



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、「安全が確認できないけど開きますか?」といった警告がですが、「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。

サーバ設定

サーバ名

tohrusaitoh

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。

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。

• 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

PHP, Apache を
選んで下さい。

☐ 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



```
~/public_html
~$ cd public_html
~/public_html$ git clone https://github.com/tohrusaitoh/recp.git
Cloning into 'recp'...
remote: Enumerating objects: 52, done.
remote: Counting objects: 100% (52/52), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 52 (delta 15), reused 46 (delta 9), pack-reused 0
Unpacking objects: 100% (52/52), done.
~/public_html$
```



メッセージを残します

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 答え

Email:

確認

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(闇夜の鳥)

■ 1.3 なんて書いてある(フォークダンス)

闇夜の鳥

Chrome なら「右ボタン」
「ページソース」を表示

```
1 <html>
2   <head>
3     <title></title>
4   </head>
5   <body bgcolor="#000000">
6     <h1><font color="#FFFFFF">闇夜の鳥</font></h1>
7     <font color="#000000">FLAG{crow-in-the-night}</font>
8   </body>
9 </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/`

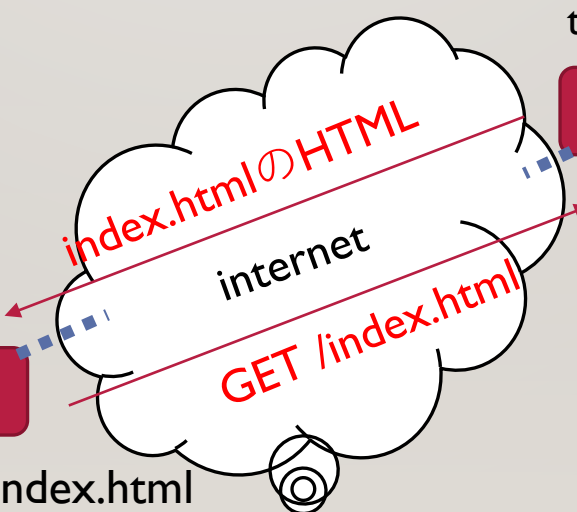
見せるつもりじゃない
ファイルを勝手に見る。

XXXX ってファイル
あるんじゃない？

`http://.../XXXX`

パソコン

`https://tsaitoh.net/index.html`



tsaitoh.net

サーバ

index.html



こんにちは
斉藤徹です。

URLトラバーサル攻撃(2)

バックアップファイルを読む



Beware-Dot-Bak-File

あびばあ会員データベース

User-ID

input user-id like t-saitoh

User-ID

1: t-saitoh , phone:12-3456

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
があるんじゃないか？



URLトラバーサル攻撃(3)

PHPプログラムの漏えい

https://...-dot-file.bak

Beware-Dot-Bak-File

あびばあ会員データベース

User-ID

input user-id like t-saitoh

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

**PHPのプログラム
が見えちゃった。**

```
<?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" />  
</head>  
<body>  
<script type="text/javascript">  
function no_flag() {  
    uid = document.input_form.uid.value ;  
    if ( uid != "t-saitoh" ) {
```

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



```
<?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](https://tsaitoh.net/~t-saitoh/ctf/database.csv)
ってURL指定したら、
読めるんじゃないか？

```
# -*- 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 `.htpasswd`

`guest:$apr1$/1234c3U$AWn2wkw8hcl0ITS3txabce`

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桁のパスワードを教えてください

ブルートフォース攻撃 総当たり攻撃

ブラウザで開く

ブルートフォース攻撃 (たった数字4桁やし)

HTMLを見ると<form>で
PWに値が入っている

8080~8090まで試す

curl コマンドでPWに値を送る

The screenshot shows a web browser window displaying a brute force attack page titled "ブルートフォース攻撃 (たった数字4桁やし)". The page has a form with "Enter PW" and a "送信" button. Below the form, it says "Wrong FLAG{ } 不正解です。" and "正解の場合は、Correct FLAG{...} と表示されます。" and "ヒント：普通のサイト管理者なら、8000回ぐらい連続アクセスされると、攻撃されたと誤認しちゃう".

On the left, a file explorer shows the directory structure: /home/ubuntu/public_html/recp. The file "brute-force-pin4.php" is highlighted, and a red arrow points to it from the "ブラウザで開く" callout. The file "brute-force-pin4.sh" is also highlighted, and a yellow arrow points to it from the "curl コマンドでPWに値を送る" callout.

At the bottom, a terminal window shows the execution of the script: `~/public_html/recp$ cd public_html/recp`, `~/public_html/recp$ bash brute-force-pin4.sh`. The output shows: `8086 Wrong FLAG{8086} 不正解です。`, `8087 Correct FLAG{8087} 正解です。`, and `8088 Wrong FLAG{8088} 不正解です。`. A blue arrow points from the "curl コマンドでPWに値を送る" callout to the `curl` command in the script.

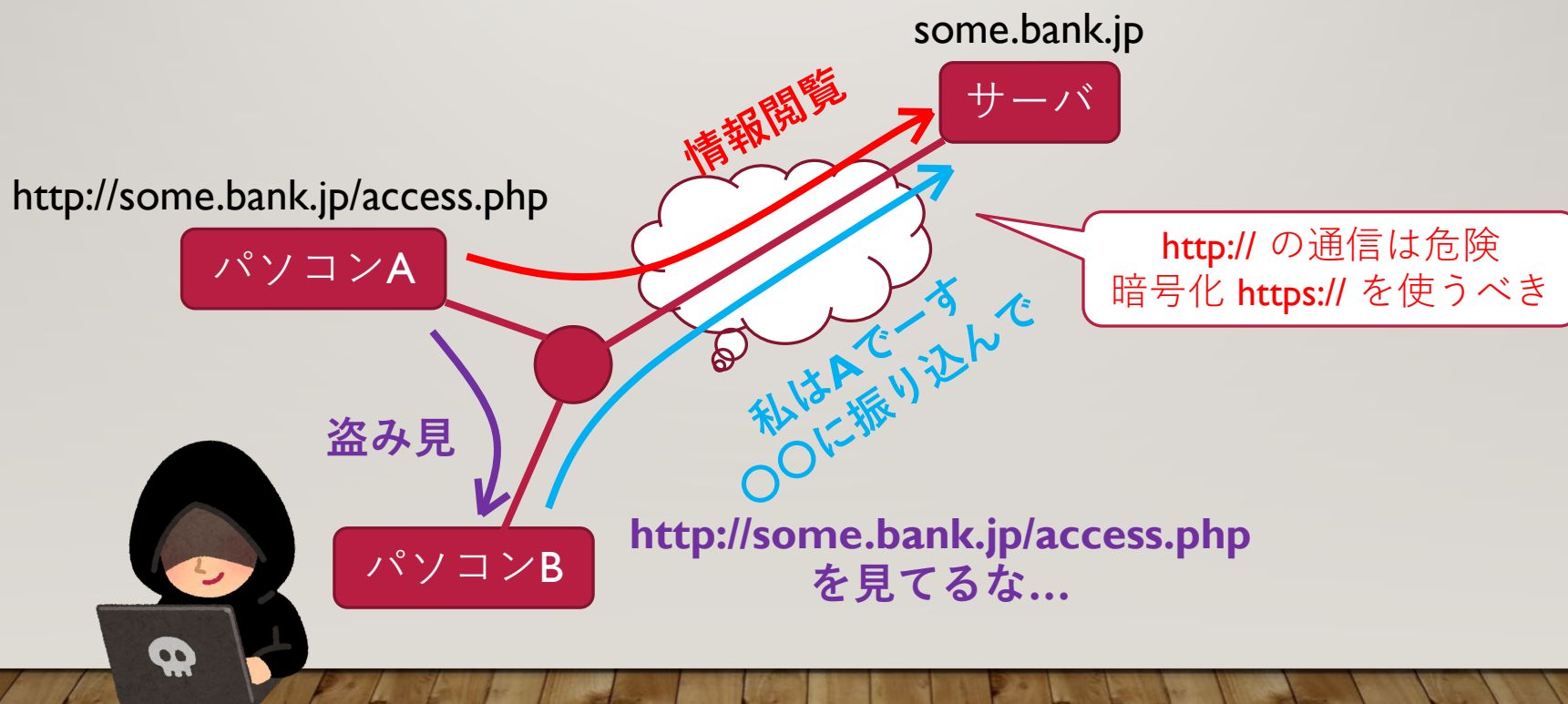
Below the terminal, a code editor shows the content of `brute-force-pin4.sh`:

```
#!/bin/bash
for PW in `seq 8000 8090`
do
    echo $PW
    curl -s http://localhost/~ubuntu/recp/brute-force-pin4.php?PW=$PW \
        | grep FLAG
done
```


セッションハイジャック攻撃(I)

http:// は危険

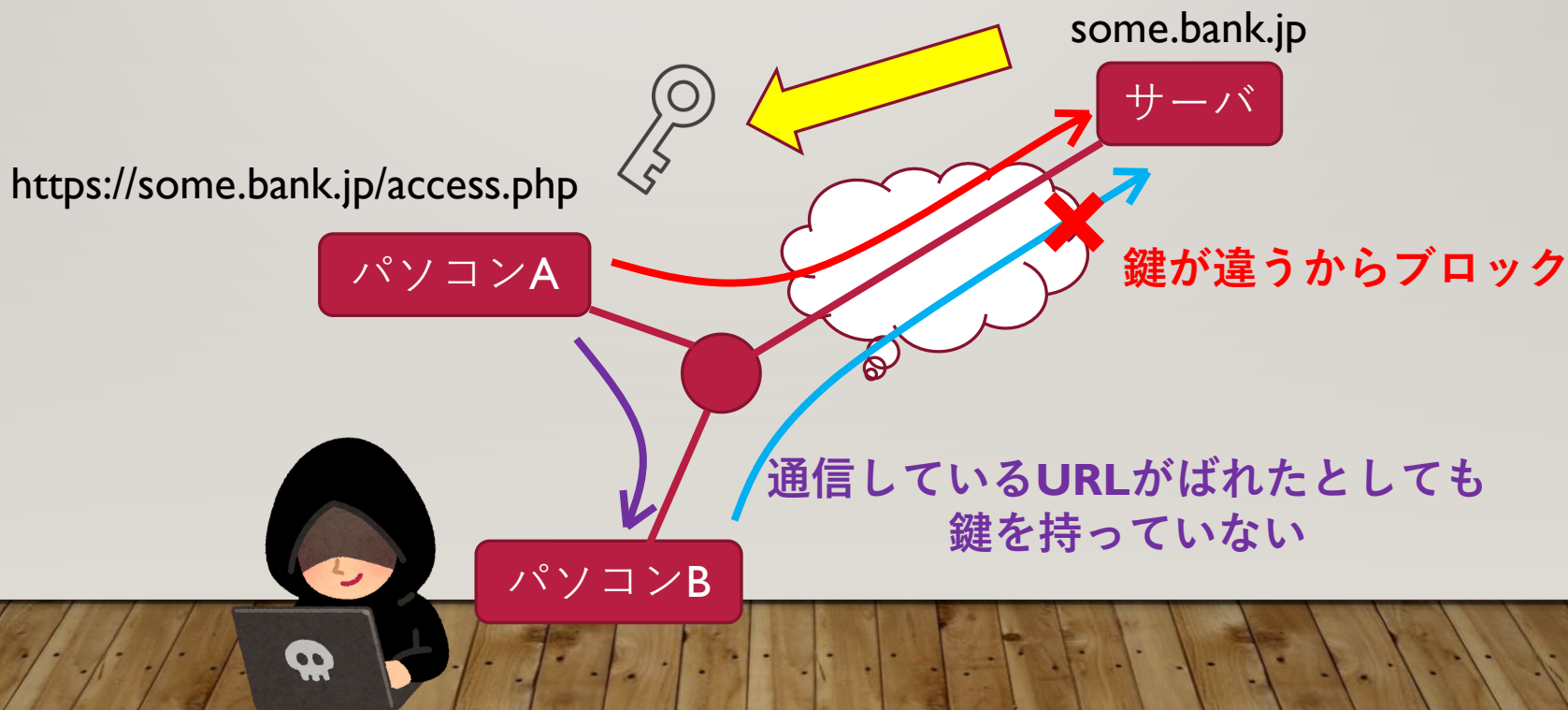
- http:// 通信は、通信内容が盗み見られるかも



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



HTMLインジェクション(I)

Sample Page 15 (様々な入力フォーム)

名前(text)	<input type="text"/>
パスワード(password)	<input type="password"/>
隠れた値(hidden)	<input type="hidden"/>
ご意見(textarea)	<div style="border: 1px solid blue; padding: 5px;">Home </div>

変数	値
\$_REQUEST["NAME"]	
\$_REQUEST["PASS"]	
\$_REQUEST["HIDDEN"]	隠れた値
\$_REQUEST["OPINION"]	Home

リンクとして
表示された...

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？

Sample Page 15 (様々な入力フォーム)

名前(text)	<input type="text"/>
パスワード(password)	<input type="password"/>
隠れた値(hidden)	
ご意見(textarea)	<div style="border: 1px solid blue; padding: 5px;"><code>Home</code></div>

‘<’ 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 やほおびーびー

Enter ID

送信

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

全データが
表示された

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}
```



SQLインジェクション攻撃(2)

個人情報漏洩の怖さ

え ...



- 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 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 「やばいんだけどなあ...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

- 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

A社

○先生

信用失墜



あんたのせいじゃ💢

メールで警告しましたよ

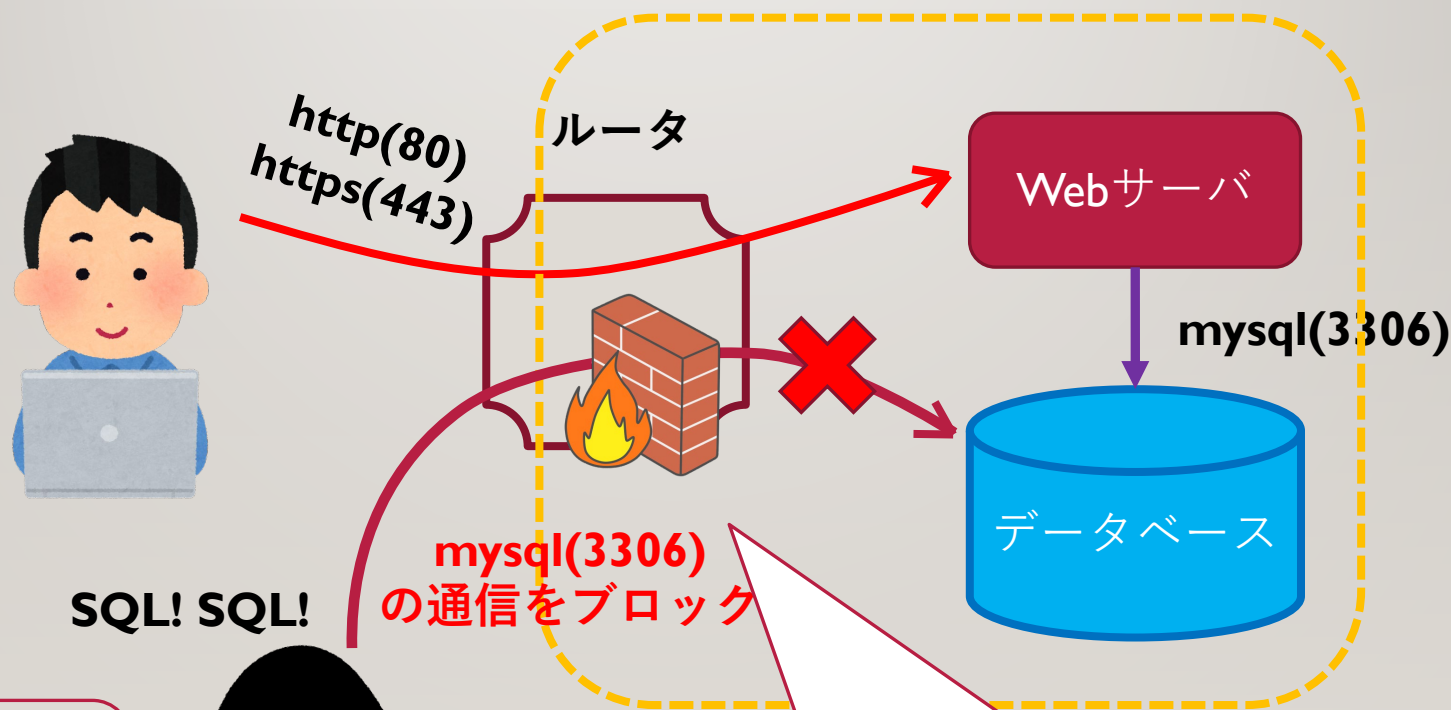
なんか揉めてるな...(笑)



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



データベース
サーバに直接
つながるぞ.

外部とデータベースが
繋がらないように
FireWall を設定!!

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