提出日：2024年5月17日

先進情報プロジェクト演習

テーマIT・1 第４回レポート

学籍番号：C0A22113

氏名：成田彩華

# CGIプログラムにおけるデータファイルの設置場所

第１回講義で作成した「/var/www/html/test/bbs.cgi」について考える。以下のURLにローカル環境からアクセスし、掲示板のプログラムが正しく動作していることを確認する。

<http://192.168.64.7/test/bbs.cgi>

5行目が以下のようになっていることを確認する。

$DATAFILE = ‘./log.txt’;

次のようなURLでアクセスするとどうなるか確認する。

<http://192.168.64.7/test/log.txt>

このURLにアクセスするとログファイルの中身が表示される。しかし、このように露ファイルが見れてしまうことはセキュリティ的に問題である。

/var/www/html/testにindex.htmlがあることを確認する。ない場合には、/var/www/htmlにあるindex.htmlをコピーして作成する。次のようなURLでアクセスするとどうなるかを確認する。

<http://192.168.64.7/test/>

このURLにアクセスするとindex.htmlが表示される。ここで、index.htmlを別の名前に変え、同じURLにアクセスするとどうなるのか確認する。index.htmlが存在しない場合、/var/www/html/test/内のファイル一覧を表示させる。

<http://192.168.64.7/test/log.txt>でログファイルにアクセスできることが知られてしまったのかを考える。log.txtがある場所はドキュメントルートから辿れるtest直下にある。もし、index.htmlが存在しない場合、フォルダの中身が見れてしまうため、ログファイルがあることが知られてしまう。パーミッションを適切に設定していない場合は、log.txtを直接閲覧することが可能であると教えてしまっているようなものである。

/var/www/html/testのパーミッションを変更する。www-dataに対して、「--x」になるよう設定する。設定したのち、以下のURLにアクセスする。

<http://192.168.64.7/test/>

アクセスするとForbidden You don’t have permission to access this resource.と言われる。これは、パーミッション設定をしたことにより、このファイルに対して「ls」コマンドを実行する権限がなくなってしまったためである。しかし、実行権限はあるため、「cd」コマンドは実行できる。

パーミッション変更後の安全性を考えていく。ログファイルがルートディレクトリ上にある場合、パーミッション変更後でも「cd」コマンドは有効であるため、ログファイルにアクセスできてしまう。そのため、安全とはいえない。

# CGIプログラムにおけるデータファイルの適切な設置場所

bbs.cgiのlog.txtを、ユーザのホームディレクトリの下に持ってくる。ユーザのホームディレクトリの確認は、以下のコマンドで確認できる。

$ cd

$ pwd

/home/ayaka

このように、ホームディレクトリが/home/ayaka/であることがわかる。ここに、/var/www/html/test/log.txtをmvコマンドで移動させる。

$ mv /var/www/html/test/log.txt /home/ayaka/

移動させたlog.txtを読み込めるようにするため、bbs.cgiを書き換える。5行目を以下のように書き換える。

$DATAFILE ‘/home/ayaka/log.txt’;

log.txtパーミッションを考える。このファイルはログファイルであるため、読み書きができる必要がある。読み書きするのは、www-dataであるため、www-dataに対して、「rw-」になるようにパーミッションを設定する。

# 予約システムのPHPを動かす

講義ページからreservation.phpとreservation\_main.phpをダウンロードし、/var/www/html/mailに設置する。この2つのファイルのパーミッションを考える。この2つのファイルはwww-dataが読み込める必要があるため、パーミッションはwww-data「r--」になるよう設定する。

# Postfixのインストール

以下のコマンドを入力してPostfixのインストールを行う。

$ sudo apt install postfix

「Please select the mail server configuration type that best meets your nees.(あなたの用途にあったメールサーバ設定形式を選んでください。)」という画面が表示されたら、Tabキーで「<了解>」を選択する。

「General mail configuration type(メール設定の一般形式)」は「インターネットサイト」にしておく。

「System mail name(システムメール名)」は、書かれているまま(vm-ayaka)にしておく。

# PHPからのメール送信

reservation.phpをreservation2.phpとして複製する。また、reservation\_main.phpをreservation\_main2.phpとして複製する。

$ cp /var/www/html/mail/reservation.php /var/www/html/mail/reservation2.php

$ cp /var/www/html/mail/reservation\_main.php /var/www/html/mail/reservation\_main2.php

reservation2.phpが呼び出すものがreservation\_main2.phpになるようにする。36行目を以下のように変更する。

header(‘Location:reservation\_main2.php’);

reservation\_main2.phpの6行目の「$address = \_SESSION[‘address’];」の下に、次の内容を追加して、予約内容が送信されるようにする。「$to」はそのLinuxのrootにする。

$to      = 'root@localhost';

$subject = 'reservation';

$message = 'reservation:' . $user\_name . ', ' . $phone . ', ' . $address;

$headers = array(

    'From' => 'ayaka@vm--ayaka',

    'Reply-To' => 'ayaka@vm-ayaka',

    'X-Mailer' => 'PHP/' . phpversion()

);

mail($to, $subject, $message, $headers);

# メールが届いたかの確認

<http://192.168.64.7/mail/reservation2.php> にアクセスし、メールを送信する。/var/mail/内にrootというファイルがあるか確認する。

$ sudo cat root

以下のような内容のファイルが確認できる。

From www-data@vm-ayaka Fri May 10 17:51:24 2024

Return-Path: <www-data@vm-ayaka>

X-Original-To: root@localhost

Delivered-To: root@localhost

Received: by vm-ayaka (Postfix, from userid 33)

id 5E6EAE8569; Fri, 10 May 2024 17:51:24 +0900 (JST)

To: root@localhost

Subject: reservation

From: ayaka@vm-ayaka

Reply-To: ayaka@vm-ayaka

X-Mailer: PHP/8.1.2-1ubuntu2.17

Message-Id: <20240510085124.5E6EAE8569@vm-ayaka>

Date: Fri, 10 May 2024 17:51:24 +0900 (JST)

reservation:090, user, tokyo

# PHPからのメールの外部への送信

reservation.phpをreservation3.phpとして複製する。また、reservation\_main.phpをreservation\_main3.phpとして複製する。

$ cp /var/www/html/mail/reservation.php /var/www/html/mail/reservation3.php

$ cp /var/www/html/mail/reservation\_main.php /var/www/html/mail/reservation\_main3.php

reservation3.phpが呼び出すものがreservation\_main3.phpになるようにする。36行目を以下のように変更する。

header('Location:reservation\_main3.php');

reservation\_main3.phpの6行目「$address = $\_SESSION['address'];」の下に以下の内容を追加する。「$to」には自分のGmailのメールアドレスを書く。

$to      = 'c0a221130b@edu.teu.ac.jp';

$subject = 'reservation';

$message = 'reservation:' . $user\_name . ', ' . $phone . ', ' . $address;

$headers = array(

    'From' => 'ayaka@vm-ayaka',

    'Reply-To' => 'ayaka@vm-ayaka',

    'X-Mailer' => 'PHP/' . phpversion()

);

mail($to, $subject, $message, $headers);

# 外部にメールが届いたかの確認

Gmialを確認する。以下のようなメールが届いていることが確認できる。

テキスト

自動的に生成された説明

# tcpdumpを使ってWebサーバへの通信内容を見る

tcpdumpのオプションについて理解する。「-A」はキャプチャデータのASCII表示。「src host」で送信元ホスト、「dst host」で送信先ホストを指定できる。「scr port」で送信元ポート、「dsp port」で送信先ポートを指定できる。

Webサーバへの通信内容を見る。reservation2.phpから情報を送信してみる。以下のコマンドで通信内容を確認していく。

$ sudo tcpdump -A dst port 80 and dst host 192.168.64.7

tcpdump: verbose output suppressed, use -v[v]... for full protocol decode

listening on enp0s1, link-type EN10MB (Ethernet), snapshot length 262144 bytes

21:01:29.341342 IP \_gateway.tfido > vm-ayaka.http: Flags [SEW], seq 2872049110, win 65535, options [mss 1460,nop,wscale 6,nop,nop,TS val 3098238113 ecr 0,sackOK,eol], length 0

E..@....@.y\_..@...@....P./...........-.............

..\.........

21:01:29.341816 IP \_gateway.tfido > vm-ayaka.http: Flags [.], ack 72050365, win 2058, options [nop,nop,TS val 3098238113 ecr 1660110805], length 0

E..4....@.yk..@...@....P./...Kf....

.......

..\.b.G.

21:01:29.342161 IP \_gateway.tfido > vm-ayaka.http: Flags [P.], seq 0:569, ack 1, win 2058, options [nop,nop,TS val 3098238114 ecr 1660110805], length 569: HTTP: POST /mail/reservation2.php HTTP/1.1

E..m....@.w0..@...@....P./...Kf....

.......

..\.b.G.POST /mail/reservation2.php HTTP/1.1

Host: 192.168.64.7

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Encoding: gzip, deflate

Accept-Language: ja

Content-Type: application/x-www-form-urlencoded

Origin: http://192.168.64.7

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/17.4.1 Safari/605.1.15

Connection: keep-alive

Upgrade-Insecure-Requests: 1

Referer: http://192.168.64.7/mail/reservation2.php

Content-Length: 38

Cookie: PHPSESSID=287r4ctqan5b45cg9qo148169p

21:01:29.342556 IP \_gateway.tfido > vm-ayaka.http: Flags [P.], seq 569:607, ack 1, win 2058, options [nop,nop,TS val 3098238114 ecr 1660110805], length 38: HTTP

E..Z....@.yC..@...@....P.0...Kf....

.......

..\.b.G.user\_name=user&phone=090&address=tokyo

21:01:29.344705 IP \_gateway.tfido > vm-ayaka.http: Flags [.], ack 350, win 2053, options [nop,nop,TS val 3098238117 ecr 1660110808], length 0

E..4....@.yk..@...@....P.0.6.Kh............

..\.b.G.

21:01:29.349316 IP \_gateway.tfido > vm-ayaka.http: Flags [P.], seq 607:1082, ack 350, win 2053, options [nop,nop,TS val 3098238122 ecr 1660110808], length 475: HTTP: GET /mail/reservation\_main2.php HTTP/1.1

E.......@.w...@...@....P.0.6.Kh............

..\.b.G.GET /mail/reservation\_main2.php HTTP/1.1

Host: 192.168.64.7

Cookie: PHPSESSID=287r4ctqan5b45cg9qo148169p

Connection: keep-alive

Upgrade-Insecure-Requests: 1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/17.4.1 Safari/605.1.15

Referer: http://192.168.64.7/mail/reservation2.php

Accept-Encoding: gzip, deflate

Accept-Language: ja

21:01:29.367577 IP \_gateway.tfido > vm-ayaka.http: Flags [.], ack 854, win 2045, options [nop,nop,TS val 3098238140 ecr 1660110831], length 0

E..4....@.yk..@...@....P.0...Kj............

..\.b.G.

この通信内容のうち、reservation2.phpから送信した場合について、キャプチャした内容のうち、Webブラウザから入力されて送信された内容は以下の通りである。

21:01:29.342556 IP \_gateway.tfido > vm-ayaka.http: Flags [P.], seq 569:607, ack 1, win 2058, options [nop,nop,TS val 3098238114 ecr 1660110805], length 38: HTTP

E..Z....@.yC..@...@....P.0...Kf....

.......

..\.b.G.user\_name=user&phone=090&address=tokyo

「/var/www/html/test」に作成した、bbs.cgiから送信した場合について、キャプチャした内容は次の通りである。

$ sudo tcpdump -A dst port 80 and dst host 192.168.64.7

tcpdump: verbose output suppressed, use -v[v]... for full protocol decode

listening on enp0s1, link-type EN10MB (Ethernet), snapshot length 262144 bytes

03:15:01.235080 IP \_gateway.60999 > vm-ayaka.http: Flags [SEW], seq 3026577746, win 65535, options [mss 1460,nop,wscale 6,nop,nop,TS val 3344115281 ecr 0,sackOK,eol], length 0

E..@....@.y\_..@...@..G.P.e.R.......................

.S&Q........

03:15:01.235487 IP \_gateway.60999 > vm-ayaka.http: Flags [.], ack 4055495302, win 2058, options [nop,nop,TS val 3344115281 ecr 1664753489], length 0

E..4....@.yk..@...@..G.P.e.S.......

.......

.S&Qc:.Q

03:15:01.235789 IP \_gateway.60999 > vm-ayaka.http: Flags [P.], seq 0:551, ack 1, win 2058, options [nop,nop,TS val 3344115281 ecr 1664753489], length 551: HTTP: POST /test/bbs.cgi HTTP/1.1

E..[....@.wB..@...@..G.P.e.S.......

k'.....

.S&Qc:.QPOST /test/bbs.cgi HTTP/1.1

Host: 192.168.64.7

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Encoding: gzip, deflate

Accept-Language: ja

Content-Type: application/x-www-form-urlencoded

Origin: http://192.168.64.7

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/17.4.1 Safari/605.1.15

Connection: keep-alive

Upgrade-Insecure-Requests: 1

Referer: http://192.168.64.7/test/bbs.cgi

Content-Length: 32

Cookie: PHPSESSID=287r4ctqan5b45cg9qo148169p

03:15:01.236186 IP \_gateway.60999 > vm-ayaka.http: Flags [P.], seq 551:583, ack 1, win 2058, options [nop,nop,TS val 3344115282 ecr 1664753489], length 32: HTTP

E..T....@.yI..@...@..G.P.e.z.......

.......

.S&Rc:.Qtitle=test&author=user&text=test

03:15:01.242197 IP \_gateway.60999 > vm-ayaka.http: Flags [.], ack 889, win 2045, options [nop,nop,TS val 3344115288 ecr 1664753496], length 0

E..4....@.yk..@...@..G.P.e.................

.S&Xc:.X

^C

5 packets captured

5 packets received by filter

0 packets dropped by kernel

このキャプチャ内容のうち、Webブラウザから入力されて送信された内容は以下の通りである。

03:15:01.236186 IP \_gateway.60999 > vm-ayaka.http: Flags [P.], seq 551:583, ack 1, win 2058, options [nop,nop,TS val 3344115282 ecr 1664753489], length 32: HTTP

E..T....@.yI..@...@..G.P.e.z.......

.......

.S&Rc:.Qtitle=test&author=user&text=test

# tcpdumpを使ってSMTPへの通信内容をみる

reservation2.phpから情報を送信してみる。以下のコマンドを打ち込みパケットをキャプチャする。

$ sudo tcpdump -A

メールに関して、それらしき内容が確認できない。WebサーバがSMTPに対応していないため、メールに関してのパケットが発生しない。

reservation3.phpから送信した場合について、以下にメールに関してのキャプチャ内容の一部を示す。

$ sudo tcpdump -A

14:28:20.136225 IP vm-ayaka.41462 > tc-in-f27.1e100.net.smtp: Flags [P.], seq 1:16, ack 74, win 502, options [nop,nop,TS val 3707512711 ecr 1754749980], length 15: SMTP: EHLO vm-ayaka

E..C..@.@.r...@...........wu..\*&....8......

..'.h.\.EHLO vm-ayaka

14:28:20.306291 IP tc-in-f27.1e100.net.smtp > vm-ayaka.41462: Flags [P.], seq 74:241, ack 16, win 256, options [nop,nop,TS val 1754750146 ecr 3707512711], length 167: SMTP: 250-mx.google.com at your service, [163.215.6.1]

E....}..x.........@.......\*&..w............

h.\...'.250-mx.google.com at your service, [163.215.6.1]

250-SIZE 157286400

250-8BITMIME

250-STARTTLS

250-ENHANCEDSTATUSCODES

250-PIPELINING

250-CHUNKING

250 SMTPUTF8

reservation3.phpから送信した場合、メールの本文がキャプチャできないのは、SMTPで送る際、googleのサーバで暗号化通信が行われているため、通信内容自体が見れないためである。

PHPによる送信方法が安全でない理由について考える。SMTPで送る際、いくつものサーバを経由する。経由するサーバがSMTPに対応していない場合、暗号化されずに通信が行われてしまう。また、SMTPにサーバが対応していても、経由するサーバに悪意のある何者かが侵入していた場合、通信内容を抜かれてしまう。そのため、PHPによるこの送信方法が安全でない。

# ディレクトリ・トラバーサルを試す

まず、「/var/www/html/dirtra」に「view.php」を作成し、「/var/www/html/mail」にある「reservation.php」の内容をコピーする。

「user\_name」を「item\_name」に変更し、「ユーザ名」を「アイテム名」に変更する。ユーザ名以外の電話番号と住所についてはプログラム上から削除し、「アイテム名」に身について入力して送信されるようにする。送信先を「reservation\_main.php」から「view\_main.php」に変更し、HTMLのタイトルも「予約」から「アイテム表示」に変更する。送信先を「reservation\_main.php」から「view\_main.php」に変更し、HTMLのタイトルも「予約」から「アイテム表示」に変更する。変更後のview.phpは以下の通りである。

<?php

ini\_set('display\_errors', true);

error\_reporting(E\_ALL);

function h($string)

{

return htmlspecialchars($string, ENT\_QUOTES, 'utf-8');

}

session\_start();

// エラーを格納する変数

$err = [];

// 「ログイン」ボタンが押されて、POST通信のとき

if (filter\_input(INPUT\_SERVER, 'REQUEST\_METHOD') === 'POST') {

$item\_name = filter\_input(INPUT\_POST, 'item\_name');

if ($item\_name === '') {

$err['item\_name'] = 'アイテム名は入力必須です。';

}

// エラーがないとき

if (count($err) === 0) {

$\_SESSION['item\_name'] = $item\_name;

header('Location:view\_main.php');

return;

}

}

?>

<!DOCTYPE HTML>

<html lang="ja">

<head>

<meta charset="UTF-8">

<title>アイテム表示</title>

<style type="text/css">

.error {

color: red;

}

</style>

</head>

<body>

<div id="wrapper">

<form action="" method="post">

<p>

<label for="">アイテム名</label>

<input id="item\_name" name="item\_name" type="text" />

<?php if (isset($err['item\_name'])) : ?>

<p class="error"><?php echo h($err['item\_name']); ?></p>

<?php endif; ?>

</p>

<p>

<button type="submit">送信</button>

</p>

</form>

</div>

</body>

</html>

次に「/var/www/html/dirtra」に「view\_main.php」を作成し、「/var/www/html/mail」にある「reservation\_main.php」の内容をコピーする。「$\_SESSION[‘item\_name’]」として「view.php」から渡されてくる内容を「$item\_name」に格納するように変更する。

$item\_name = $\_SESSION[‘item\_name’];

「$data\_dir」という変数に、データが格納される「data」というディレクトリを相対パスで指定する。

$data\_dir = ‘./data/’;

ファイルの内容を読み込むには次のようにすれば良い

$content = file\_get\_contents($data\_dir . $item\_name);

ファイルが存在しなかった場合には「$content」に「FALSE」が返されるので、次のようにしてファイルがなかったことを示すようにする。

if ($content === false) {

$content = ‘None’;

}

以上のようにして「view\_main.php」が出力するHTMLの内容には、「view.php」で指定されたファイルの内容が表示されるようにする。以下に変更後の「view.php」を示す。

<?php

session\_start();

$item\_name = $\_SESSION['item\_name'];

$data\_dir = './data/';

$content = file\_get\_contents($data\_dir . $item\_name);

if ($content === false) {

$content = 'None';

}

?>

<!DOCTYPE HTML>

<html lang="ja">

<head>

<meta charset="UTF-8">

<title></title>

</head>

<body>

<p><?php echo $content; ?></p>

</body>

</html>

データも作成する。「/var/www/html/dirtra/data」ディレクトリを作成し、その中に「dog」、「cat」、「mouse」という3つのテキストファイルを作成する。「dog」というファイルには「Dog」と書いておく。「cat」というファイルには「Cat」と書いておく。「mouse」というファイルには「Mouse」と書いておく。これらのファイルがApach(www-data)からの読み込みアクセスができるようパーミッションをwww-dataに対して「r--」にしておく。

プログラムが完成したら、「view.php」で「dog」、「cat」、「mouse」、「tiger」を入力し、それぞれ何が表示されるか確認する。「dog」と入力すると「dog」ファイルの中身の「Dog」、「cat」と入力すると「cat」ファイルの中身の「Cat」、「mouse」と入力すると「Mouse」、が表示される。「tiger」と入力すると、ファイルが存在しないため「None」と表示される。

最後に、ホームディレクトリに設置した「log.txt」にApachがアクセスできることを確認する。www-dataに対して「r--」以上のパーミッションが設定されていることを確認する。その状態で、「view.php」にディレクトリ・トラバーサルを行い、「log.txt」の内容を表示できるか確認する。確認するにはアイテム名に../../../../../home/ayaka/log.txtを入力する。相対パスでアクセスするパスを指定しているため、このように入力することで、ホームディレクトリ上のlog.txtにアクセスできる。

# PHPからHTMLメールを送信する。

reservation.phpをreservation4.phpとして複製する。reservation\_main.phpをreservation\_main4.phpとして複製する。

予約情報がWebサーバに送信されると、予約内容をユーザ名と同名のファイルから読み込み、その内容をHTMLメールとしてGmailに送信する。

reservation4.phpの送信先をreservation\_main4.phpに変更する。reservation\_main4.phpの内容を以下のようにする。

<?php

session\_start();

$user\_name = $\_SESSION['user\_name'];

$data\_dir = './data/';

$content = file\_get\_contents($data\_dir . $user\_name . '.txt');

if ($content === false) {

$phone = $\_SESSION['phone'];

$address = $\_SESSION['address'];

file\_put\_contents($data\_dir . $user\_name . '.txt', $phone . "\n" . $address . "\n");

}

else {

$phone = $\_SESSION['phone'];

$address = $\_SESSION['address'];

file\_put\_contents($data\_dir . $user\_name . '.txt', $phone . "\n" . $address . "\n");

$fp = fopen($data\_dir . $user\_name . '.txt', 'r');

$phone = fgets($fp);

$address = fgets($fp);

fclose($fp);

}

$to = 'c0a221130b@edu.teu.ac.jp';

$subject = 'reservation';

$message = "<html lang='ja'><body><h1>complete reservation</h1><p>name: " . $user\_name . "</p><p>phone: " . $phone . "</p><p>address: " . $address . "</p><p>" . $content . "</p>";

$headers = array(

'From' => 'ayaka@vm-ayaka',

'Reply-To' => 'ayaka@vm-ayaka',

'Content-type' => 'text/html; charset=UTF-8'

);

mail($to, $subject, $message, $headers);

?>

<!DOCTYPE HTML>

<html lang="ja">

<head>

<meta charset="UTF-8">

<title>\u4e88\u7d04\u5b8c\u4e86</title>

</head>

<body>

<h>\u4e88\u7d04\u5b8c\u4e86</h>

<p><?php echo $user\_name; ?></p>

<p><?php echo $phone; ?></p>

<p><?php echo $address;?></p>

<p><?php echo $content;?></p>

</body>

</html>

上記の通り、mailディレクトリ直下にデータ保存用のディレクトリ「data」を作成し、そのディレクトリ内にファイルを保存する。ユーザ名が存在していなければ新たに「ユーザ名.txt」のファイルを作成し、入力した内容を書き込んでからそのファイルを読み込む。存在していれば新たに入力した内容に書き換え、読み込む。

HTMLメールを送るにはheaderで'Content-type' => 'text/html; charset=UTF-8'を指定することでHTMLメールを送れるようになる。messageをHTMLで書いてあげることでHTMLメールになる。

ホームディレクトリにある「log.txt」の内容もユーザ名を../../../../../home/ayaka/logと指定することで、相対パスを辿り、log.txtを表示させることができる。