と難易度に応じて ポイントをつけて競う大会です。 基本は、与えられたヒントを元に、データの中に埋め込まれた FLAG{XXXXX} という形式 のデータを探し回答サーバに送ります。

練習問題

答え合わせ： 問題 問題を選んで 答え

Email:

確

1. 暗号・コーディング系

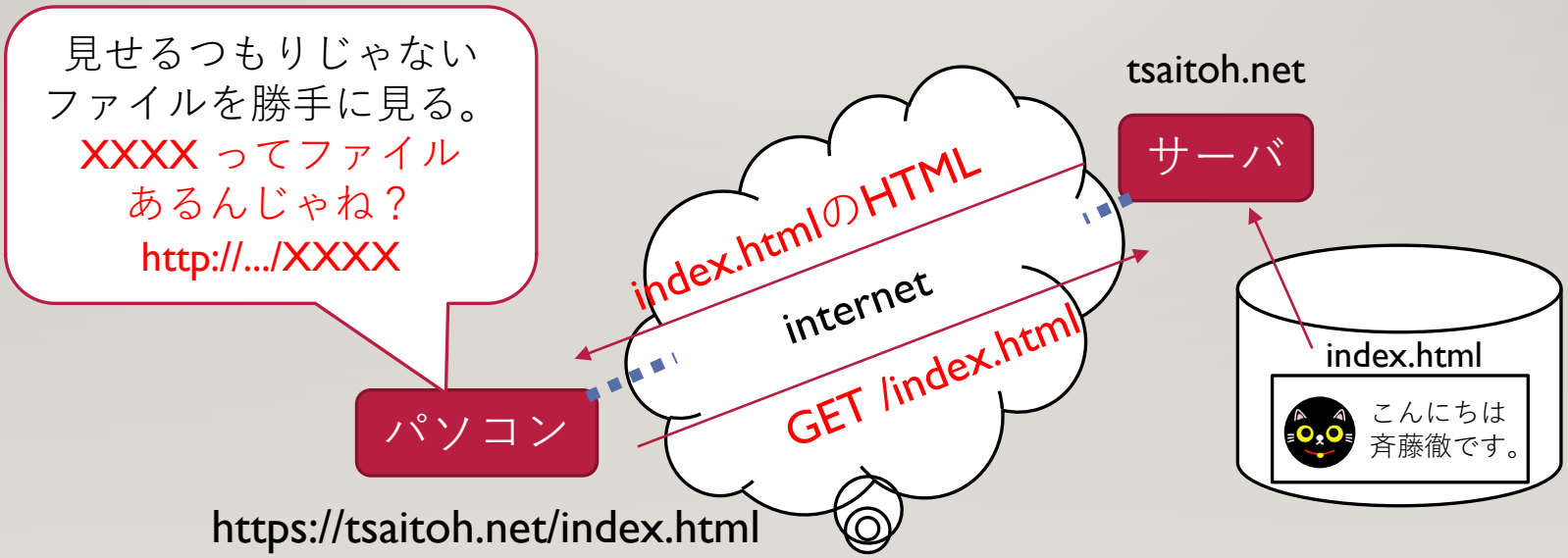
- 1.1 簡単な暗号
- 1.2 なんて書いてある(闇夜の烏)
- 1.3 なんて書いてある(フォークダンス)

Chrome なら「右ボタン」「ページソース」を表示

```
<html>
  <head>
    <title></title>
  </head>
  <body bgcolor="#000000">
    <h1><font color="#FFFFFF">闇夜の烏</font></h1>
    <font color="#000000">FLAG{crow-in-the-night}</font>
  </body>
</html>
```

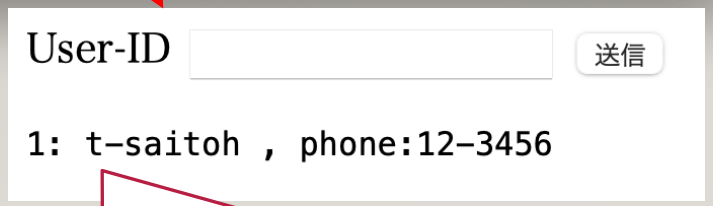
URLトラバーサル攻撃(I)
サーバの仕組みを悪用

- WebはURLで情報の場所を指定
  - ルートフォルダ　/var/www/html
  - ユーザの公開フォルダ　~/public_html/



CTF 3.1 バックアップファイルに注意しよう

URLトラバーサル攻撃(2)
バックアップファイルを読む



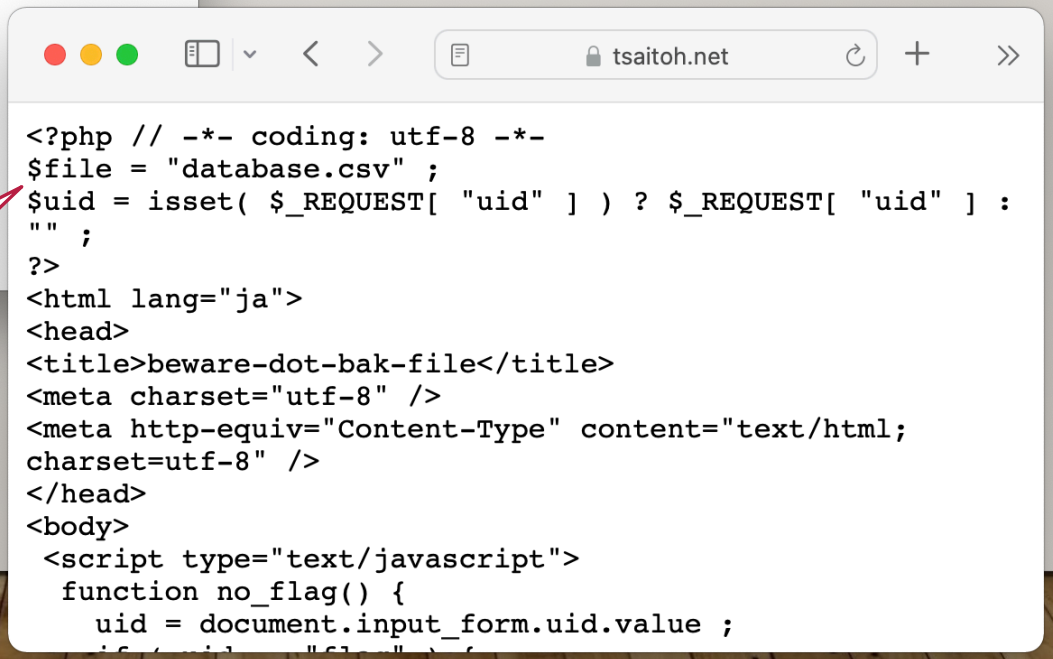
指定されたユーザの情報を探して
表示するWebページ

- プログラムのエディタの、バックアップ機能
- hoge.php の1つ前を hoge.bak で保存

https://tsaitoh.net/
~t-saitoh/ctf/beware-dot-file.php
↓
~t-saitoh/ctf/beware-dot-file.bak があるんじゃね？

URLトラバーサル攻撃(3)
PHPプログラムの漏えい

https://tsaitoh/net/
~t-saitoh/ctf/beware-dot-file.bak



PHPのプログラム
が見えちゃった。

URLトラバーサル攻撃(4)
データファイルのアクセス制限不備

tsaitoh.net

```
<?php // -*- coding: utf-8 -*-
$file = "database.csv" ;
$uid = isset( $_REQUEST[ "uid" ] ) ? $_REQUEST[ "uid"
"" ;
?>
<html lang="ja">
<head>
<title>beware-dot-bak-file</title>
<meta charset="utf-8" />
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
 <?php
      if ( ($fp = fopen( $file , “r” )) !== false ) {
           :
```

あれ、**同じフォルダに**ある
database.csv を読んでるな...

https://tsaitoh.net/~t-saitoh/ctf/database.csv
ってURL指定したら、
読めるんじゃね？

```
# -*- coding: utf-8; -*-
# データファイル
t-saitoh phone:12-3456
sakamoto email:sakamoto@example.jp
aoyama phone:090-9999-9999
flag FLAG{MatsushimaNanako}
```

読めちゃった
全ユーザのデータじゃん...

URLトラバーサル攻撃(5)
対策は？

- **URLトラバーサル(ディレクトリトラバーサル)攻撃**
- どうすればよかったのか？

1. ○.bak ファイルは消す！
   最近のWebサーバは、○.bak , ○~ はアクセス禁止
2. データファイルはアクセス禁止 or 公開用フォルダ
   /var/www/html , public_html/ フォルダ内に保存しない

前回11/21のsampleH.php も、プログラムと同じフォルダに shopping.db がある。**やばくね？**

ユーザ認証のやり方
BASIC認証

- アクセス制限したいフォルダに .htaccess を置く

AuthType Basic	単純な認証方法
AuthName “Input ID and Password“	表示するメッセージ
AuthUserFile …/public_html/…/.htpasswd	パスワードファイル
require valid-user	ユーザ名の一致が必要

- パスワードファイル .hpasswd

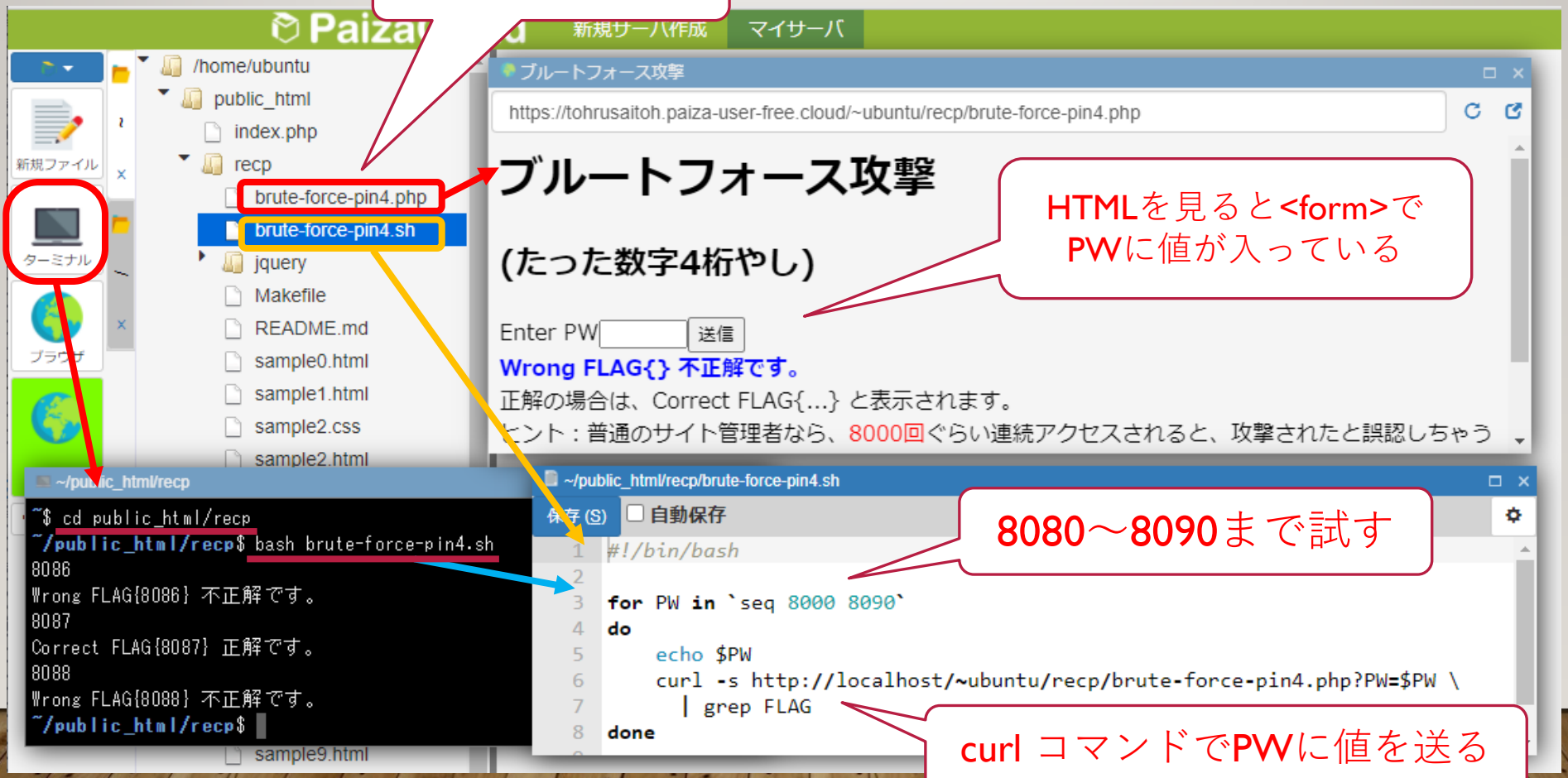
guest:$apr1$/1234c3U$AWn2wkw8hcI0ITS3txabce

ユーザ　　暗号化されたパスワード

Paiza.Cloud は、BASIC認証
を使うには設定が必要みたい

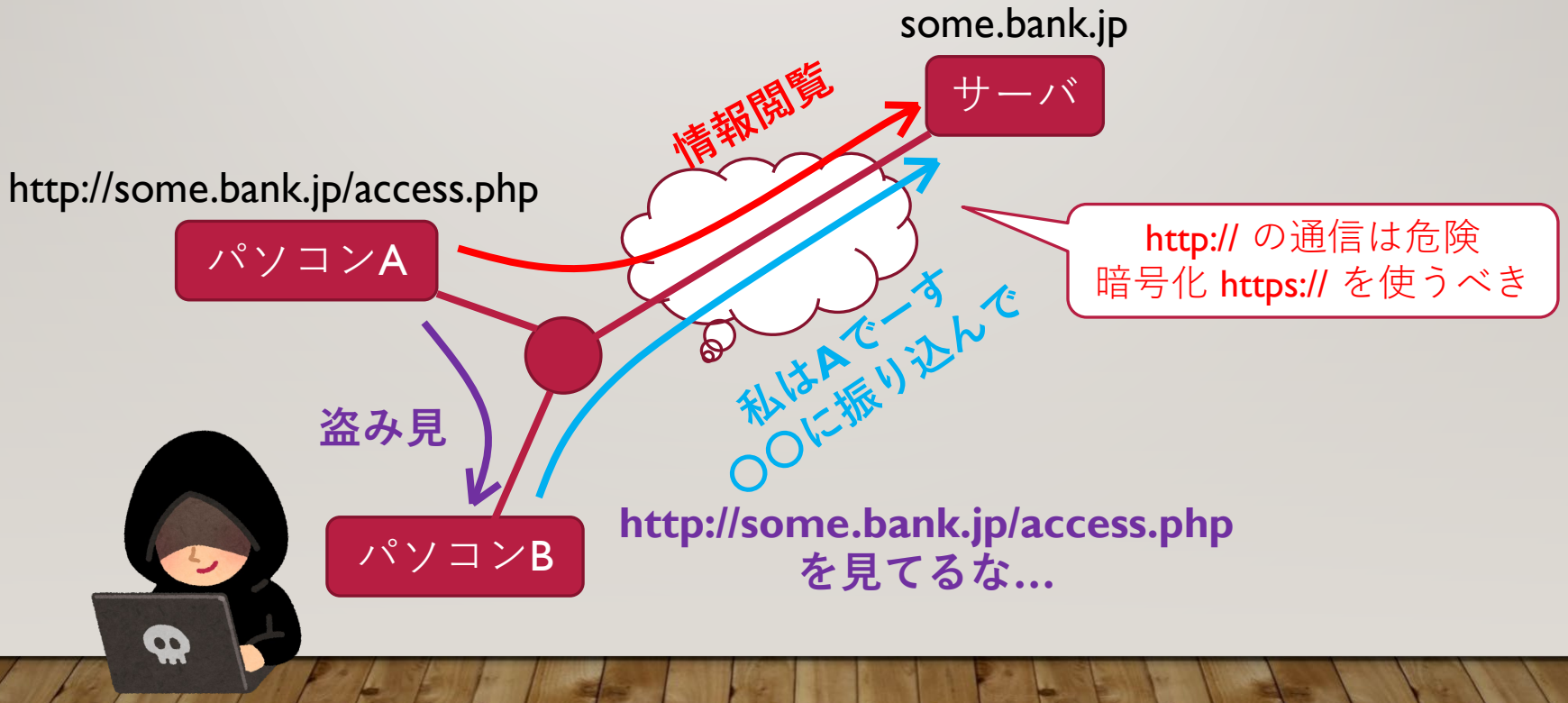
CTF 4.3 数字4桁のパスワードを答えてください

ブルートフォース攻撃
総当たり攻撃



セッションハイジャック攻撃(I)
http:// は危険

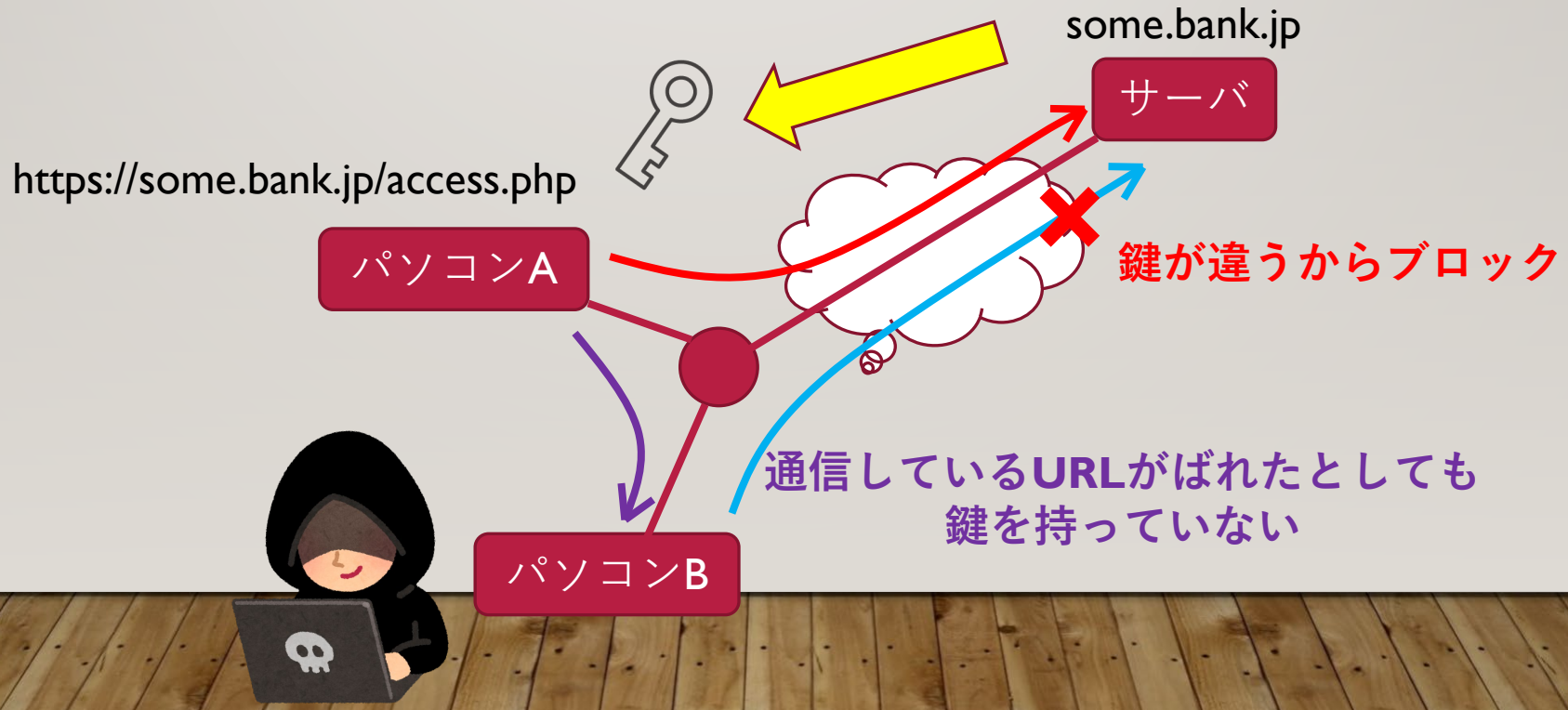
- http:// 通信は、通信内容が盗み見られるかも



セッションハイジャック攻撃(2)
セッションキーの発行

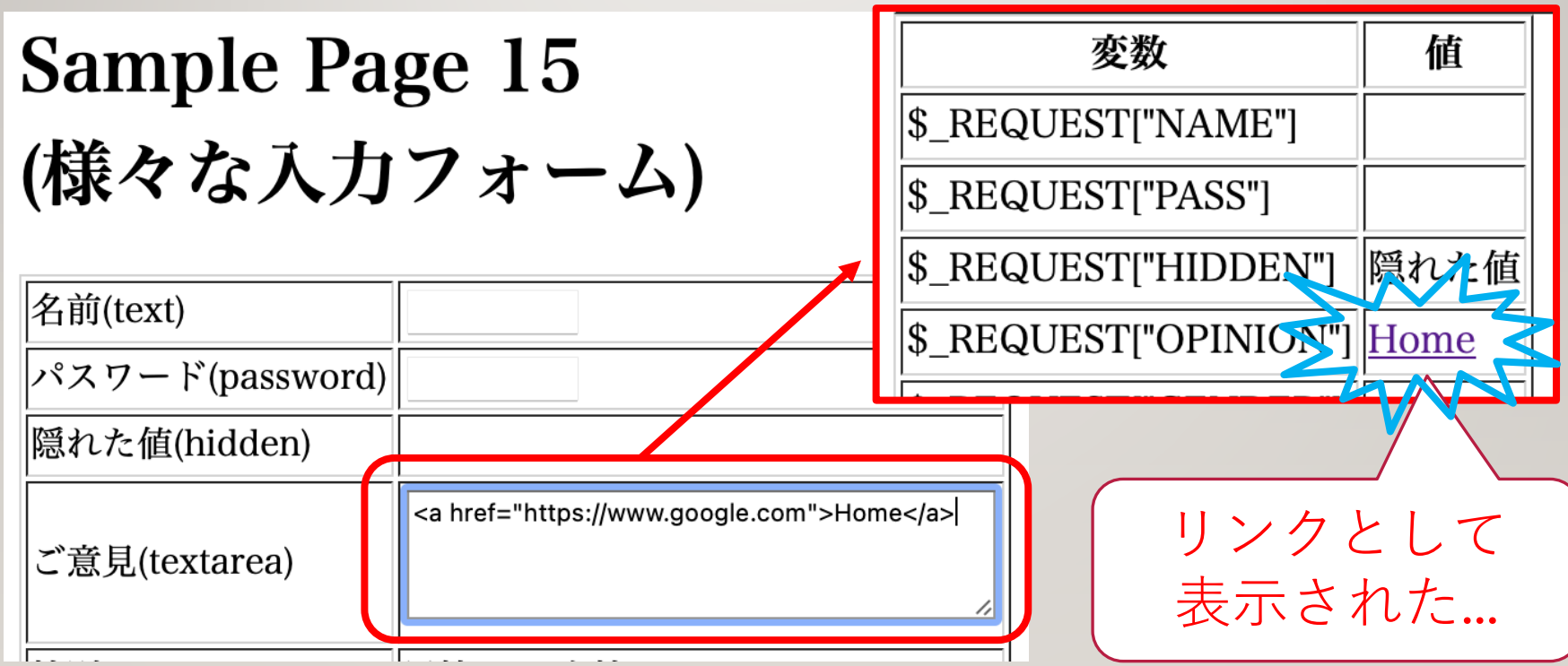
- **Cookie** は、ブラウザにデータを覚えてもらう仕組み
- サーバは、**通信相手に鍵を発行**

session_start(...) を使う



演習資料 sampleF.php

HTMLインジェクション(I)

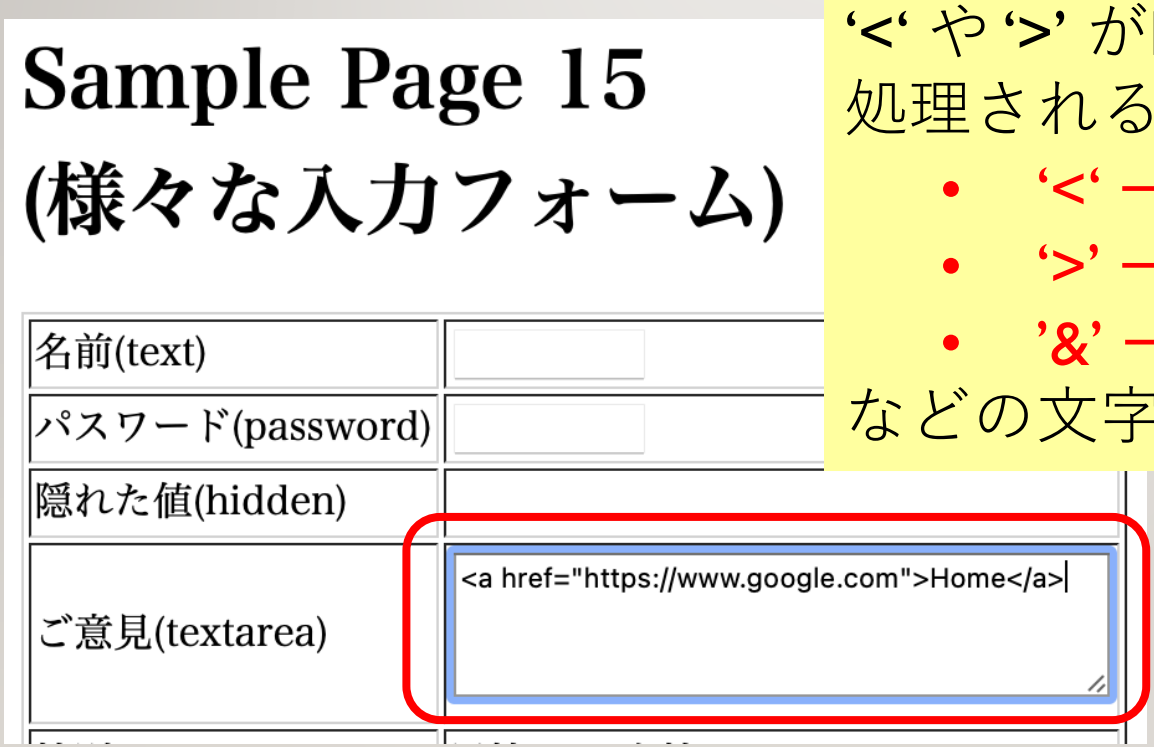


銀行のホームページにこういった表示が出たら、銀行のTOPに戻るボタンと誤解されて、偽物サイトに誘導されるかもしれない。

◆フィッシング(Phishing)攻撃

◆クロスサイトスクリプティング(XSS)攻撃

HTMLインジェクション(2)
対策は？



‘<‘ や ‘>’ がHTMLとして
処理されると危険

- ‘<‘ → <
- ‘>’ → >
- ’&’ → &

などの文字列の書き換えが必要

```
<td><?php echo $_REQUEST[“OPINION”] ; ?></td>
```

専用の関数を使う

```
<td><?php echo htmlspecialchars($_REQUEST[“OPINION”]) ;?></td>
```

SQLインジェクション攻撃(I)

SQL-Injection

Welcome やほぉびーびー

Enter ID [ ] 送信

```
select * from ID_Profile where id='' ;
input user-id like t-saitoh
```

Enter ID t-saitoh 送信

```
select * from ID_Profile where id='t-saitoh' ;
1 t-saitoh phone:12-3456
```

正しくSQLを実行してくれた

Enter ID ’ or 1=1 – ’ 送信

```
select * from ID_Profile where id='' or 1=1 -- '' ;
1 t-saitoh phone:12-3456
2 sakamoto email:sakamoto@example.jp
3 aoyama phone:090-9999-9999
4 flag FLAG{YBB=500Yen}
```

全データが
表示された

SQLインジェクション攻撃(2)
個人情報漏洩の怖さ

- Yahoo! BB 顧客情報漏えい事件
  - SQLインジェクション攻撃
- 2004年2月27日 450万人の個人情報漏洩
- 孫正義 氏は、漏洩対象者に**500円の金券**
- 500円×450万人(23億円)+株価下落=100億円被害

sql = ”select * from DB where id = **’$a’** ” ;
$a の中の「’ ,”」を安全な文字に置き換える。
(例) `$a = $dbh->real_escape_string( $a ) ;`

インジェクション対策
JavaScript では不十分

- HTML/SQL インジェクションは、
  ユーザの入力の中の「<,>,‘ 」が原因
- フロントエンドの HTML + JavaScript プログラムで
  「<,>,’」を入力できなくすればいい...

甘いね!

HTML,Javaをダウンロードして
対策処理を消せばいいじゃん...

curl コマンドでサーバに
直接データ送ればいいじゃん...

情報セキュリティと倫理(I)
とある親切な先生の話

- A$_{ccs}$社：「Webページを作ったぞ!」
- O$_{ffice}$先生：「あれ？SQLインジェクションできるよ！やばくね？」

- O先生→A社(Web管理人) ✉
  - O先生「SQLインジェクション攻撃できるから直した方がいいよ！」
  - A社　「SQLインジェクションって何？、ま、いっか、しーらね！」
- O先生「やばいんだけどなぁ...**講習会で他の人に知ってもらおう**」
  - 「SQLインジェクションはこうやって発生します！
    ためしに、A社のサイトを例にやってみよう！」

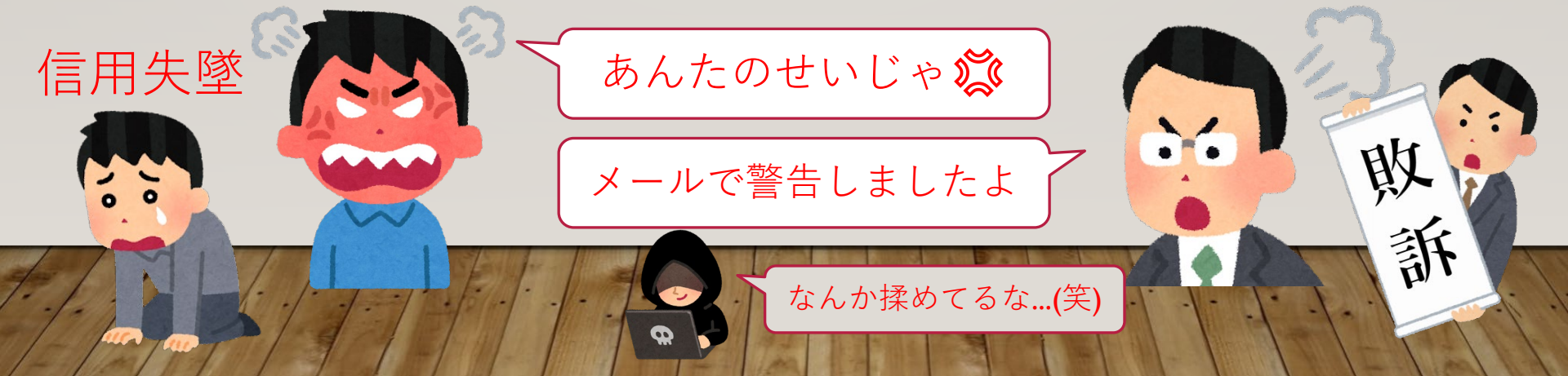
講習資料は、CD-ROM で配布しますね

情報セキュリティと倫理(2)
とある親切な先生の話

- O先生「SQLインジェクションはこうやって発生します！
  ためしに、A社のサイトでやってみよう！」

- 講習会参加Bさん「SQLやべーな、2ch で他の人にも教えよう!」
- 2ch 読者「やってみた凸!すげー個人情報ダウンロードできたぞ!」
  「Winny(フィル共有ソフト)に流すか...」

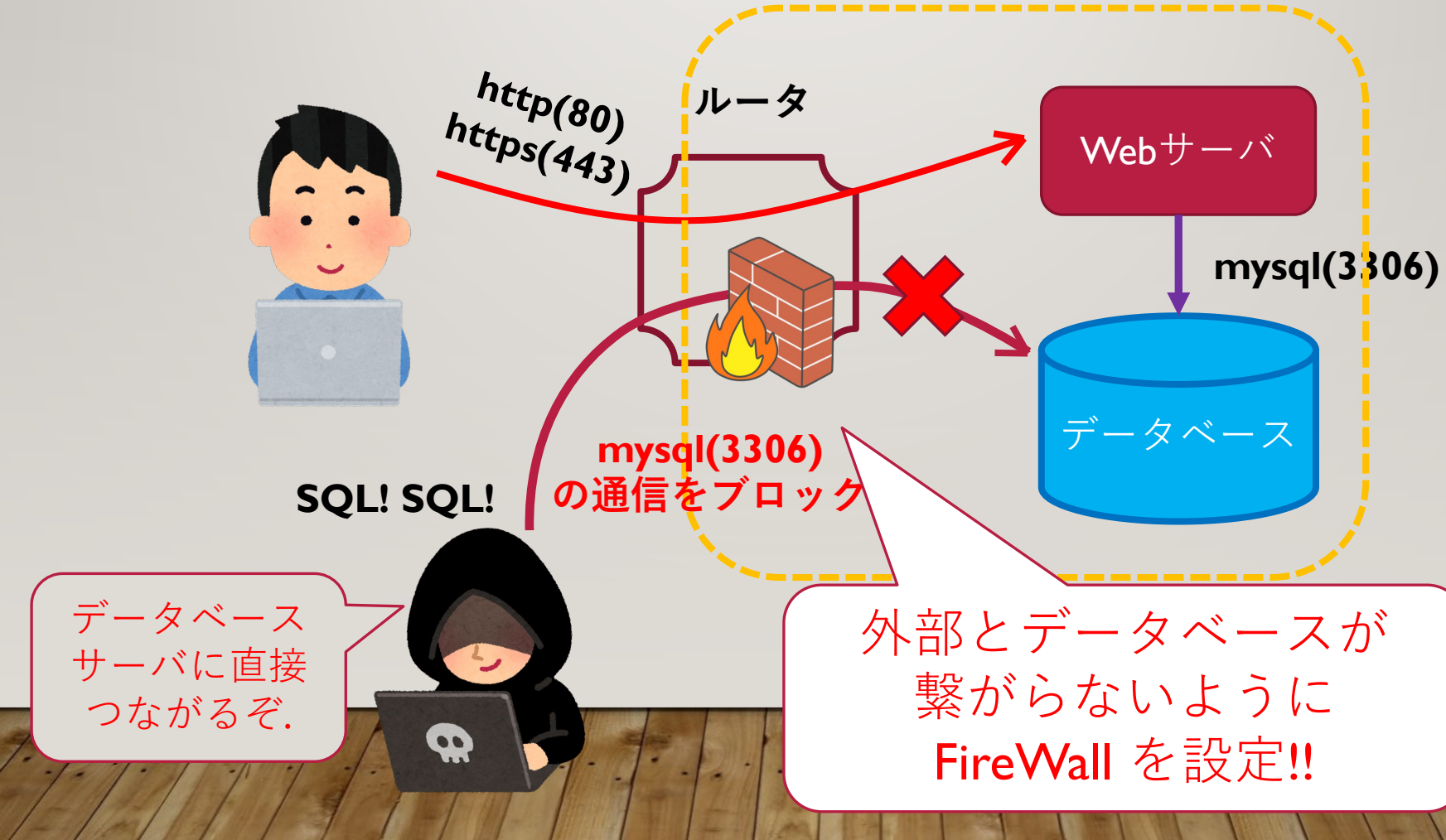
派手に情報漏洩



サニタイジングとは

- 利用者が入力した文字データを受け取る際に、プログラムにとって特別な意味を持つ可能性のある文字や文字列を検知して、一定の規則に従って別の表記に置き換えること。「無害化」とも呼ばれる。(IT用語辞典 e Wordsより引用)
  - ディレクトリトラバーサル　「/ , .」
  - HTMLインジェクション　「<,>,&」
  - SQLインジェクション　「’ , ” 」
- フロントエンド(JavaScript)での対策は不十分

ネットワークとFireWall



まとめ

今日のまとめ

- Webプログラミングで、セキュリティ対策としての攻撃方法と防御方法を解説
- インターネットの仕組みを理解して対策を考える
- 個人情報漏洩は、きわめて重大なリスク。

全４回のまとめ

- Webの仕組みとHTML
- フロントエンドのプログラミング
- バックエンドのプログラミング
- 情報セキュリティ