3.1 HTTPとセッション管理

クッキーによるセッション管理

アプリケーションデータを保持する目的でクッキーに直接、値を入れることはあまり行われません。その理由は

①クッキーに保持できる値の個数や文字長に制限がある。【 クッキー制約 ： クッキー1個当たり4kバイト、1Webサーバで20個、1ブラウザで300個 】
②クッキーの値は利用者本人には参照、変更できるので、秘密情報の格納には向かない。

クッキーには「キー番号」としてのセッションIDを格納し、実際の値はサーバ側で管理するのが一般的です。
※ 昔は、Webブラウジングをすると利用者が知らないうちに、自動的に蓄積されていくクッキーの有害論があったが、現在は開発メーカとユーザ側間での歩み寄りの結果、
セッション管理リスクの常識としては、「クッキーを使用する」のが最も安全になっている。

セッションIDの脆弱性 （セッションハイジャックなどを引き起こす）

①セッションIDは推測できてはいけない（連番や社員番号などは避けて、疑似乱数生成器を使用する。セッション管理機能は自作しない。メジャーな開発ツールの提供するセッション管理機能を使用する。）
②セッションIDは盗み出されないように厳重に管理する （XSS、HTTPヘッダ・インジェクション、Refererの悪用などから防御 → Webアプリに必要な脆弱性対策を作り込む必要がある ）
③セッションIDは強制されてはいけない （セッションフィクセーションから防御）

Refererの脆弱性

・セッション管理をするためには、Cookie、hiddenパラメータの他に、URL埋め込みのセッションIDが使えるが、情報漏洩の危険性が高い。
・ホームページにユーザのために用意されている機能に、外部サイトへのリンクがあったり、リンクを利用者が作成できる場合には、情報漏洩の危険性が高まる。

※ 利用者がセッションID付きのURLを自らSNSなどに投稿したり、何らかのきっかけでセッションID付きのURLが検索サイトに登録され、情報漏洩事故となった事例があります

セッションフィクセーション （セッションIDの固定化攻撃）

※ セッションアダプション → 未知のセッションIDを受け付ける （PHPやASP.NETにある脆弱性）
PHP5．5．4以降では、「session.use_strict_mode = 1」を指定すると解消される （php.ini）

・認証前にセッション変数を使用している場合に、セッションフィクセーションを受ける場合がある
・クッキーのみにセッションIDを保存する場合でも、セッションフィクセーションを受ける場合がある

♠♠♠ クッキーモンスターバグ、XSS脆弱性、HTTPヘッダインジェクション脆弱性によって、中間者攻撃によるクッキー改竄攻撃は防ぐ方法がない ♠♠♠
（HTTPでクッキーを設定すれば、そのクッキーはHTTPSでも有効になります。サイトがHTTPSであっても、クッキー改変の可能性があるということです。）
【対策】
セッションIDの固定化攻撃されることは許容し、攻撃を受けても、セッションハイジャックは防ぐように対策を行う
①認証後にセッションIDを変更する(認証前にはセッション変数に秘密情報を保存しない)
②ログイン時にトークンを生成し、クッキーとセッション変数の両方に記録し、認証確認時に比較
（トークンが外部に出力されるのはログイン時のクッキー生成時のみなので、トークンを攻撃者が知る手段はありません）
③セッション管理機構を自作しない

クッキーの属性対応

Domain	ブラウザがクッキー値を送信するサーバのドメインを設定する(複数のサーバを介して、情報発信している場合など)	→ 指定しない状態がもっとも安全
Path	デフォルトは「path=/」。デレクトリを指定しても安全性に影響はない。(オリジンポリシーはホスト単位で、デレクトリ単位ではない	→ 指定しない状態がもっとも安全
Expires	設定しないとブラウザ終了でクッキー削除される。設定するとブラウザ終了後も認証状態が継続することになる。	→ 指定しない状態がもっとも安全
Secure	HTTPSの場合のみクッキーを送信	→ 指定する方が普通は良い
HttpOnly	JavaScriptからのアクセスを禁止	→ 指定する方が普通は良い

PHPのセッションID生成の安全性強化 （PHP-5.4.0からのデフォルト値）

php.ini

```
[Session]
;; entropy_file は Windowsでは設定不要
session.entropy_file = /dev/urandom
session.entropy_length = 32
```

PHPのセッションIDがURL埋め込みになる条件

php.ini のセッションID設定項目

項目	説明	デフォルト
session.use_cookies	セッションの保存にCookieを使用する	有効(On)
session.use_only_cookies	セッションの保存にCookieのみを使用す	有効(On)
session.use_trans_sid	URLにセッションIDを自動埋め込みする	無効(Off)

session.use_cookies と session.use_only_cookies

セッションIDの保存場所	session.use_cookies	session.use_only_cookies
セッションの保存にCookieのみを使用する	On	On
cookieが使えるときはcookieに、使えないときはURL埋め込み	On	Off
無意味な組み合わせ	Off	On
セッションIDを常にURL埋め込みにする	Off	Off

「.user.ini」(デレクトリ毎のカスタマイズ設定ファイル)

セッションIDをURL埋め込みにする

```
session.use_cookies=Off
session.use_only_cookies=Off
session.use_trans_sid=On
```

3.1 HTTPとセッション管理

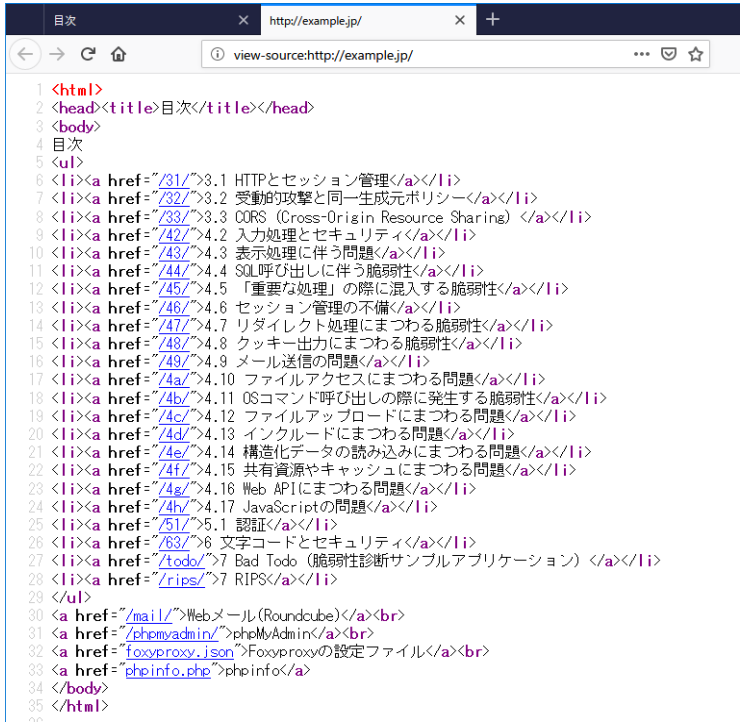
【サーバ：index.html】

/var/www/html/index.html - wasbook@192.168.56.101 - エディタ - WinSCP

```
<html>
<head><title>目次</title></head>
<body>
目次
<ul>
<li><a href="/31/">3.1 HTTPとセッション管理</a></li>
<li><a href="/32/">3.2 受動的攻撃と同一生成元ポリシー</a></li>
<li><a href="/33/">3.3 CORS（Cross-Origin Resource Sharing）</a></li>
<li><a href="/42/">4.2 入力処理とセキュリティ</a></li>
<li><a href="/43/">4.3 表示処理に伴う問題</a></li>
<li><a href="/44/">4.4 SQL呼び出しに伴う脆弱性</a></li>
<li><a href="/45/">4.5 「重要な処理」の際に混入する脆弱性</a></li>
<li><a href="/46/">4.6 セッション管理の不備</a></li>
<li><a href="/47/">4.7 リダイレクト処理にまつわる脆弱性</a></li>
<li><a href="/48/">4.8 クッキー出力にまつわる脆弱性</a></li>
<li><a href="/49/">4.9 メール送信の問題</a></li>
<li><a href="/4a/">4.10 ファイルアクセスにまつわる問題</a></li>
<li><a href="/4b/">4.11 OSコマンド呼び出しの際に発生する脆弱性</a></li>
<li><a href="/4c/">4.12 ファイルアップロードにまつわる問題</a></li>
<li><a href="/4d/">4.13 インクルードにまつわる問題</a></li>
<li><a href="/4e/">4.14 構造化データの読み込みにまつわる問題</a></li>
<li><a href="/4f/">4.15 共有資源やキャッシュにまつわる問題</a></li>
<li><a href="/4g/">4.16 Web APIにまつわる問題</a></li>
<li><a href="/4h/">4.17 JavaScriptの問題</a></li>
<li><a href="/51/">5.1 認証</a></li>
<li><a href="/63/">6 文字コードとセキュリティ</a></li>
<li><a href="/todo/">7 Bad Todo（脆弱性診断サンプルアプリケーション）</a></li>
<li><a href="/rips/">7 RIPS</a></li>
</ul>
<a href="/mail/">Webメール(Roundcube)</a><br>
<a href="/phpmyadmin/">phpMyAdmin</a><br>
<a href="foxyproxy.json">Foxyproxyの設定ファイル</a><br>
<a href="phpinfo.php">phpinfo</a>
</body>
</html>
```

【ブラウザ】


目次
example.jp
目次
3.1 HTTPとセッション管理
3.2 受動的攻撃と同一生成元ポリシー
3.3 CORS（Cross-Origin Resource Sharing）
4.2 入力処理とセキュリティ
4.3 表示処理に伴う問題
4.4 SQL呼び出しに伴う脆弱性
4.5 「重要な処理」の際に混入する脆弱性
4.6 セッション管理の不備
4.7 リダイレクト処理にまつわる脆弱性
4.8 クッキー出力にまつわる脆弱性
4.9 メール送信の問題
4.10 ファイルアクセスにまつわる問題
4.11 OSコマンド呼び出しの際に発生する脆弱性
4.12 ファイルアップロードにまつわる問題
4.13 インクルードにまつわる問題
4.14 構造化データの読み込みにまつわる問題
4.15 共有資源やキャッシュにまつわる問題
4.16 Web APIにまつわる問題
4.17 JavaScriptの問題
5.1 認証
6 文字コードとセキュリティ
7 Bad Todo（脆弱性診断サンプルアプリケーション）
7 RIPS
Webメール(Roundcube)
phpMyAdmin
Foxyproxyの設定ファイル
phpinfo


目次
http://example.jp/
view-source:http://example.jp/
<html>
<head><title>目次</title></head>
<body>
目次

3.1 HTTPとセッション管理
3.2 受動的攻撃と同一生成元ポリシー
3.3 CORS (Cross-Origin Resource Sharing)
4.2 入力処理とセキュリティ
4.3 表示処理に伴う問題
4.4 SQL呼び出しに伴う脆弱性
4.5 「重要な処理」の際に混入する脆弱性
4.6 セッション管理の不備
4.7 リダイレクト処理にまつわる脆弱性
4.8 クッキー出力にまつわる脆弱性
4.9 メール送信の問題
4.10 ファイルアクセスにまつわる問題
4.11 OSコマンド呼び出しの際に発生する脆弱性
4.12 ファイルアップロードにまつわる問題
4.13 インクルードにまつわる問題
4.14 構造化データの読み込みにまつわる問題
4.15 共有資源やキャッシュにまつわる問題
4.16 Web APIにまつわる問題
4.17 JavaScriptの問題
5.1 認証
6 文字コードとセキュリティ
7 Bad Todo (脆弱性診断サンプルアプリケーション)
7 RIPS

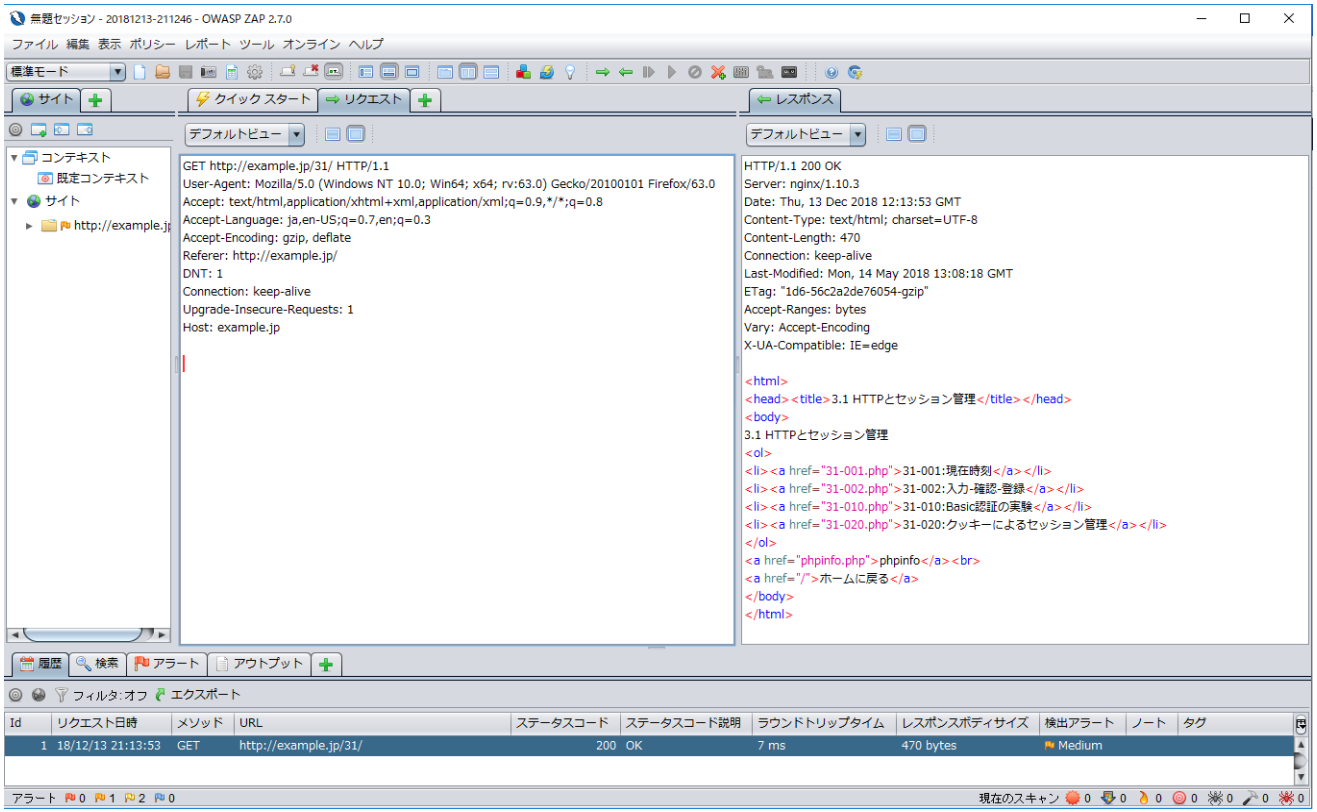
Webメール(Roundcube)
phpMyAdmin
Foxyproxyの設定ファイル
phpinfo
</body>
</html>

【サーバ： 31/index.html】

/var/www/html/31/index.html - wasbook@192.168.56.101 - エディタ - WinSCP

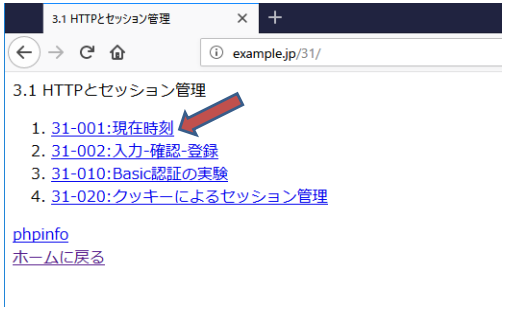
```
<html>
<head><title>3.1 HTTPとセッション管理</title></head>
<body>
3.1 HTTPとセッション管理
<ol>
<li><a href="31-001.php">31-001:現在時刻</a></li>
<li><a href="31-002.php">31-002:入力-確認-登録</a></li>
<li><a href="31-010.php">31-010:Basic認証の実験</a></li>
<li><a href="31-020.php">31-020:クッキーによるセッション管理</a></li>
</ol>
<a href="phpinfo.php">phpinfo</a><br>
<a href="/">ホームに戻る</a>
</body>
</html>
```

【ブラウザ→サーバ： リクエスト 31/index.html → レスポンス 】

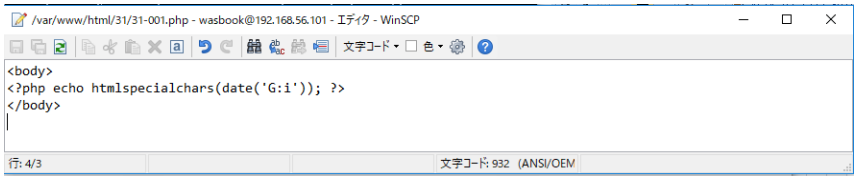


31-001:現在時刻

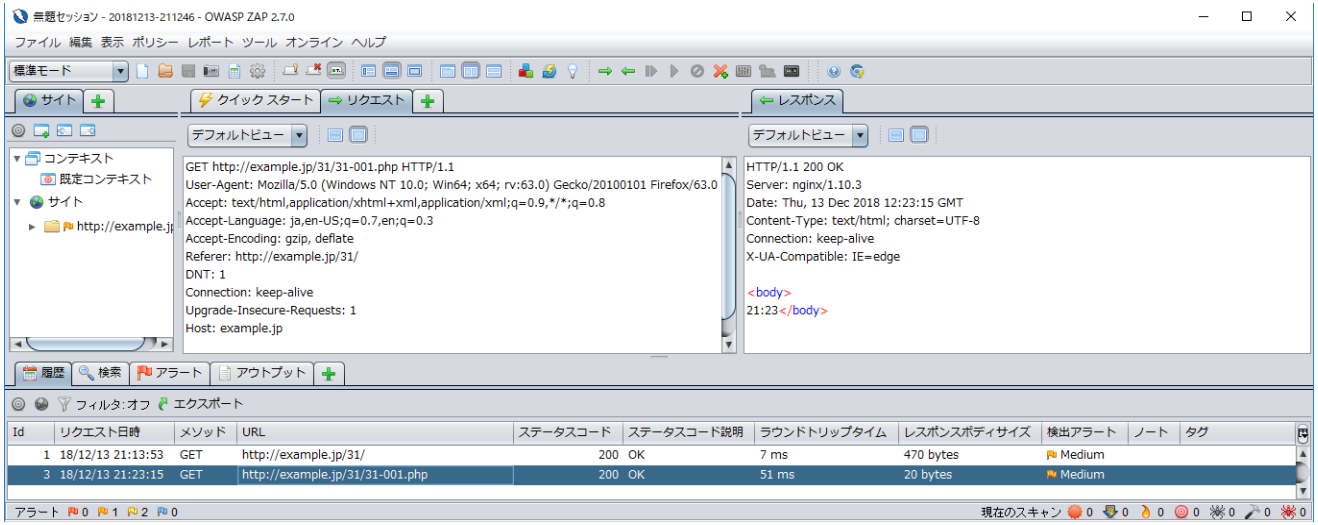
【ブラウザ】



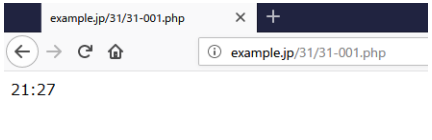
【サーバ： 31/31-001.php 】

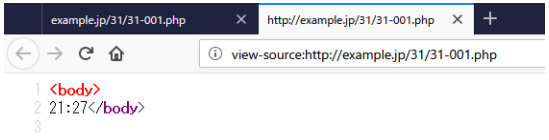


【ブラウザ→サーバ：リクエスト 31/31-001.php → レスポンス 】



【ブラウザ】



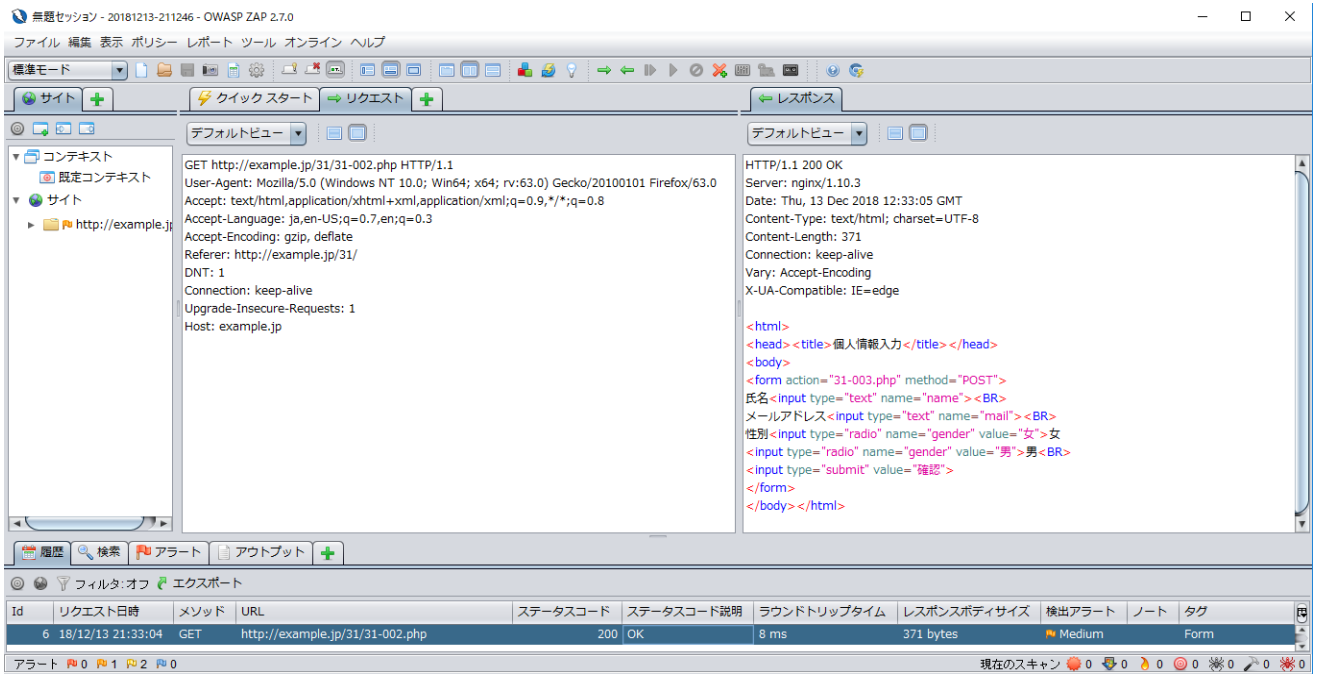


31-002:入力-確認-登録

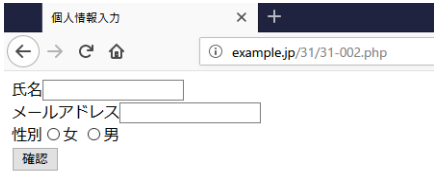
【ブラウザ】

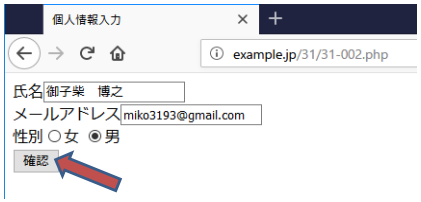
【サーバ： 31/31-002.php 】

【ブラウザ→サーバ： リクエスト 31/31-002.php → レスポンス 】



【ブラウザ】





<input>タグに入力した内容が、POSTメソッドで、サーバに送られます。

【サーバ： 31/31-003.php 】

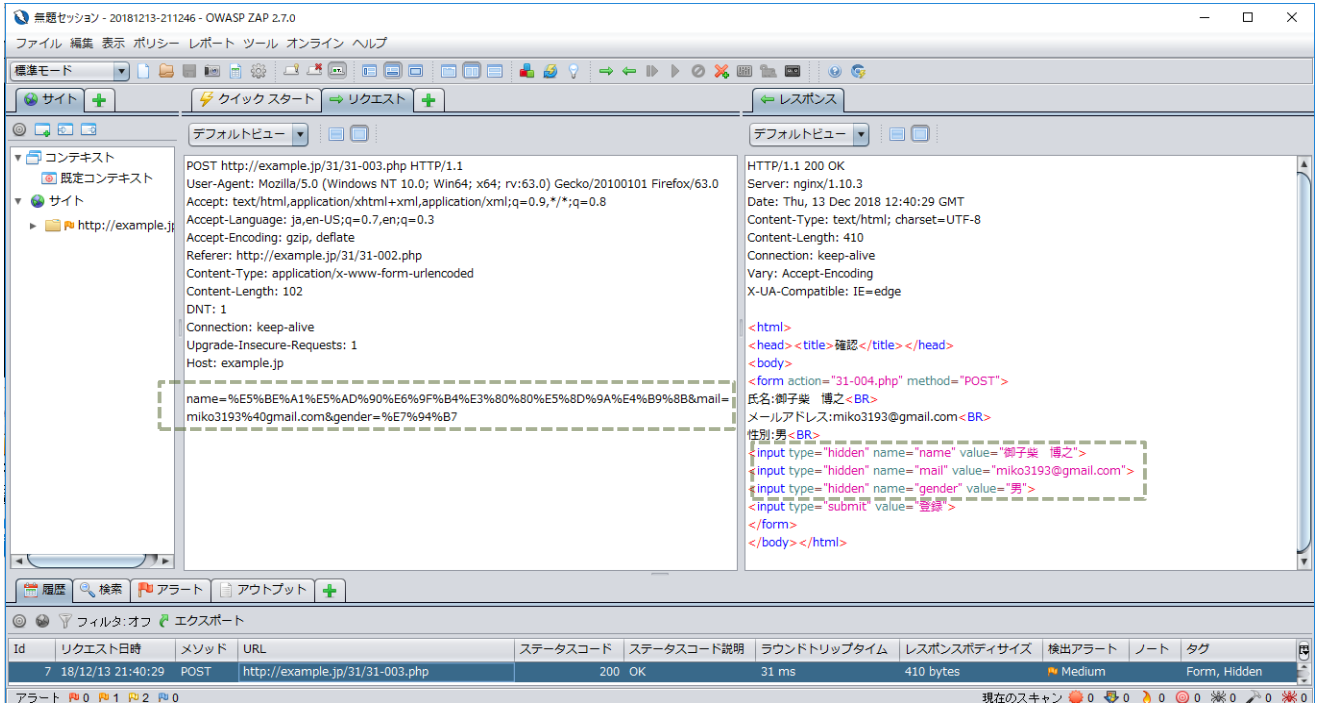
/var/www/html/31/31-003.php - wasbook@192.168.56.101 - エディタ - WinSCP

```
<?php
  $name = @$_POST['name'];
  $mail = @$_POST['mail'];
  $gender = @$_POST['gender'];
?>
<html>
<head><title>確認</title></head>
<body>
<form action="31-004.php" method="POST">
氏名:<?php echo htmlspecialchars($name, ENT_NOQUOTES, 'UTF-8'); ?><BR>
メールアドレス:<?php echo htmlspecialchars($mail, ENT_NOQUOTES, 'UTF-8'); ?><BR>
性別:<?php echo htmlspecialchars($gender, ENT_NOQUOTES, 'UTF-8'); ?><BR>
<input type="hidden" name="name" value="<?php echo htmlspecialchars($name, ENT_COMPAT, 'UTF-8'); ?>">
<input type="hidden" name="mail" value="<?php echo htmlspecialchars($mail, ENT_COMPAT, 'UTF-8'); ?>">
<input type="hidden" name="gender" value="<?php echo htmlspecialchars($gender, ENT_COMPAT, 'UTF-8'); ?>">
<input type="submit" value="登録">
</form>
</body></html>
```

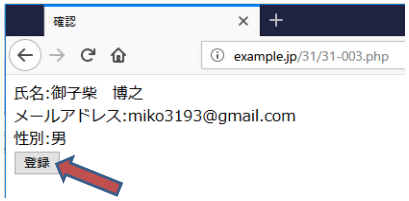
ブラウザで入力した<input>タブの内容を、$_POSTで受け取って、変数に保存しています。

変数の内容について、htmlspecialchars関数で、サーバに危険な影響を及ぼさないようにエスケープ処理をしています。

【ブラウザ→サーバ： リクエスト 31/31-003.php → レスポンス 】

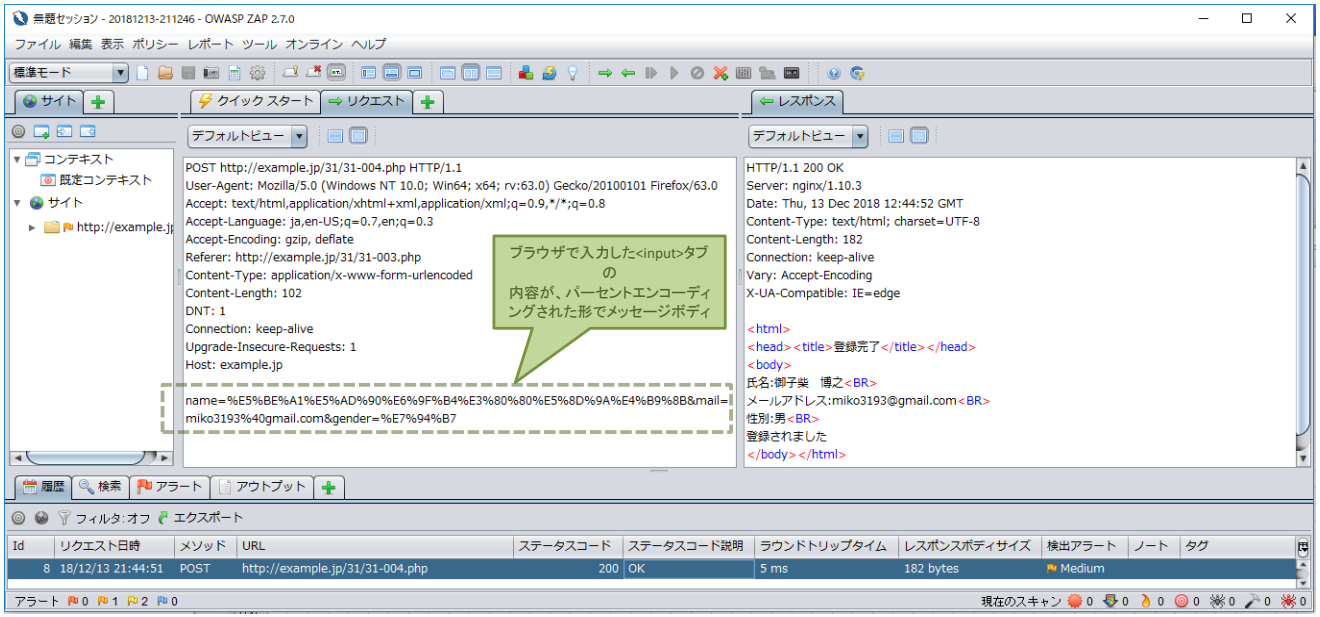


【ブラウザ】

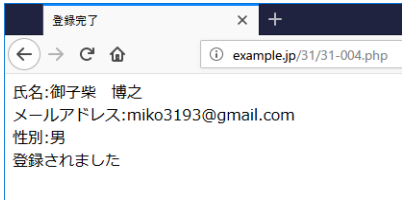


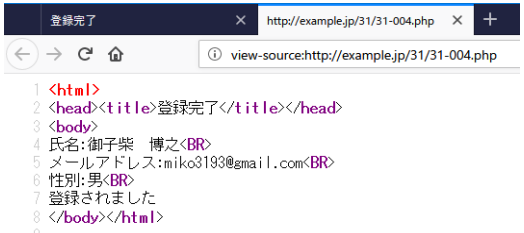
【サーバ：　31/31-004.php　】

【ブラウザ→サーバ： リクエスト 31/31-004.php → レスポンス 】



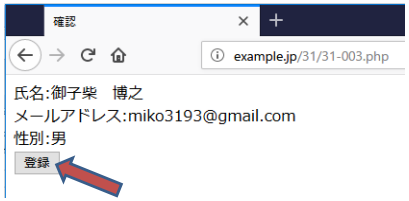
【ブラウザ】



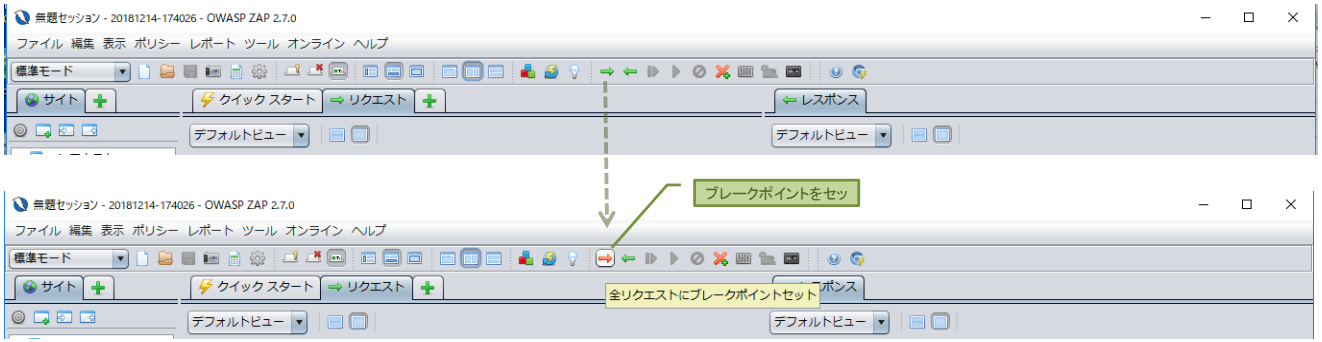


hiddenパラメータの書き換え

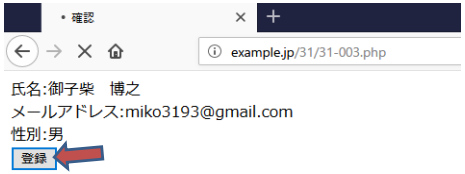
【ブラウザ】



ブレークポイントをセット

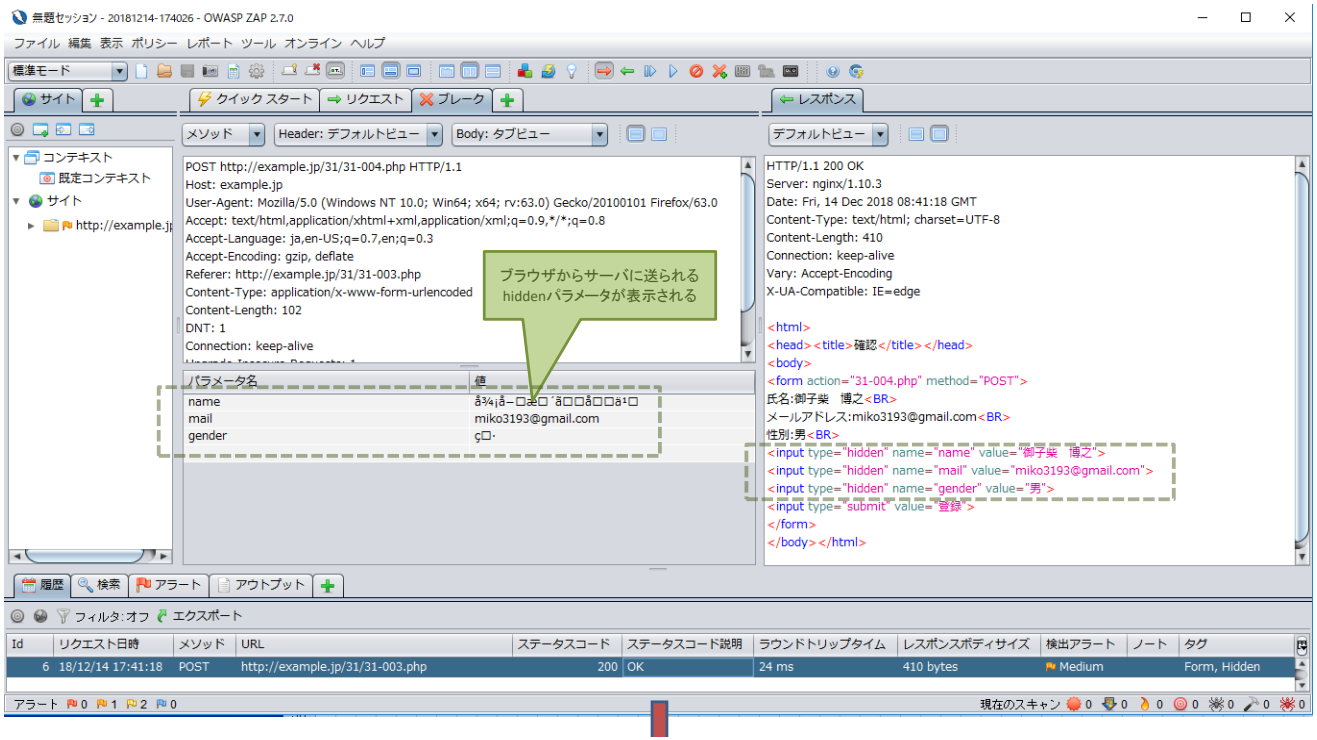


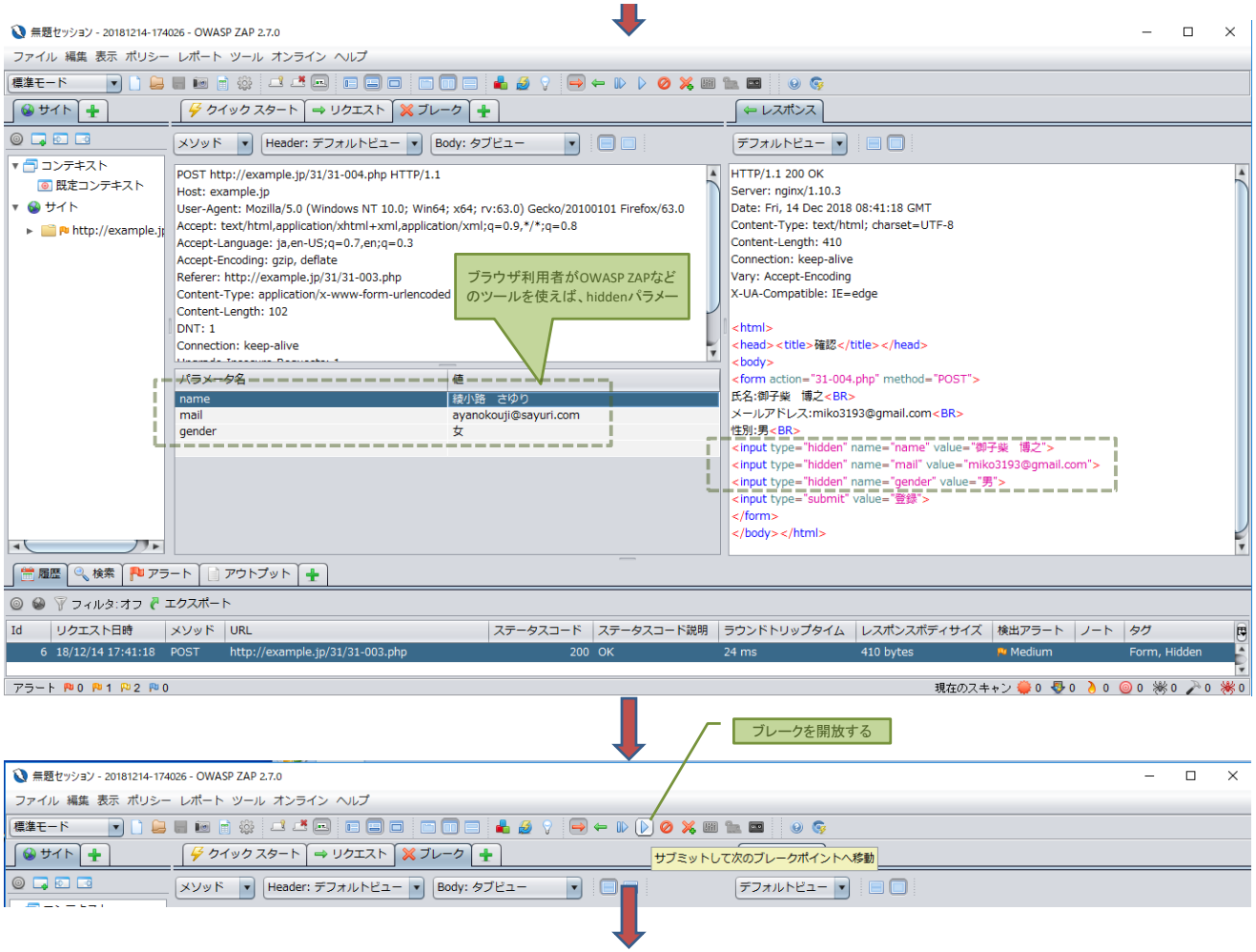
【ブラウザ】

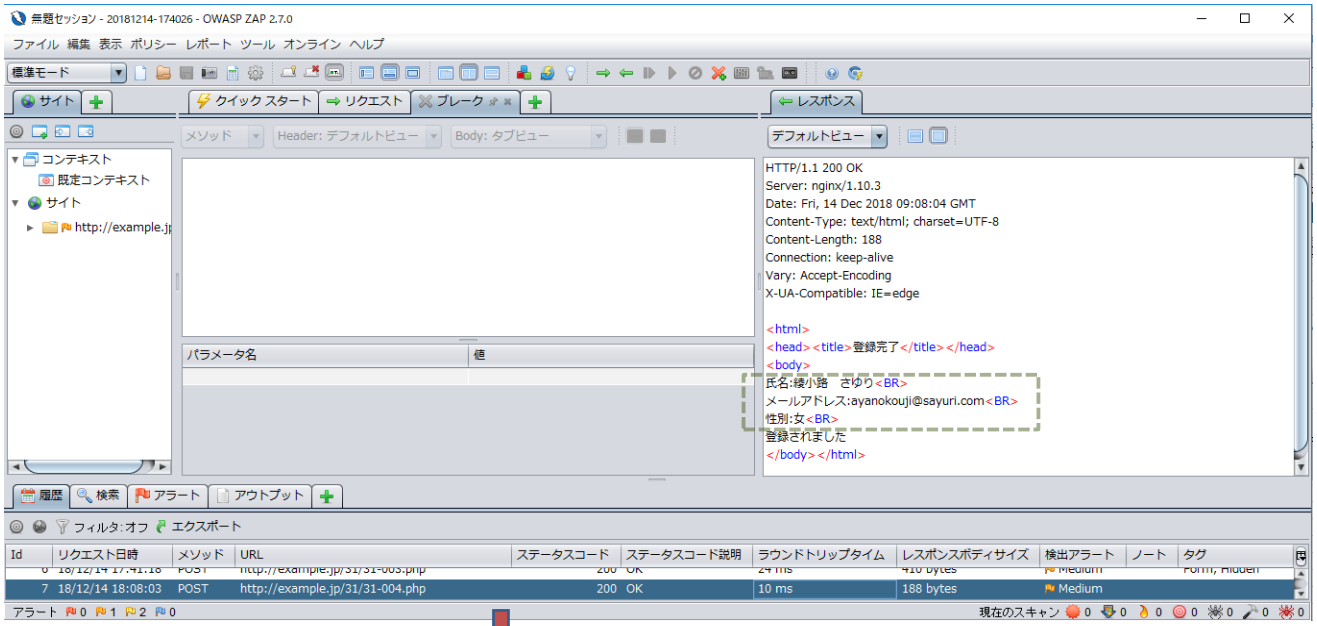


ブラウザから登録内容を送信

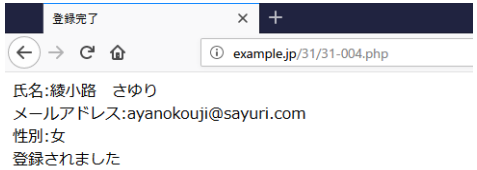
【ブラウザ→サーバ：リクエスト 31/31-004.php → レスポンス 】




無題セッション - 20181214-174026 - OWASP ZAP 2.7.0
ファイル 編集 表示 ポリシー レポート ツール オンライン ヘルプ
標準モード
サイト
クイック スタート
リクエスト
ブレーク
レスポンス
メソッド
Header: デフォルトビュー
Body: タブビュー
デフォルトビュー
コンテキスト
既定コンテキスト
サイト
POST http://example.jp/31/31-004.php HTTP/1.1
Host: example.jp
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://example.jp/31/31-003.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 102
DNT: 1
Connection: keep-alive
ブラウザ利用者がOWASP ZAPなどのツールを使えば、hiddenパラメー
パラメータ名
値
name
綾小路　さゆり
mail
ayanokouji@sayuri.com
gender
女
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Fri, 14 Dec 2018 08:41:18 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 410
Connection: keep-alive
Vary: Accept-Encoding
X-UA-Compatible: IE=edge
<html>
<head><title>確認</title></head>
<body>
<form action="31-004.php" method="POST">
氏名:御子柴　博之
メールアドレス:miko3193@gmail.com
性別:男
<input type="hidden" name="name" value="御子柴　博之">
<input type="hidden" name="mail" value="miko3193@gmail.com">
<input type="hidden" name="gender" value="男">
<input type="submit" value="登録">
</form>
</body></html>
履歴
検索
アラート
アウトプット
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
ステータスコード
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
6 18/12/14 17:41:18 POST http://example.jp/31/31-003.php 200 OK 24 ms 410 bytes Medium Form, Hidden
アラート 0 1 2 0
現在のスキャン 0 0 0 0 0 0 0
ブレークを開放する
無題セッション - 20181214-174026 - OWASP ZAP 2.7.0
ファイル 編集 表示 ポリシー レポート ツール オンライン ヘルプ
標準モード
サイト
クイック スタート
リクエスト
ブレーク
サブミットして次のブレークポイントへ移動
メソッド
Header: デフォルトビュー
Body: タブビュー
デフォルトビュー



【ブラウザ】



hiddenパラメータのメリット

hiddenパラメータは利用者自身からは書き換え可能であるが、情報漏洩や第3者からの書き換えに関しては堅牢です。

クッキーやセッション変数は、セッションIDの固定化攻撃に弱い欠点があります。クッキーモンスターバグの影響で、セッション変数の漏洩に対する具体的な対策がありません。

利用者に書き換えられては困る認証や認可に関する情報はセッション変数に保存すべきだが、それ以外の特にログイン前の情報はhiddenパラメータが安全です。

31-010:Basic認証の実験

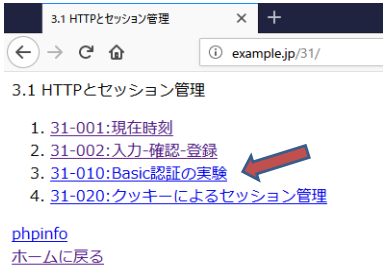
【サーバ： 31/31-010.php 】

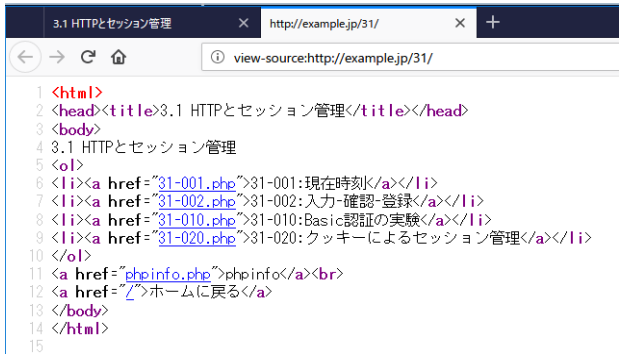
/var/www/html/31/31-010.php - wasbook@192.168.56.101 - エディタ - WinSCP

文字コード 色

```
<?php
  $user = @$_SERVER['PHP_AUTH_USER'];
  $pass = @$_SERVER['PHP_AUTH_PW'];
  if (! $user || ! $pass) {
    header('HTTP/1.1 401 Unauthorized');
    header('WWW-Authenticate: Basic realm="Basic Authentication Sample"');
    echo "ユーザ名とパスワードが必要です";
    exit;
  }
?>
<body>
認証しました<BR>
ユーザ名:<?php echo htmlspecialchars($user, ENT_NOQUOTES, 'UTF-8'); ?><BR>
パスワード:<?php echo htmlspecialchars($pass, ENT_NOQUOTES, 'UTF-8'); ?> <BR>
</body>
```

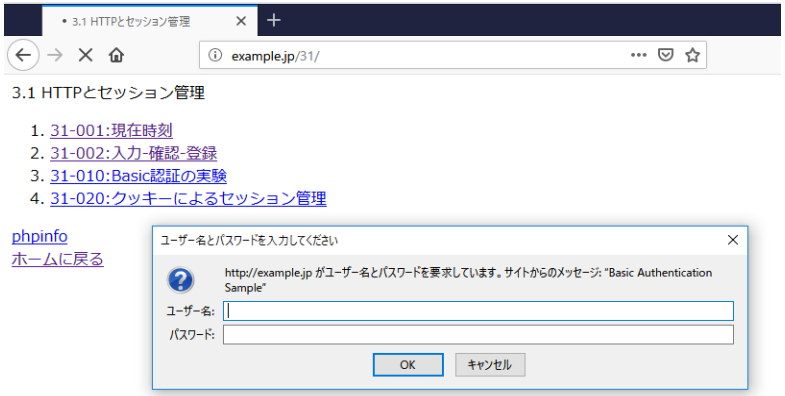
【ブラウザ】


3.1 HTTPとセッション管理
example.jp/31/
3.1 HTTPとセッション管理
1. 31-001:現在時刻
2. 31-002:入力-確認-登録
3. 31-010:Basic認証の実験
4. 31-020:クッキーによるセッション管理
phpinfo
ホームに戻る

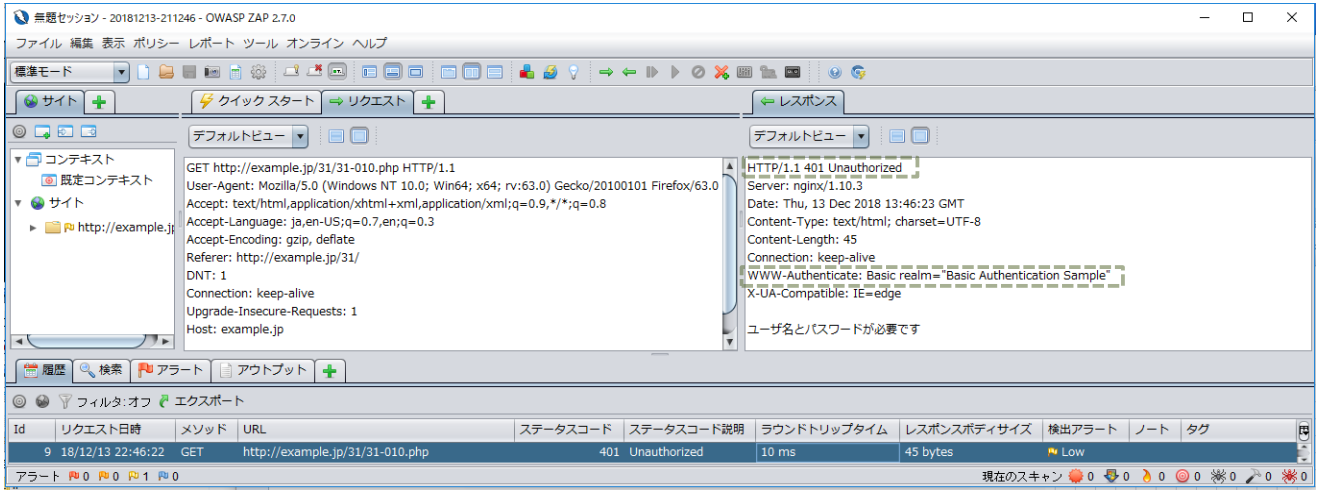

3.1 HTTPとセッション管理
http://example.jp/31/
view-source:http://example.jp/31/
<html>
<head><title>3.1 HTTPとセッション管理</title></head>
<body>
3.1 HTTPとセッション管理

31-001:現在時刻
31-002:入力-確認-登録
31-010:Basic認証の実験
31-020:クッキーによるセッション管理

phpinfo
ホームに戻る
</body>
</html>

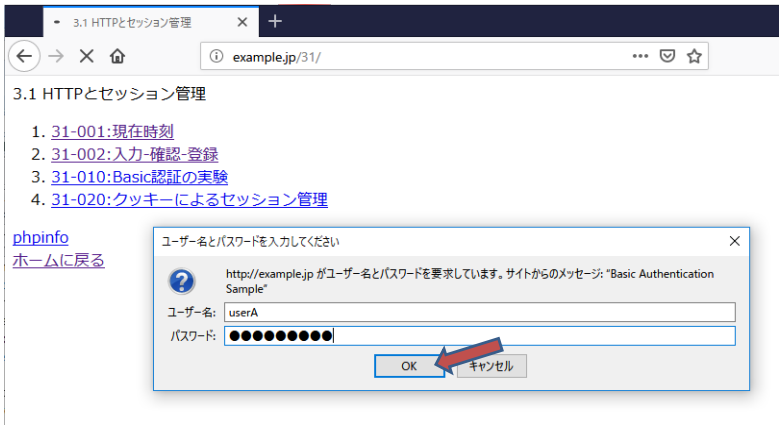

3.1 HTTPとセッション管理
example.jp/31/
3.1 HTTPとセッション管理
1. 31-001:現在時刻
2. 31-002:入力-確認-登録
3. 31-010:Basic認証の実験
4. 31-020:クッキーによるセッション管理
phpinfo
ホームに戻る
ユーザー名とパスワードを入力してください
http://example.jp がユーザー名とパスワードを要求しています。サイトからのメッセージ: "Basic Authentication Sample"
ユーザー名:
パスワード:
OK
キャンセル

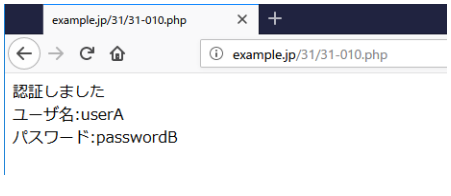
【ブラウザ→サーバ(初期)： リクエスト 31/31-010.php → レスポンス 】

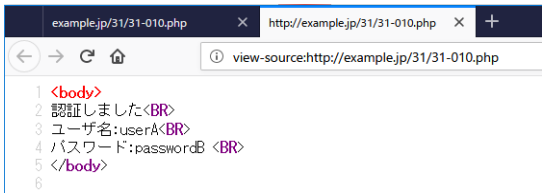


（認証OKの場合）

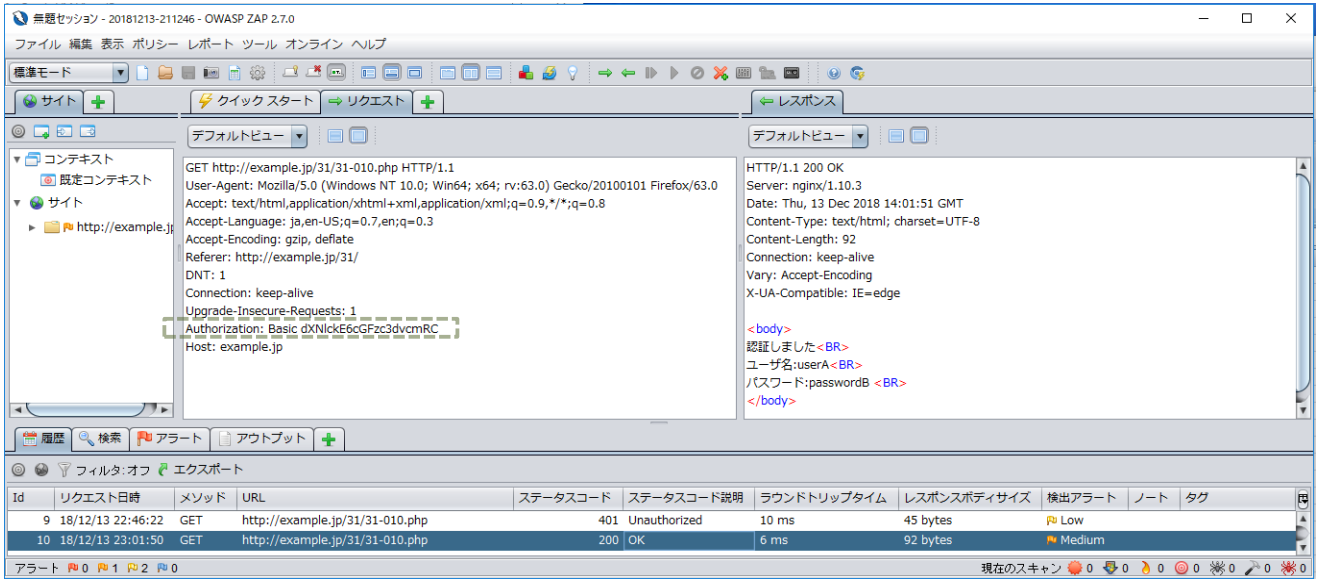
【ブラウザ】

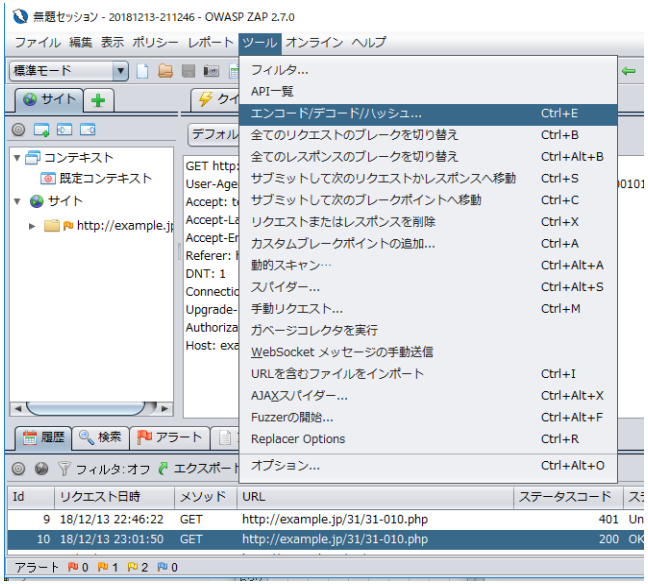


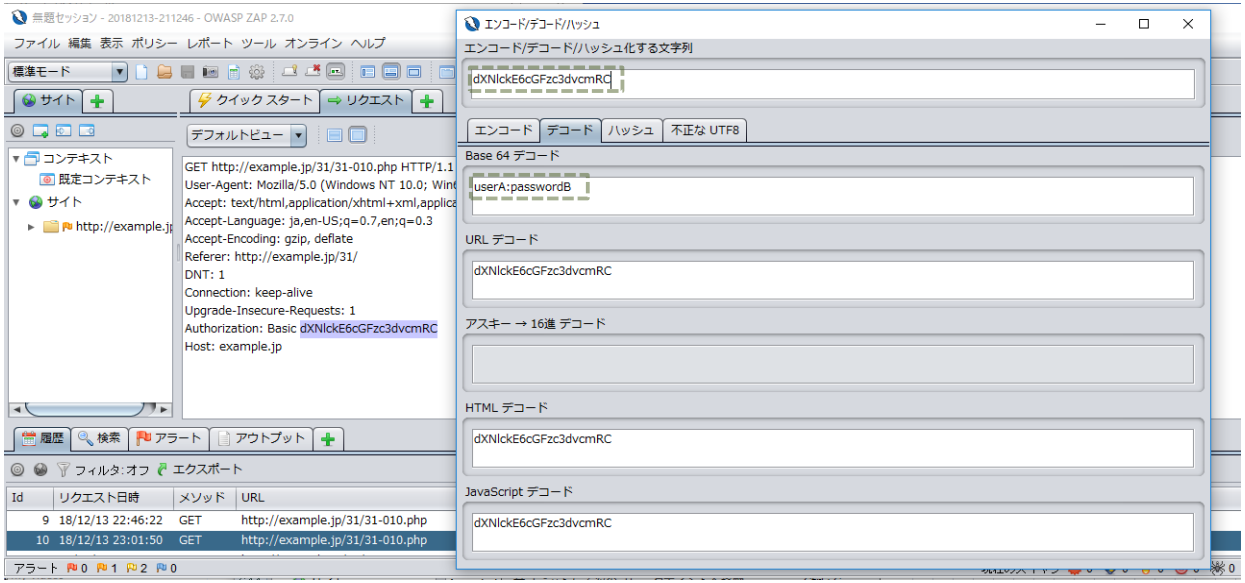




【ブラウザ→サーバ(OK)： リクエスト 31/31-010.php → レスポンス 】







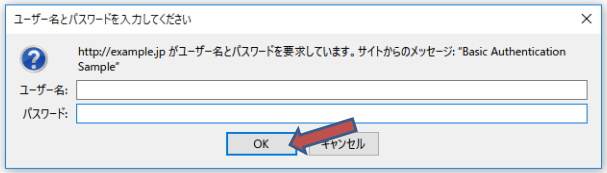
（認証NGの場合）

【ブラウザ】

3.1 HTTPとセッション管理

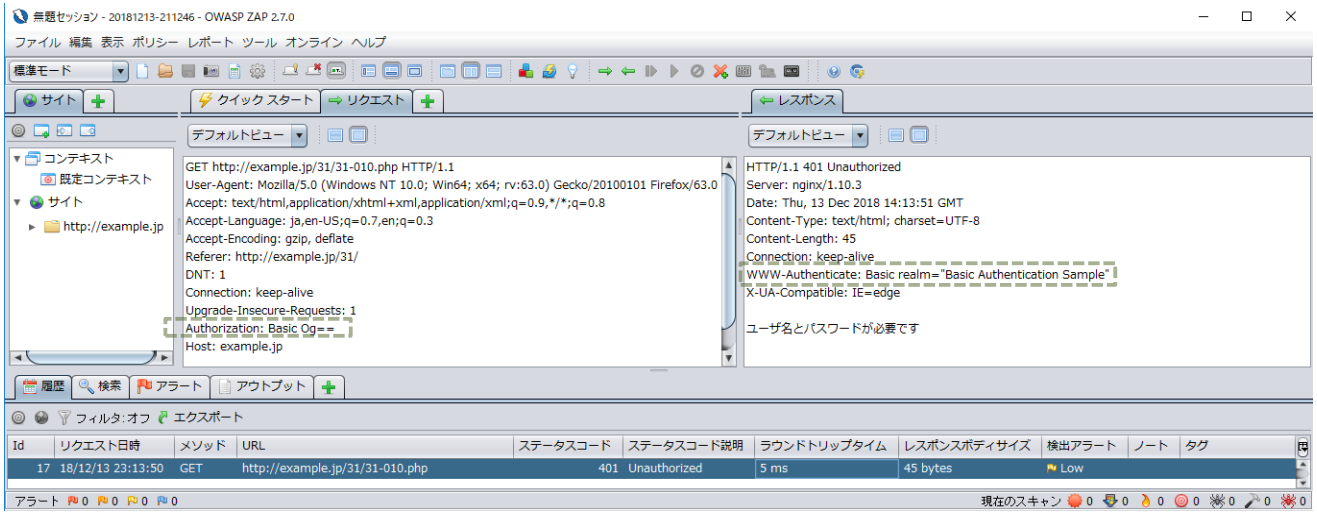
1. 31-001:現在時刻
2. 31-002:入力-確認-登録
3. 31-010:Basic認証の実験
4. 31-020:クッキーによるセッション管理

phpinfo
ホームに戻る



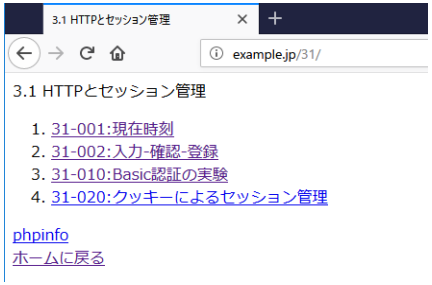
ユーザ、パスワードの認証に失敗すると

【ブラウザ→サーバ(NG)： リクエスト 31/31-010.php → レスポンス 】



31-020:クッキーによるセッション管理

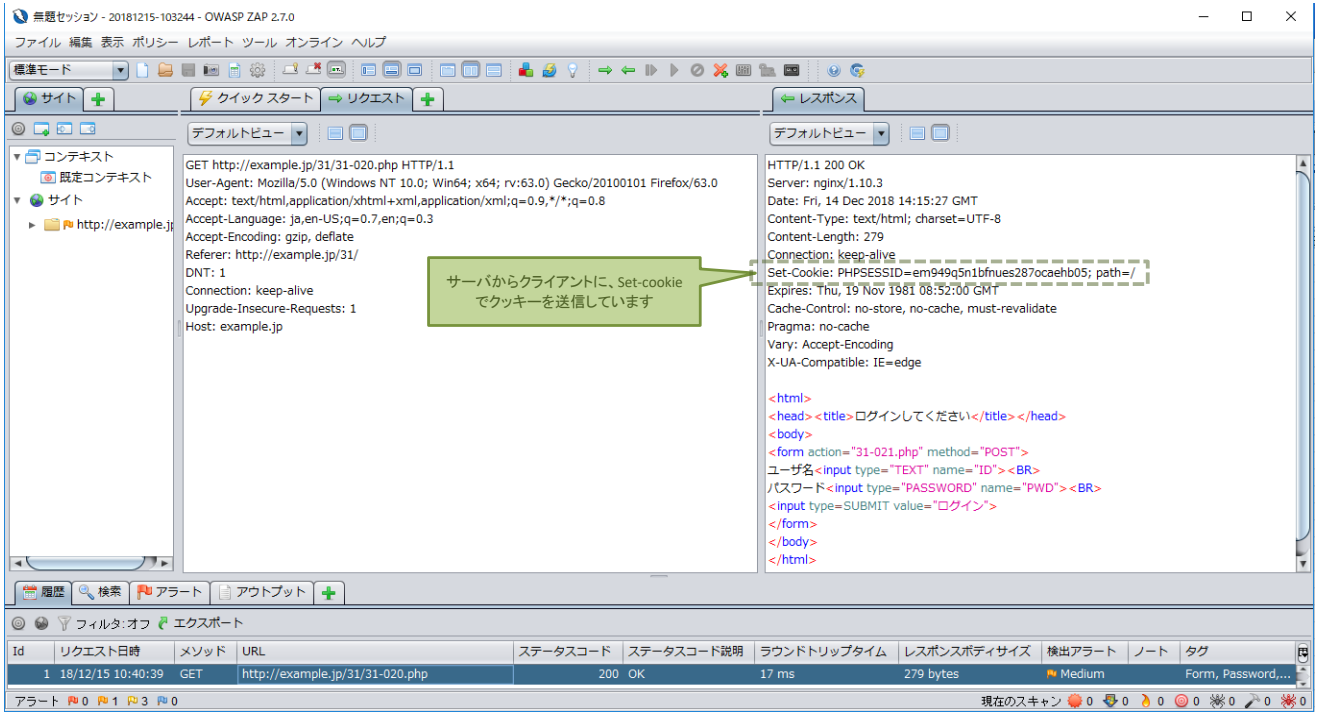
【ブラウザ】



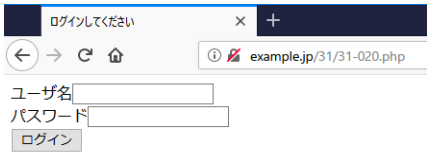
【サーバ： 31/31-020.php 】

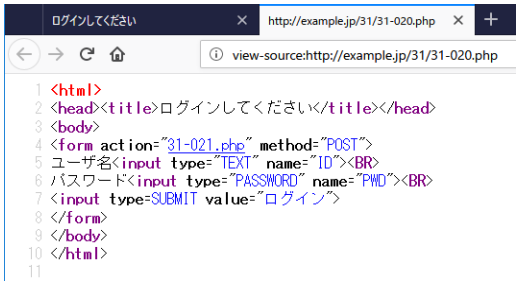
サーバとクライアント間で、セッション情報をやり取りします。
（具体的には、session_start(); を実行します）

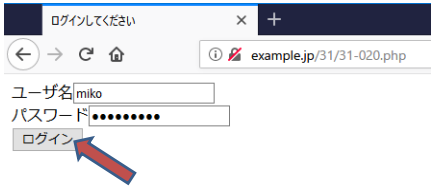
【ブラウザ→サーバ：リクエスト 31/31-020.php → レスポンス 】

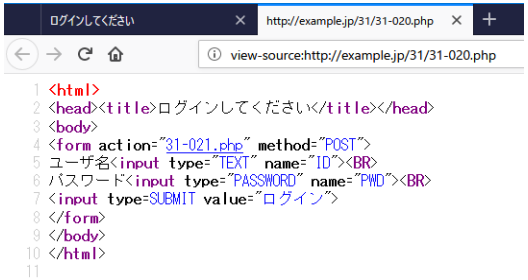


【ブラウザ】

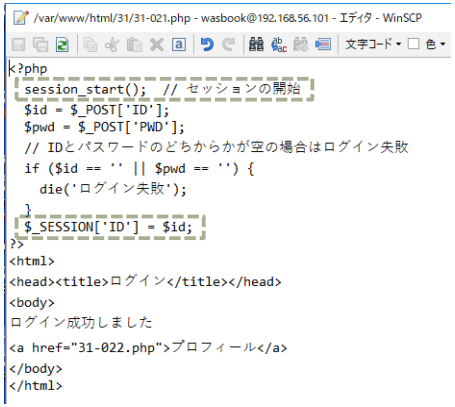








【サーバ： 31/31-021.php 】

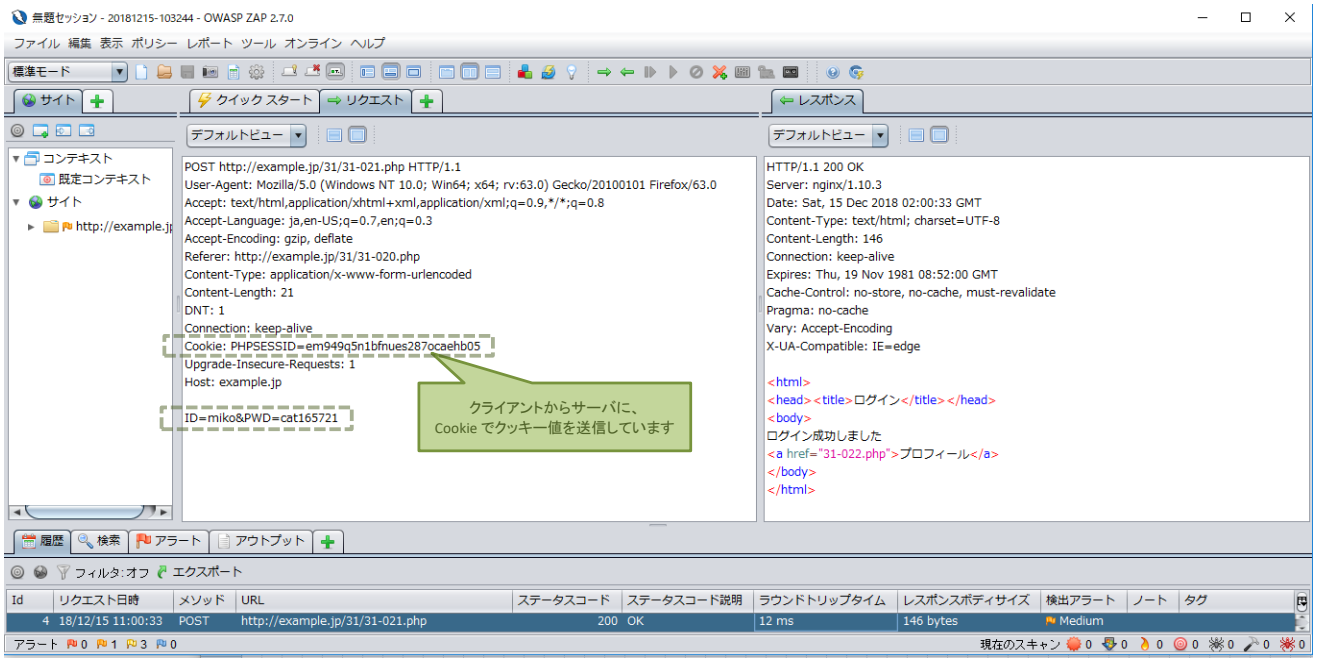


サーバとクライアント間で、セッション情報をやり取りします。
（具体的には、session_start(); を実行します）

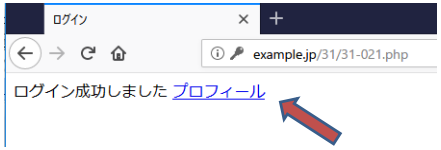
ID,Passwordチェックの結果、問題がない場合に、

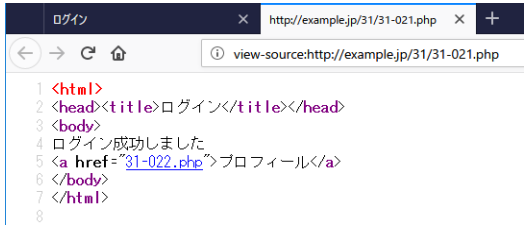
$_SESSION に ID をセットしている。

【ブラウザ→サーバ： リクエスト 31/31-021.php → レスポンス 】



【ブラウザ】





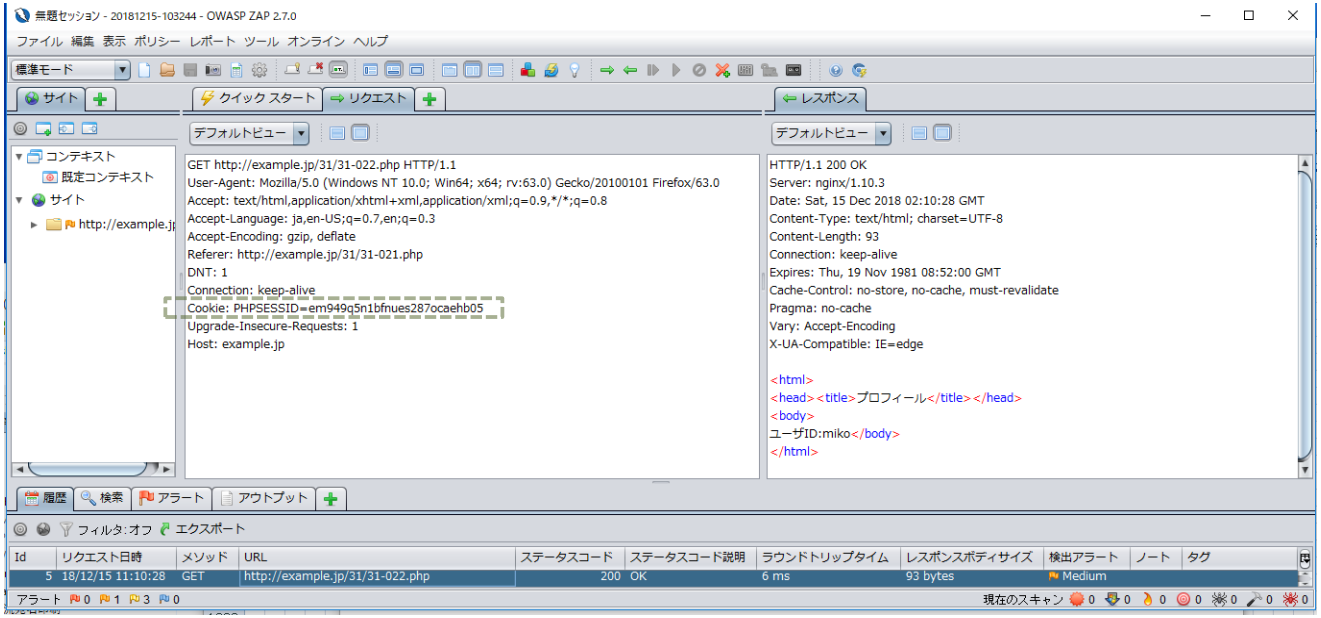
【サーバ： 31/31-022.php 】

/var/www/html/31/31-022.php - wasbook@192.168.56.101 - エディタ - WinSCP

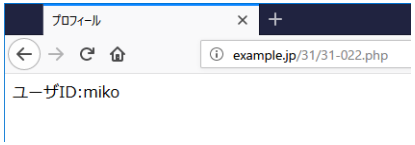
```
<?php
 session_start();  // セッションの開始
  $id = $_SESSION['ID'];
  if ($id == '') {
    die('ログインしてください');
  }
?>
<html>
<head><title>プロフィール</title></head>
<body>
ユーザID:<?php echo htmlspecialchars($id, ENT_NOQUOTES, 'UTF-8'); ?>
</body>
</html>
```

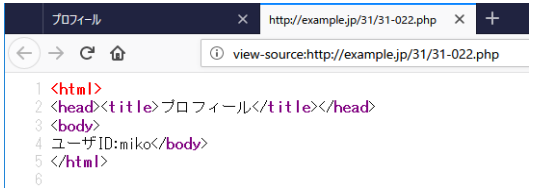
変数の内容について、htmlspecialchars関数で、サーバに危険な影響を及ぼさないように
エスケープ処理をしています。

【ブラウザ→サーバ： リクエスト 31/31-022.php → レスポンス 】



【ブラウザ】


プロフィール
example.jp/31/31-022.php
ユーザID:miko


プロフィール
http://example.jp/31/31-022.php
view-source:http://example.jp/31/31-022.php
1 <html>
2 <head><title>プロフィール</title></head>
3 <body>
4 ユーザID:miko</body>
5 </html>
6