



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(10/29)



プログラミング応用 講義資料URL

<https://tsaitoh.net/~t-saitoh/2022-10-recp/>

login: guest

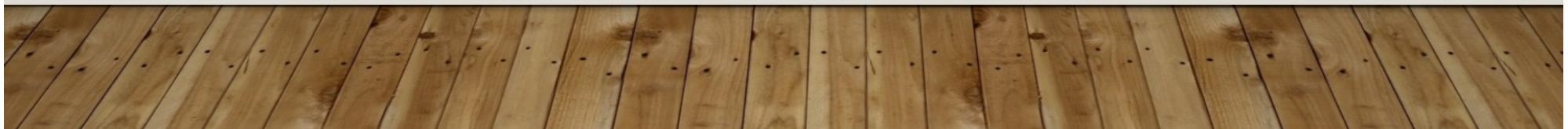
password: Guest

**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 | 講義内容と講義資料 |
|--|---|
| <ul style="list-style-type: none">Twitter @TohruSaitohFacebook tsaitoh.nettsaitoh.net@google.com  | <ul style="list-style-type: none">Webアプリケーションとプログラム言語(11/07)<ul style="list-style-type: none">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<ol style="list-style-type: none">理解度確認(11/07) (Google Formsに回答してください)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。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。 |

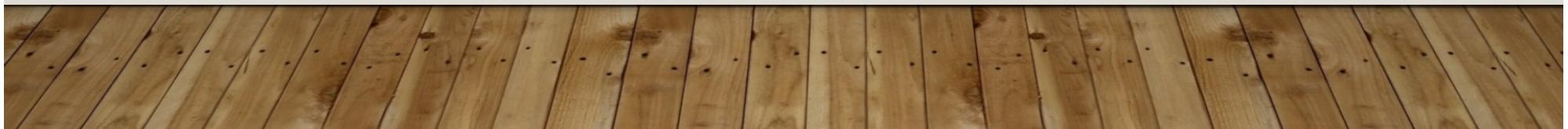
WEBプログラミングと セキュリティ

- ・ 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。



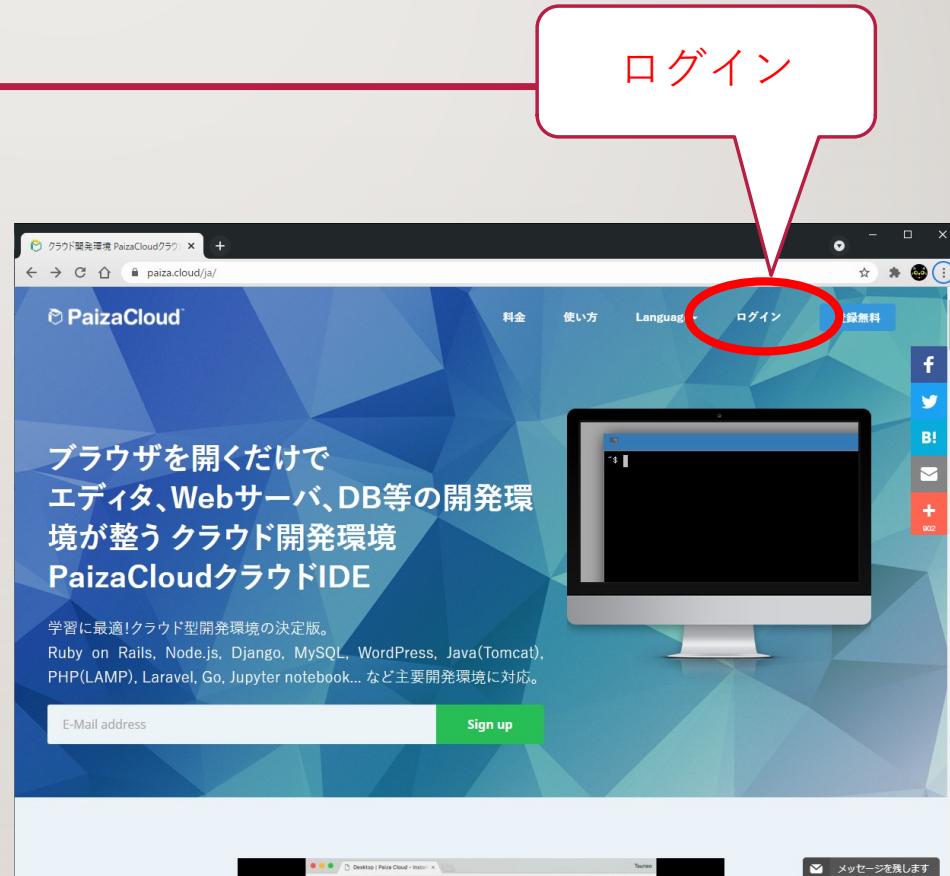
Paiza Cloud のサーバを使って インターネットの攻撃の怖さを体験

- **Paiza Cloud** の登録
- **Paiza Cloud** でサーバを作成
- **Paiza Cloud** の基本的な使い方

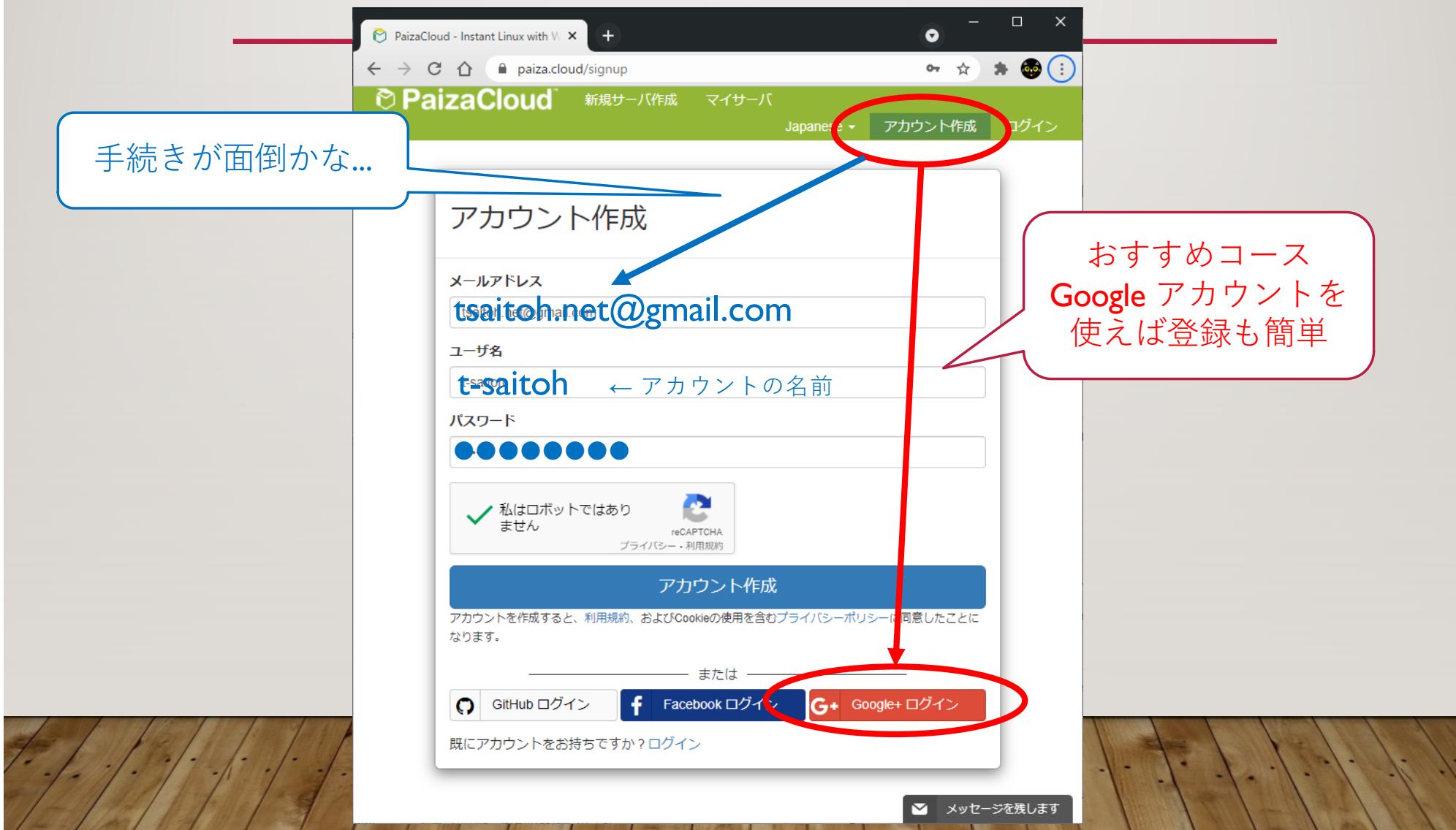


PAIZA CLOUD の登録(I)

- Webシステム構築の勉強用に
サーバを借りられる
- googleのメールアドレス
で登録
- <https://paiza.cloud/ja/>



PAIZA CLOUD の登録(2)



Paiza Cloud のサーバを作成

The screenshot shows two browser windows for PaizaCloud. The top window displays a green button labeled '新規サーバ作成' (Create New Server) which is circled in red. A red callout box contains the text: '無料コースで使うので
サーバは2時間使用すると
使えなくなります' (You can use it for free, so the server will be disabled after 2 hours of use). A red arrow points from this callout to the 'サーバ設定' (Server Settings) window in the bottom window. This window shows a 'サーバ名' (Server Name) input field containing 'tohrusaitoh'. Below it is a section for selecting initial installations and settings, with several options like Node.js, PHP, MySQL, and Apache listed.

まだサーバはありません。サーバを作成してください。

新規サーバ作成

無料コースで使うので
サーバは2時間使用すると
使えなくなります

新規サーバ作成

マイサーバ

サーバ設定

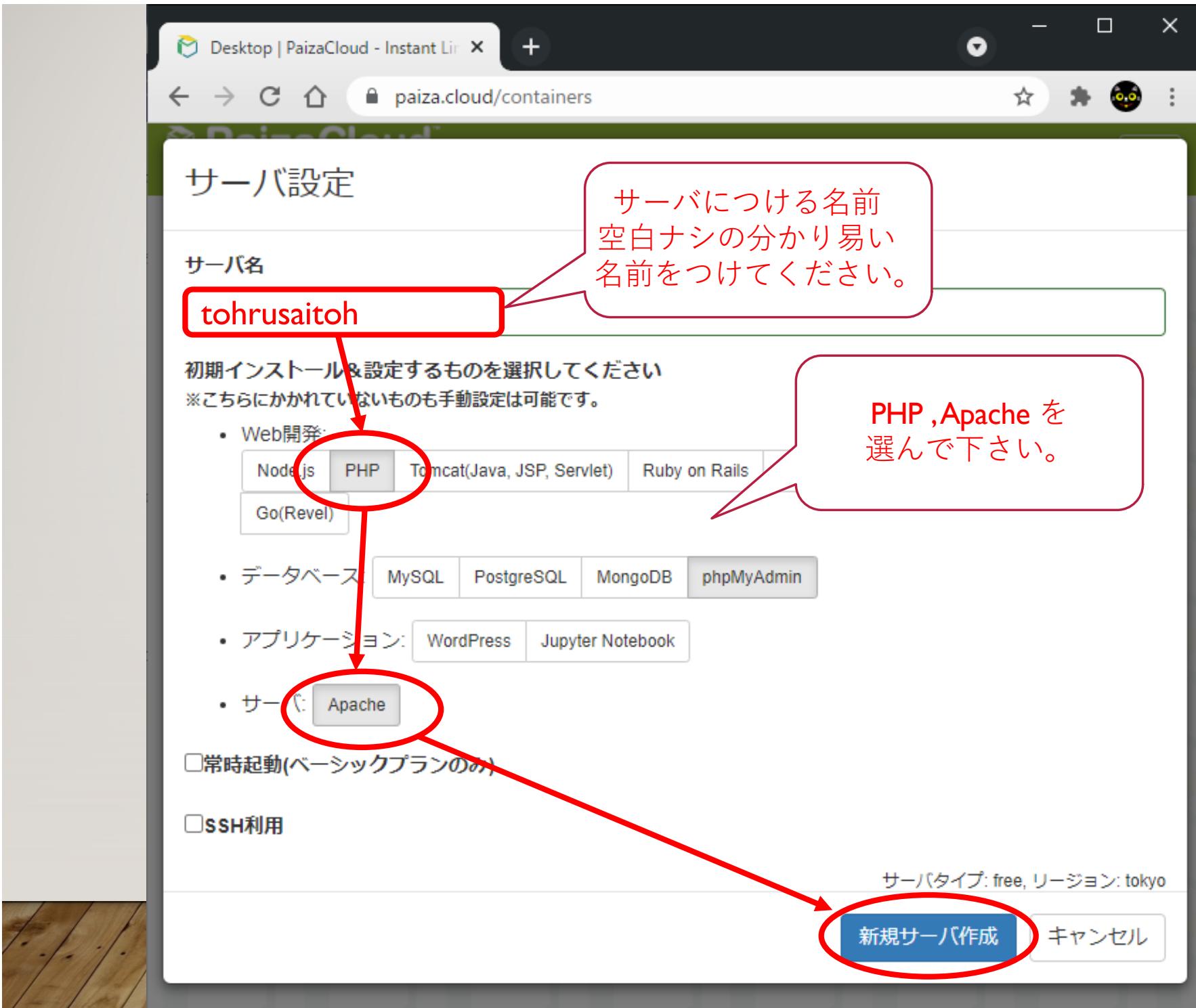
サーバ名

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ら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 Ruby Sinatra Django Laravel
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)



Paiza Cloud の最初の画面



教材データのダウンロード

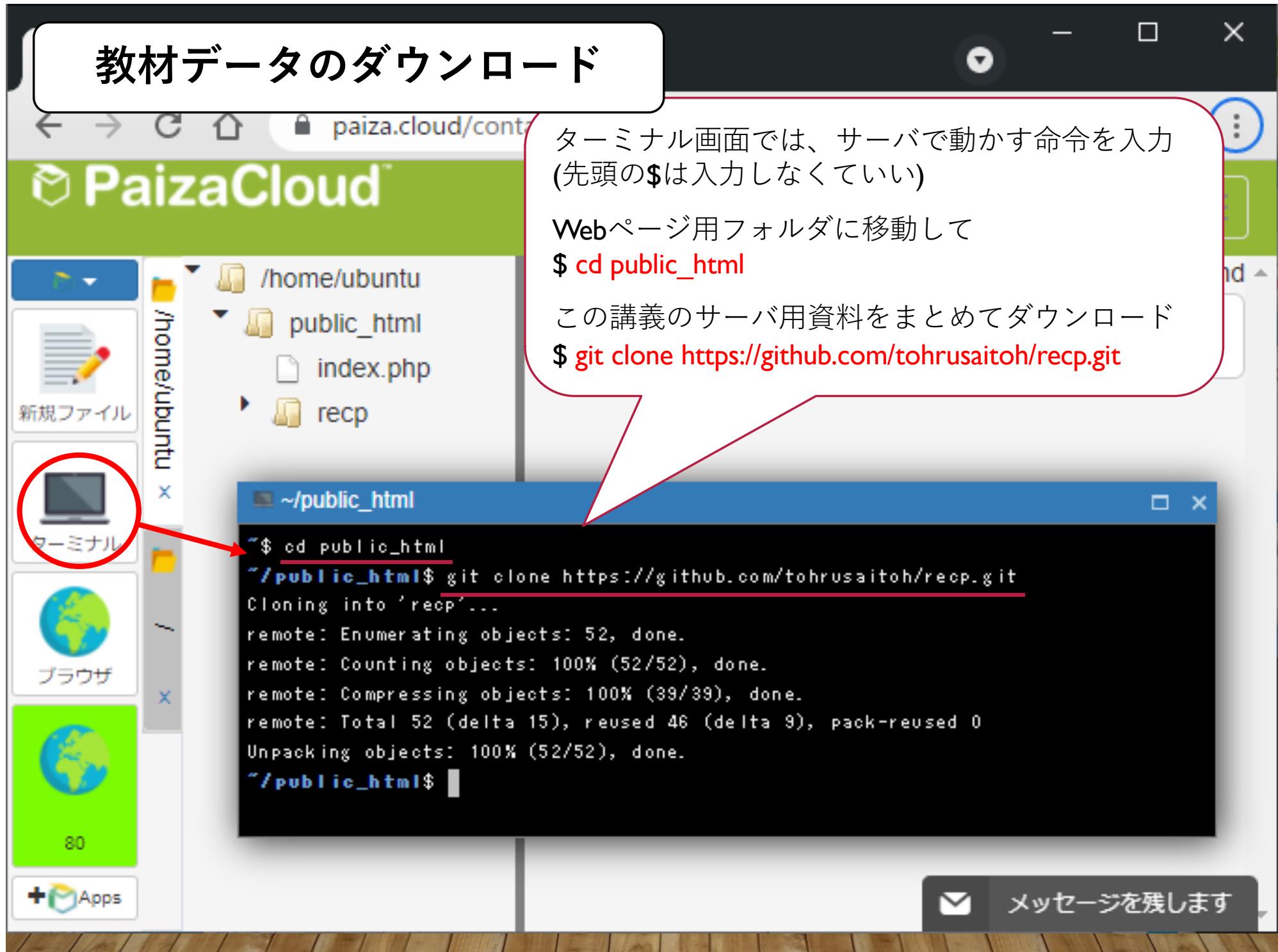
ターミナル画面では、サーバで動かす命令を入力
(先頭の\$は入力しなくていい)

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



演習で使うソフトを サーバにインストール

```
~/public_html/recp
~/public_html/recp$ sudo apt-get update
Get:1 https://dl.yarnpkg.com/debian stable InRelease [17.1 kB]
Err:2 https://apt.dockerproject.org/repo ubuntu InRelease
  Something wicked happened resolving 'apt.dockerproject.org'
  (-5 - No address associated with host name)
Ign:3 https://repo.mongodb.org/apt/ubuntu bionic InRelease
Get:4 https://cli-assets.heroku.com/apt ./ InRelease [2,550 B]
Hit:5 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu bionic In
~/public_html/recp$ sudo apt-get install nmap -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
nmap is already the newest version (7.60-1+deb9u1).
The following packages were newly installed:
  node-minimatch node-mkdirp node-mute-stream node-node-uuid
                                          
```

sudo は管理者権限で命令を実行

sudo apt-get update
利用可能なソフトの情報の更新

sudo apt-get install nmap -y
ポートスキャンツール nmap をインストール

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つてもらうために、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:

闇夜の鳥

Chromeなら「右ボタン」
「ページソース」を表示

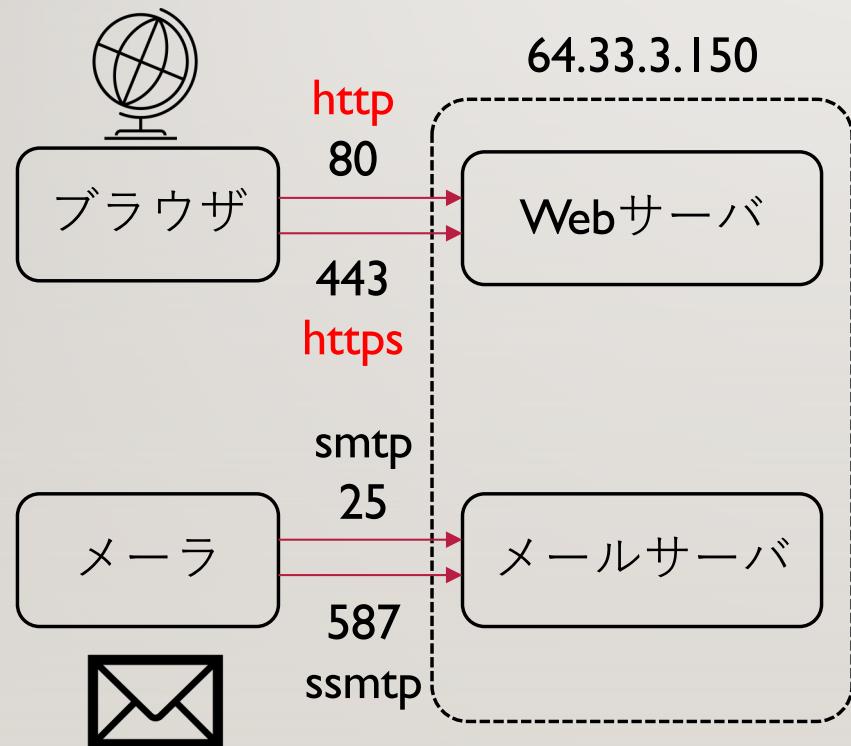
1. 暗号・コーディング系

- 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)

```
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>
```

ポート番号とプロトコル

IPアドレス = 電話番号
ポート番号 = 内線番号



- 一つのコンピュータの中で、通信する複数のプログラムを区別するためにポート番号を使う。
- 通信する時は、
IPアドレス:64.33.3.150
ポート番号:80 ... でつなごう
- ポート番号覚えにくいよね
名前を付けよう
- 80 : HTTP (Hyper Text Transfer Protocol)**
- 443: HTTPS (HTTPの暗号化版)**

サーバのプログラムに直接アクセス

```
■ ~public_html/recp
~/public_html/recp$ telnet tsaitoh.net 80
Trying 64.33.3.150...
Escape character is '^]'.
GET /
<html>
トップページを読むために
GET / <改行>
```

telnet は指定したIPアドレスに
指定したポート番号で接続

Web http は80番ポート

```
■ telnet tsaitoh.net 25 - ~/public_html/recp
telnet tsaitoh.net 25 - ~/public_html/recp
~/public_html/recp$ telnet tsaitoh.net 25
Trying 64.33.3.150...
Connected to tsaitoh.net.
Escape character is '^]'.
220 tsaitoh.net ESMTP unknown
QUIT
```

メール SMTP は25番ポート

Webサーバ専用のテキストブラウザ

```
~$ curl -s http://tsaitoh.net/
```

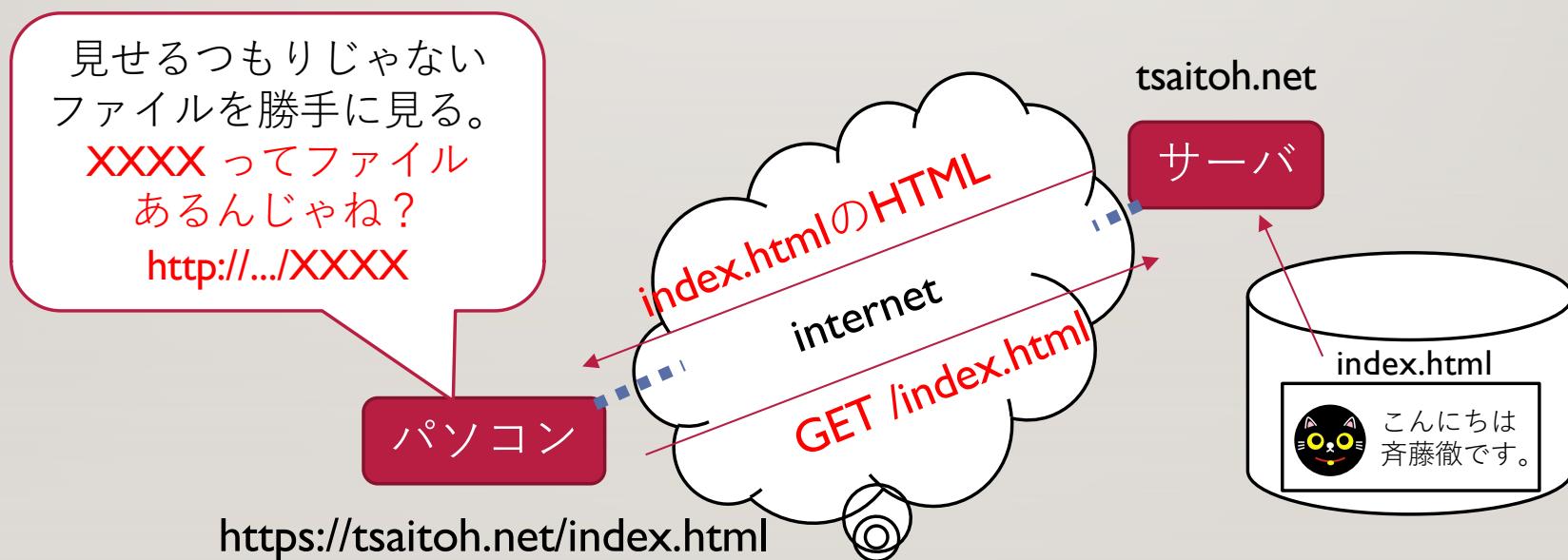
curl は 指定したURLのHTMLファイル
を直接ダウンロードできる

```
<html>
<head>
    <title>tsaitoh.net</title>
```

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 or C:\xampp\htdocs
- 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

Beware-Dot-Bak-File

あびばあ会員データベース

User-ID 送信

input user-id like t-saitoh

User-ID 送信

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 があるんじゃね？



URL トラバーサル攻撃(3) PHPプログラムの漏えい

The image shows two browser windows. The left window displays a form with a User-ID input field containing 't-saitoh'. The right window shows the raw PHP source code of the application.

User-ID 送信

input user-id like t-saitoh

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 == "flag") {  
        document.write("flag");  
    }  
}</script>  
</body>
```

https://tsaitoh.net/~t-saitoh/ctf/beware-dot-file.bak を開いてみる

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...

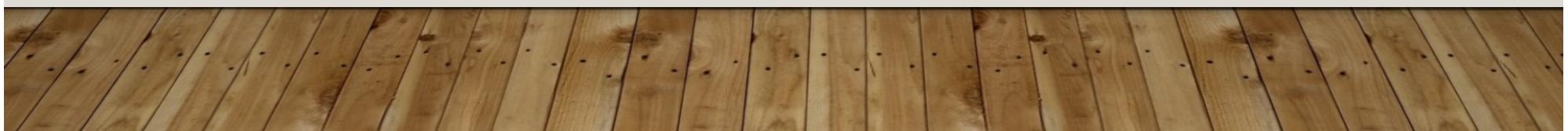
<https://tsaitoh.net/~t-saitoh/ctf/database.csv>
ってURL指定したら、
読めるんじゃね？

```
# -*- 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}
```

読みちゃった
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です？
 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
XAMPP なら C:\xampp\htdocs フォルダ内に保存しない
前回10/22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？



簡単なアクセス制限の方法

BASIC認証

- アクセス制限したいフォルダに `.htaccess` を置く

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

- パスワードファイル `.htpasswd`

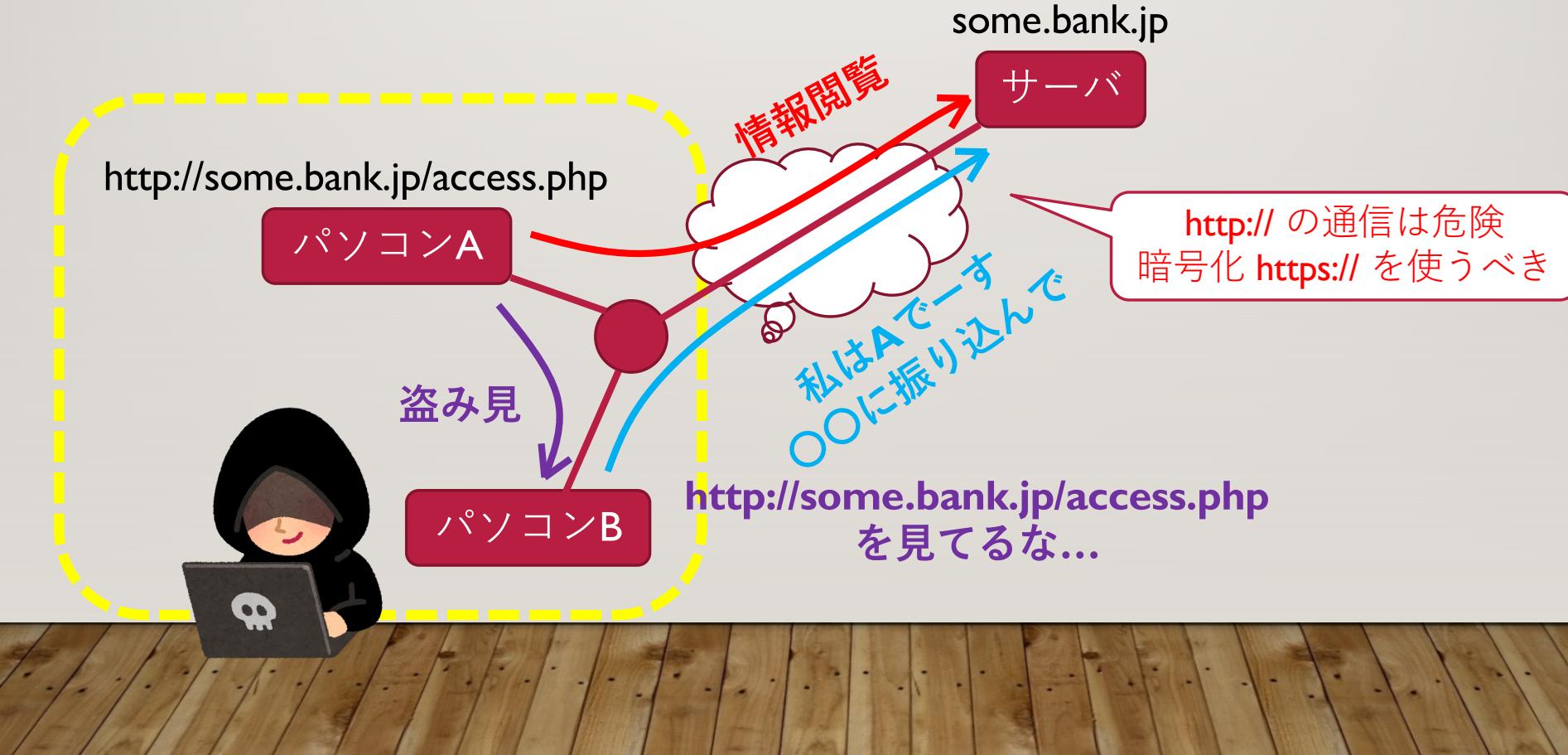
`guest:$apr1$1234c3U$AWn2wkw8hcl0ITS3txabce`

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

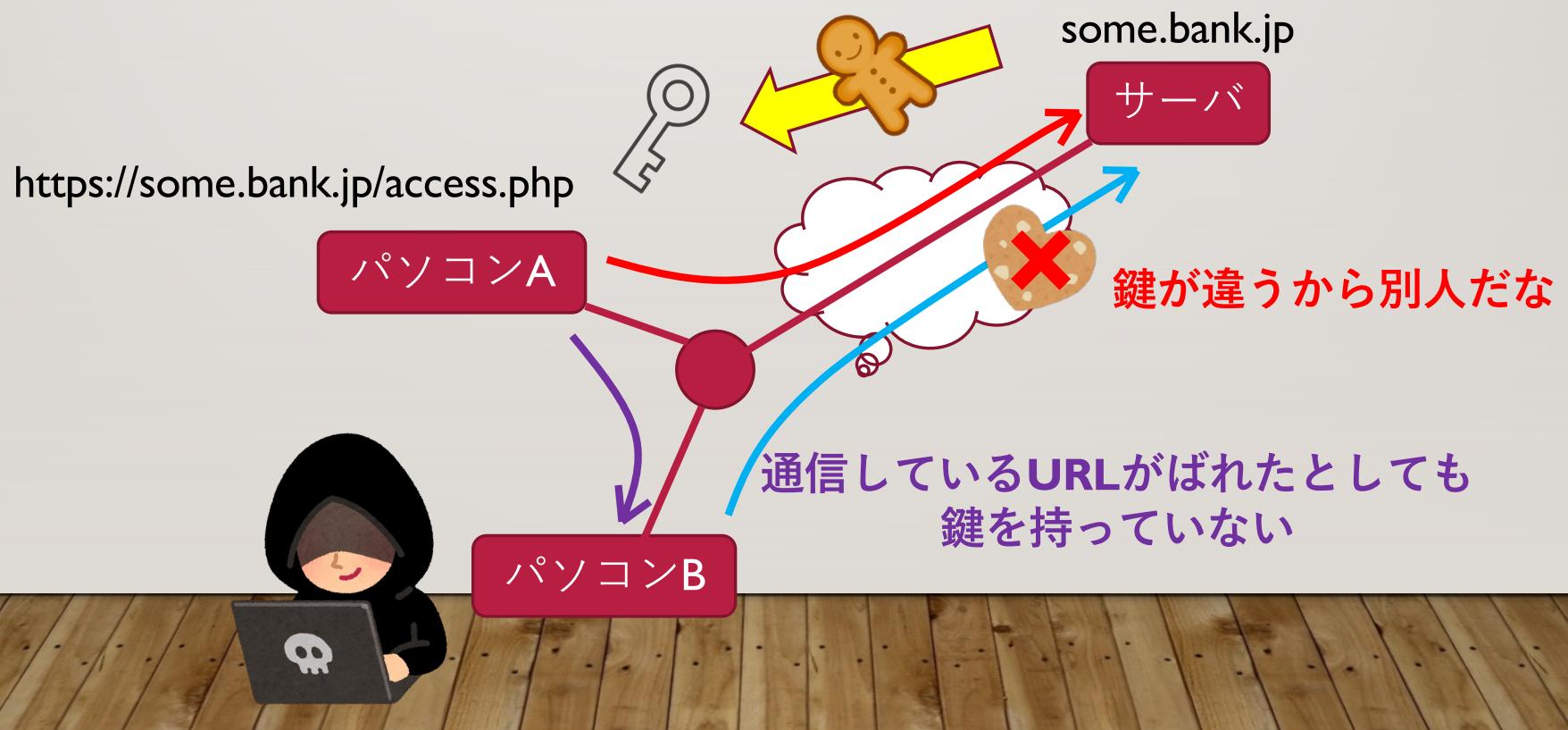
セッションハイジャック攻撃(I) http:// は危険

- http:// 通信は、通信内容が盗み見られるかも

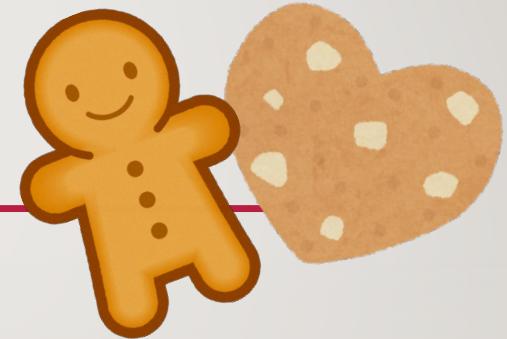


セッションハイジャック攻撃(2) セッションキーの発行

- **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



Cookieとトラッキング



- **1st Party Cookie**

- Cookieで購入情報を保存
- 訪問中のサイトから発行されるCookie

- **3rd Party Cookie**

- 見ているページに広告が表示される場合
- 広告業者から発行されるCookie
- ターゲッティング広告

- **ユーザのサイト閲覧情報を収集・分析**



- どこから接続しているのか分かる。
- ユーザがどこにいるか追跡できる。
- 不気味じゃね？

トラッキングが不気味と思って、Cookieを拒否したらセッション情報がとれなくてページ表示ができなくなる。

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"/> |

ブラウザのURL欄には
`http://.....sampleF.php?PASS=hogehoge&OPINION=...`

| 変数 | 値 |
|------------------------------------|----------|
| <code>\$_REQUEST["NAME"]</code> | |
| <code>\$_REQUEST["PASS"]</code> | hogehoge |
| <code>\$_REQUEST["HIDDEN"]</code> | 隠された値 |
| <code>\$_REQUEST["OPINION"]</code> | Home |

パスワードなどは、GETメソッドで送ってはいけない
`<form method="POST" action="sampleF.php">`

HTMLインジェクション(2)

Sample Page 15 (様々な入力フォーム)

| | |
|-----------------|--|
| 名前(text) | <input type="text"/> |
| パスワード(password) | ●●●●●●●●●● |
| 隠れた値(hidden) | <input type="hidden"/> |
| ご意見(textarea) | <pre>Home</pre> |

| 変数 | 値 |
|------------------------------------|----------|
| <code>\$_REQUEST["NAME"]</code> | |
| <code>\$_REQUEST["PASS"]</code> | hogehoge |
| <code>\$_REQUEST["HIDDEN"]</code> | 隠れた値 |
| <code>\$_REQUEST["OPINION"]</code> | 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(3) 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"/> |
| 隠された値(hidden) | <input type="hidden"/> |
| ご意見(textarea) | <div style="border: 1px solid blue; padding: 5px;">Home</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 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 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

- 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!



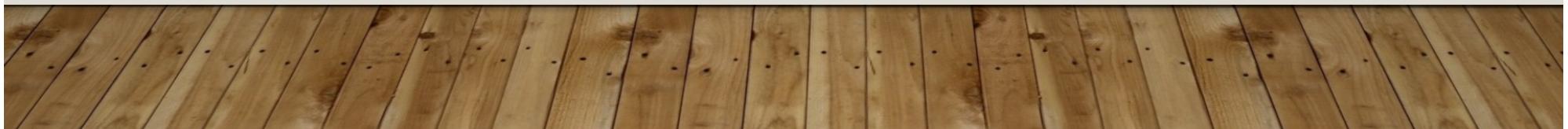
HTMLをダウンロードして
対策処理を消せばいいじゃん...

curl コマンドでサーバに
直接データ送ればいいじゃん...



サニタイジングとは

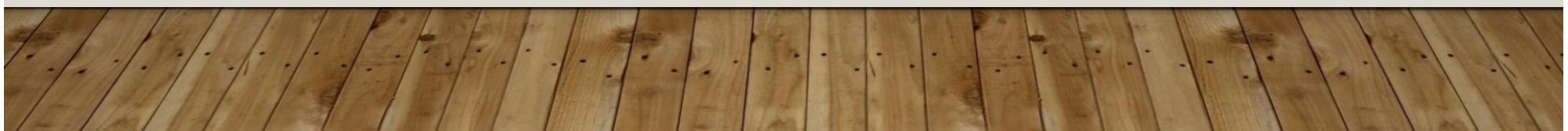
- 利用者が入力した文字データを受け取る際に、プログラムにとって特別な意味を持つ可能性のある文字や文字列を検知して、一定の規則に従って別の表記に置き換えること。
「無害化」とも呼ばれる。(IT用語辞典 e Wordsより引用)
 - ディレクトリトラバーサル 「/,.」
 - HTMLインジェクション 「<,>,&」
 - SQLインジェクション 「',,"」
- フロントエンド(JavaScript)での対策は不十分



個人情報漏洩の怖さ



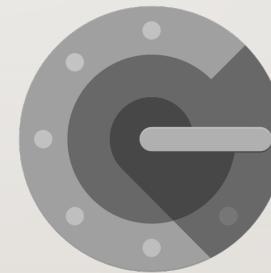
- 住所・氏名・年齢・職業の流出
 - 500円!??
- クレジットカードとPINコードの流出
 - 不正アクセスの恐怖
 - カード再発行の手間
- ログインIDとパスワードの流出
 - ディクショナリ攻撃の恐怖
 - 別のサービスに同じパスワードを使うのは危険



多要素認証の重要性



- パスワード以外の認証
 - 生体認証
 - 多要素認証
- 多要素認証
 - SMSを使ったPINコード認証
 - 認証アプリ
 - アプリで登録した鍵の元から時間で変化するPINコードを生成
 - セキュリティキー
 - 暗号で認証する鍵を発行する鍵
 - FIDO キー



Google Authenticatorの画面の例

fidoTM
ALLIANCE | simpler
stronger
authentication

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！」



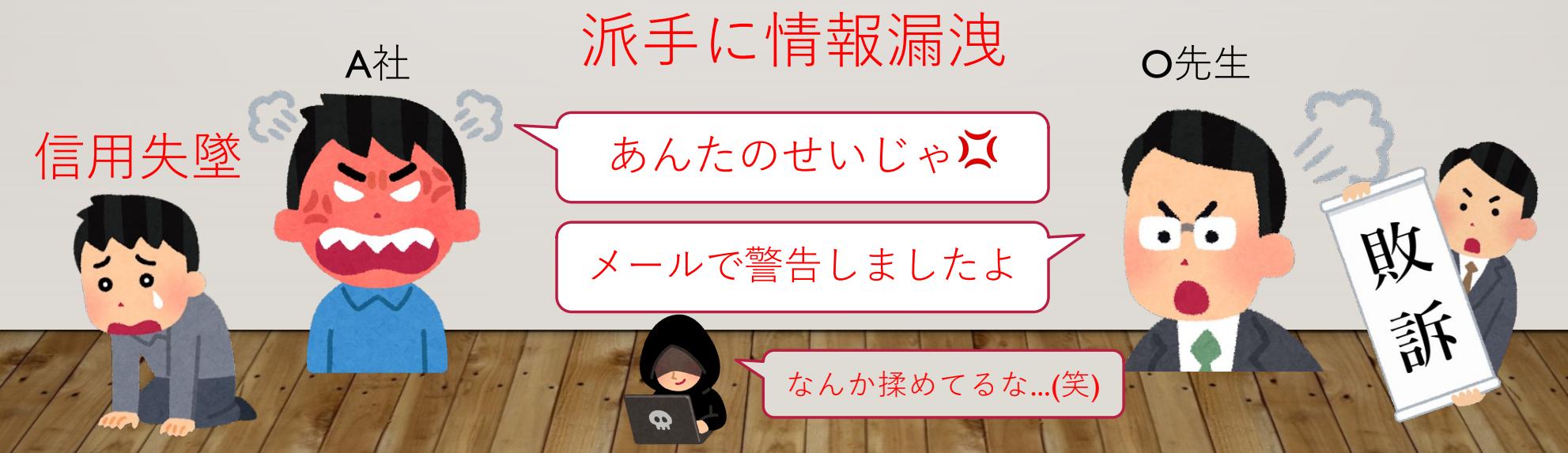
講習資料は、CD-ROMで配布しますね

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(ファイル共有ソフト)に流すか...」



CTF 4.3 数字4桁のパスワードを答えてください

パスワード認証すれば安全とは限らない
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 CTF challenge interface. On the left, there's a file browser window titled 'Paiza Cloud' showing a directory structure under '/home/ubuntu/public_html/recp'. A red circle highlights the 'ターミナル' (Terminal) icon in the sidebar. Inside the terminal, a session is running in the public_html/recp directory, executing 'cd public_html/recp' and 'bash brute-force-pin4.sh'. The output shows several attempts at a password, with '8087' being the correct one. A blue arrow points from the terminal output to a code editor window on the right. The code editor displays a bash script named 'brute-force-pin4.sh' with the following content:

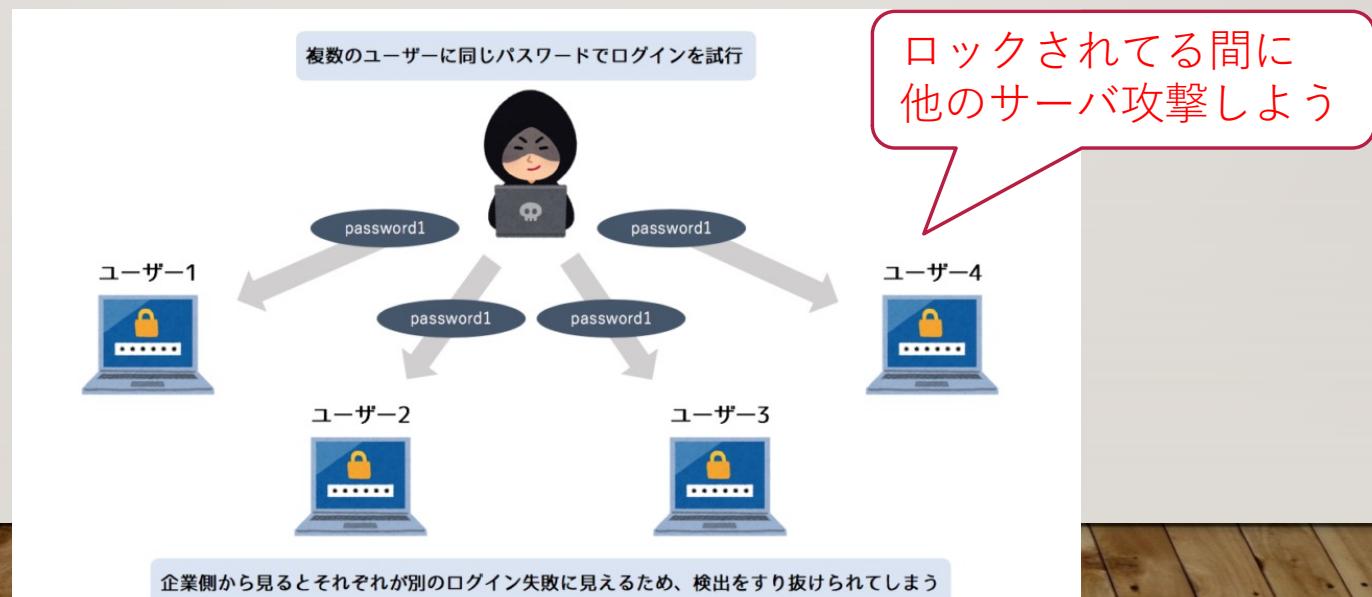
```
#!/bin/bash
for PW in `seq 8000 8090`;
do
    echo $PW
    curl -s http://localhost/~ubuntu/recp/brute-force-pin4.php?PW=$PW \
        | grep FLAG
done
```

A yellow arrow points from the terminal's password attempt '8087' to the 'PW' variable in the script. A red arrow points from the 'PW' variable to a callout box containing the text 'curl コマンドでPWに値を送る'.

ブルートフォース攻撃の一種 パスワードスプレー攻撃



- 一定期間に一定回数のログインエラーが起こると、アカウントが**一定時間ロック**される
- パスワードスプレー攻撃**はロックを避けるために、**アクセス間隔を空けてパスワード攻撃**を行う

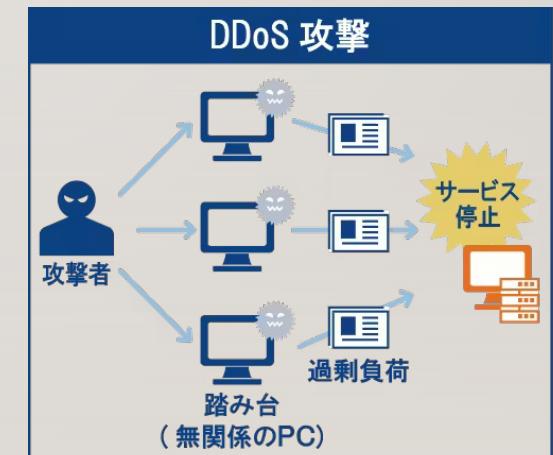


CyberSecurity社パスワードスプレー攻撃とは? (引用)

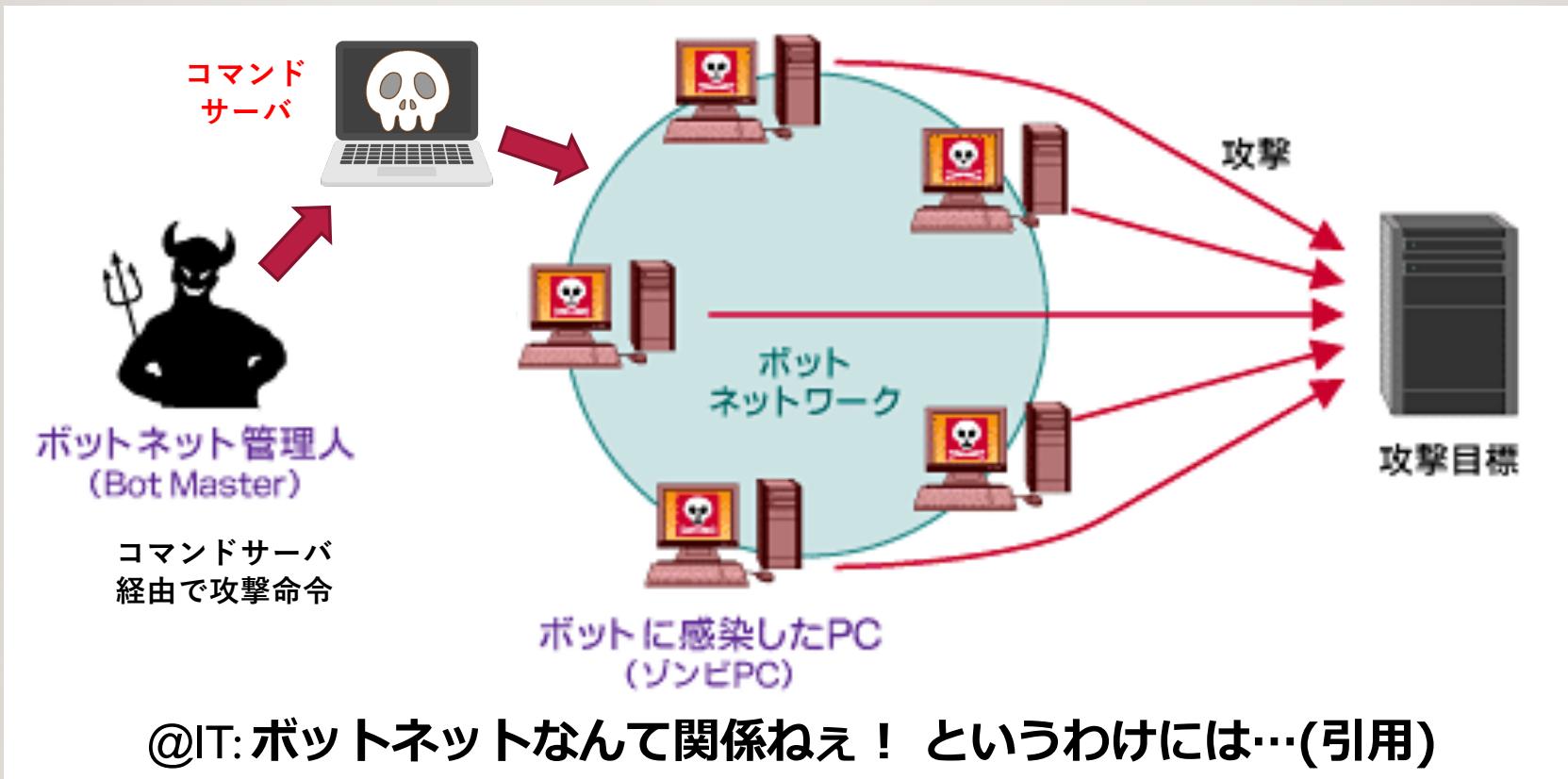
分散DoS攻撃

- サーバもクライアントの対応に処理能力が必要
- Webサーバに大量のアクセスを受けたらどうなる?
 - **F5アタック**
 - ブラウザのF5(リロード)を時間を合わせて実行
 - 大量アクセスで処理能力が低下
 - **DoS攻撃 Denial of Service**
 - サービスができなくなる攻撃
 - 特定のIPアドレスからの大量アクセスならFireWallでアクセス拒否も簡単

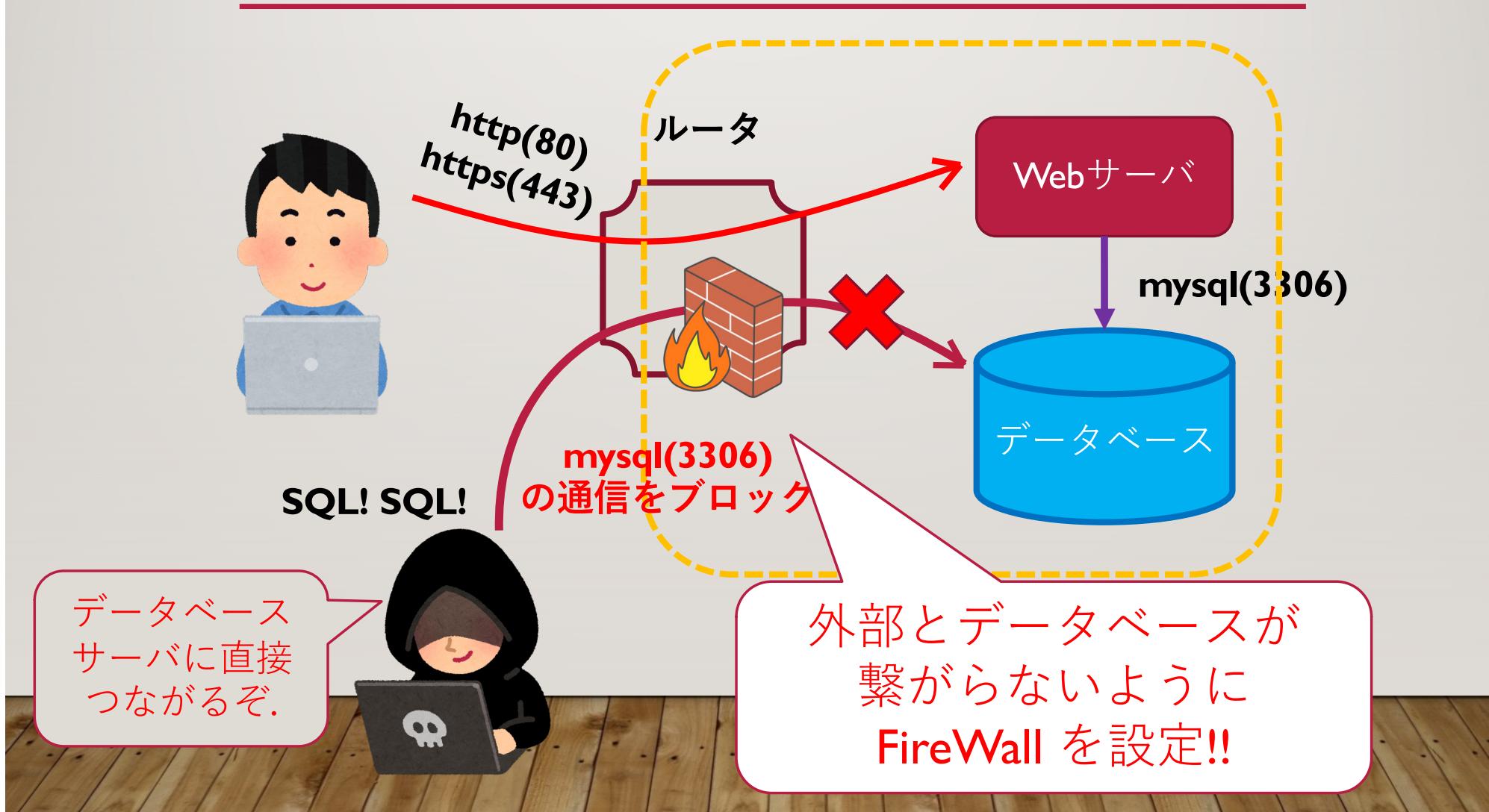
ウィルスに感染した大量のPCを使って
時間を決めてDoS攻撃を行えばいい



ボットネット(BotNet) ウィルス感染で操られるパソコン



ネットワークとFireWall



nmap はポート番号を変えながら
telnet を試すようなツール

ポートスキャンで 利用できるポートを探す

```
~/public_html/recp$ nmap tsaitoh.net

Starting Nmap 7.60 ( https://nmap.org ) at 2022-10-28 03:22
Nmap scan report for tsaitoh.net (64.33.3.150)
Host is up (0.012s latency).
rDNS record for 64.33.3.150: ttn64-33-3-150.ttn.ne.jp
Not shown: 991 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
113/tcp   closed ident
443/tcp   open  https
587/tcp   closed submis...
993/tcp   open  imaps
8080/tcp  closed http-proxy

Nmap done: 1 IP address (1 host up) scanned in 6.48 seconds
~/public_html/recp$
```

自宅内で nmap を動かすと
\$ nmap localhost
3306/tcp open mysql
↑
MySQLが見える

外部から
SMTP, HTTP, HTTPS
などに接続できる

nmap は攻撃者が使う定番のツール
相手に「攻撃者」と勘違いされる
可能性が高いので使用には要注意

FireWallの設定の例

さくらインターネットのクラウド

パケットフィルタ » Filter-FNCT

コピー 反映 パケットフィルタを削除

情報 ルール (17) 適用サーバ

| # | プロトコル | 送信元ネットワーク | 送信元ポート | ポート | アクション | 操作 | | |
|----|-------|------------------------------|--------|-------|-------|----|--|--|
| 1 | tcp | 192.156.145.0/24 福井高専(1) | * | 22 | allow | | | |
| 2 | tcp | 192.156.146.0/23 福井高専(2) | * | 22 | allow | | | |
| 3 | tcp | 192.156.148.0/23 福井高専(3) | * | 22 | allow | | | |
| 4 | tcp | 64.33.3.150 tsaitoh.net/斎藤自宅 | * | 22 | allow | | | |
| 11 | tcp | * sshの追加ポート | * | 10022 | allow | | | |
| 12 | tcp | * ssmtp | * | 465 | allow | | | |
| 13 | tcp | * http | * | 80 | allow | | | |
| 14 | tcp | * https | * | 443 | allow | | | |
| 15 | tcp | * Django? | * | 8000 | allow | | | |
| 16 | tcp | * smtp | * | 25 | allow | | | |
| 17 | tcp | * | * | 22 | deny | | | |

リモート接続の許可

Webやメールサーバへの接続許可

追加

| # | プロトコル | 送信元ネットワーク | 送信元ポート | ポート | アクション | 操作 | | |
|----|-------|------------------------------|--------|-------|-------|----|--|--|
| 1 | tcp | 192.156.145.0/24 福井高専(1) | * | 22 | allow | | | |
| 2 | tcp | 192.156.146.0/23 福井高専(2) | * | 22 | allow | | | |
| 3 | tcp | 192.156.148.0/23 福井高専(3) | * | 22 | allow | | | |
| 4 | tcp | 64.33.3.150 tsaitoh.net/斎藤自宅 | * | 22 | allow | | | |
| 11 | tcp | * sshの追加ポート | * | 10022 | allow | | | |
| 12 | tcp | * ssmtp | * | 465 | allow | | | |
| 13 | tcp | * http | * | 80 | allow | | | |
| 14 | tcp | * https | * | 443 | allow | | | |
| 15 | tcp | * Django? | * | 8000 | allow | | | |
| 16 | tcp | * smtp | * | 25 | allow | | | |
| 17 | tcp | * | * | 22 | deny | | | |

OSSオープンソースとライセンス

- オープンソースライセンス
 - ソフトウェアやそのソースコード、設計書の利用、修正、頒布を認めるソフトウェアのライセンスの総称
 - Linux, Apache などはインターネットの中で発達してきた



- 開発者はプログラムを**ダウンロードして改良**していい。
- ただし修正した内容は**インターネットに公開**する必要がある。

OSSライセンス違反



- オープンソースのプログラムを改良して使ったのに
インターネットに公開しない
- OSS ライセンス違反
- 事例
 - FANTECのメディアプレーヤーがGPLv2のOSSを利用
 - 開示コードに明記されていなかった
 - 訴訟・罰金・信用低下



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

