

## 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言論があったが、現在は開発メーカとユーザ側間の歩み寄りの結果、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=/'。ディ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、ディレクトリ単位ではなし）	→ 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のみを使用する	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」(ディ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

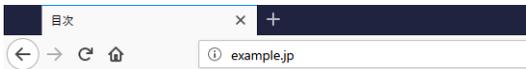
```
session.use_cookies=Off  
session.use_only_cookies=Off  
session.use_trans_sid=On
```

## 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
<khtml>
<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がリシー</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

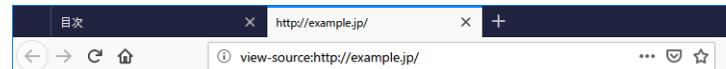
- [3.1 HTTPとセッション管理](#)
- [3.2 受動的攻撃と同一生成元ポリシー](#)
- [3.3 CORS \(Cross-Origin Resource Sharing\)](#)
- [4.2 入力処理とセキュリティ](#)
- [4.3 表示処理に伴う問題](#)
- [4.4 SQL呼び出しに伴う脆弱性](#)
- [4.5 「重要な処理」の際に混入する脆弱性](#)
- [4.6 セッション管理の不備](#)
- [4.7 リダイレクト処理にまつわる脆弱性](#)
- [4.8 クッキー出力にまつわる脆弱性](#)
- [4.9 メール送信の問題](#)
- [4.10 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](#)



```
1 <html>
2 <head><title>目次</title></head>
3 <body>
4 目次
5 <ul>
6 <li><a href="/31/">3.1 HTTPとセッション管理</a></li>
7 <li><a href="/32/">3.2 受動的攻撃と同一生成元ポリシー</a></li>
8 <li><a href="/33/">3.3 CORS (Cross-Origin Resource Sharing) </a></li>
9 <li><a href="/42/">4.2 入力処理とセキュリティ</a></li>
10 <li><a href="/43/">4.3 表示処理に伴う問題</a></li>
11 <li><a href="/44/">4.4 SQL呼び出しに伴う脆弱性</a></li>
12 <li><a href="/45/">4.5 「重要な処理」の際に混入する脆弱性</a></li>
13 <li><a href="/46/">4.6 セッション管理の不備</a></li>
14 <li><a href="/47/">4.7 リダイレクト処理にまつわる脆弱性</a></li>
15 <li><a href="/48/">4.8 クッキー出力にまつわる脆弱性</a></li>
16 <li><a href="/49/">4.9 メール送信の問題</a></li>
17 <li><a href="/4a/">4.10 ファイルアクセスにまつわる問題</a></li>
18 <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>
19 <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>
20 <li><a href="/4d/">4.13 インクルードにまつわる問題</a></li>
21 <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>
22 <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>
23 <li><a href="/4g/">4.16 Web APIにまつわる問題</a></li>
24 <li><a href="/4h/">4.17 JavaScriptの問題</a></li>
25 <li><a href="/51/">5.1 認証</a></li>
26 <li><a href="/63/">6 文字コードとセキュリティ</a></li>
27 <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>
28 <li><a href="/rips/">7 RIPS</a></li>
29 </ul>
30 <a href="/mail/">Webメール(Roundcube)</a><br>
31 <a href="/phpmyadmin/">phpMyAdmin</a><br>
32 <a href="/foxyproxy.json">Foxyproxyの設定ファイル</a><br>
33 <a href="/phpinfo.php">phpinfo</a>
34 </body>
35 </html>
36
```

## 【サーバ: 31/index.html】

```
 /var/www/html/31/index.html - wasbook@192.168.56.101 - エディタ - WinSCP
khtml
<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】

無題セッション - 20181213-211246 - 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 レスポンス

コンテキスト 既定コンテキスト サイト http://example.jp

リクエスト: GET http://example.jp/31/ HTTP/1.1  
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/  
DNT: 1  
Connection: keep-alive  
Upgrade-Insecure-Requests: 1  
Host: example.jp

レスポンス: HTTP/1.1 200 OK  
Server: nginx/1.10.3  
Date: Thu, 13 Dec 2018 12:13:53 GMT  
Content-Type: text/html; charset=UTF-8  
Content-Length: 470  
Connection: keep-alive  
Last-Modified: Mon, 14 May 2018 13:08:18 GMT  
ETag: "1d6-56c2a2de76054-gzip"  
Accept-Ranges: bytes  
Vary: Accept-Encoding  
X-UA-Compatible: IE=edge

```
<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>
```

履歴 検索 アラート アウトプット

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
1	18/12/13 21:13:53	GET	http://example.jp/31/	200	OK	7 ms	470 bytes	Medium		

アラート 0 1 2 0 現在のスキャン 0 0 0 0 0 0 0 0

## 31-001:現在時刻

### 【ブラウザ】



3.1 HTTPとセッション管理

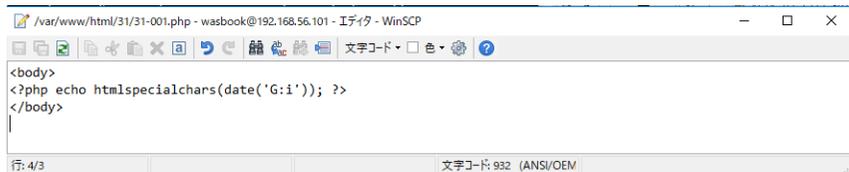
1. [31-001:現在時刻](#)
2. [31-002:入力-確認-登録](#)
3. [31-010:Basic認証の実験](#)
4. [31-020:クッキーによるセッション管理](#)

[phpinfo](#)  
[ホームに戻る](#)



```
1 <html>
2 <head><title>3.1 HTTPとセッション管理</title></head>
3 <body>
4 3.1 HTTPとセッション管理
5 <ol>
6 <li><a href="/31-001.php">31-001:現在時刻</a></li>
7 <li><a href="/31-002.php">31-002:入力-確認-登録</a></li>
8 <li><a href="/31-010.php">31-010:Basic認証の実験</a></li>
9 <li><a href="/31-020.php">31-020:クッキーによるセッション管理</a></li>
10 </ol>
11 <a href="/phpinfo.php">phpinfo</a><br>
12 <a href="/">ホームに戻る</a>
13 </body>
14 </html>
15
```

### 【サーバ: 31/31-001.php】



```
/var/www/html/31/31-001.php - wasbook@192.168.56.101 - エディタ - WinSCP
<body>
<?php echo htmlspecialchars(date('G:i')); ?>
</body>
|
行: 4/3 文字コード: 932 (ANSI/OEM)
```

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

The screenshot shows the Burp Suite interface with the following details:

- Request:**

```
GET http://example.jp/31/31-001.php HTTP/1.1
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/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp
```
- Response:**

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Thu, 13 Dec 2018 12:23:15 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-UA-Compatible: IE=edge

<body>
21:23</body>
```
- Log Table:**

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
1	18/12/13 21:13:53	GET	http://example.jp/31/	200	OK	7 ms	470 bytes	Medium		
3	18/12/13 21:23:15	GET	http://example.jp/31/31-001.php	200	OK	51 ms	20 bytes	Medium		

### 【ブラウザ】

The screenshot shows two browser tabs and the source view of the selected page:

- Tab 1: example.jp/31/31-001.php
- Tab 2: http://example.jp/31/31-001.php
- Source View Content:

```
1 <body>
2 21:27</body>
3
```

## 31-002:入力-確認-登録

### 【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](#)



```
1 <html>
2 <head><title>3.1 HTTPとセッション管理</title></head>
3 <body>
4 3.1 HTTPとセッション管理
5 <ol>
6 <li><a href="31-001.php">31-001:現在時刻</a></li>
7 <li><a href="31-002.php">31-002:入力-確認-登録</a></li>
8 <li><a href="31-010.php">31-010:Basic認証の実験</a></li>
9 <li><a href="31-020.php">31-020:クッキーによるセッション管理</a></li>
10 </ol>
11 <a href="phpinfo.php">phpinfo</a><br>
12 <a href="/">ホームに戻る</a>
13 </body>
14 </html>
15
```

### 【サーバ: 31/31-002.php】



```
/var/www/html/31/31-002.php - wasbook@192.168.56.101 - エディタ - WinSCP
khtml>
<head><title>個人情報入力</title></head>
<body>
<form action="31-003.php" method="POST">
氏名<input type="text" name="name"><BR>
メールアドレス<input type="text" name="mail"><BR>
性別<input type="radio" name="gender" value="女">女
<input type="radio" name="gender" value="男">男<BR>
<input type="submit" value="確認">
</form>
</body></html>
```

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

The screenshot shows the OWASP ZAP interface with the following details:

- Request:** GET http://example.jp/31/31-002.php HTTP/1.1  
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/  
DNT: 1  
Connection: keep-alive  
Upgrade-Insecure-Requests: 1  
Host: example.jp
- Response:** HTTP/1.1 200 OK  
Server: nginx/1.10.3  
Date: Thu, 13 Dec 2018 12:33:05 GMT  
Content-Type: text/html; charset=UTF-8  
Content-Length: 371  
Connection: keep-alive  
Vary: Accept-Encoding  
X-UA-Compatible: IE=edge

The response body contains the following HTML code:

```
<html>
<head> <title>個人情報入力</title></head>
<body>
<form action="31-003.php" method="POST">
氏名<input type="text" name="name"><BR>
メールアドレス<input type="text" name="mail"><BR>
性別<input type="radio" name="gender" value="女">女
<input type="radio" name="gender" value="男">男<BR>
<input type="submit" value="確認">
</form>
</body></html>
```

The bottom of the interface shows a table of request logs:

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	18/12/13 21:33:04	GET	http://example.jp/31/31-002.php	200	OK	8 ms	371 bytes	Medium		Form

At the bottom, there are status indicators for alerts (0), errors (1), warnings (2), and other metrics, along with the text "現在のスキャン" (Current scan).

## 【ブラウザ】

個人情報入力

example.jp/31/31-002.php

氏名

メールアドレス

性別  女  男

```
1 <html>
2 <head><title>個人情報入力</title></head>
3 <body>
4 <form action="31-003.php" method="POST">
5 氏名<input type="text" name="name"><BR>
6 メールアドレス<input type="text" name="mail"><BR>
7 性別<input type="radio" name="sender" value="女">女
8 <input type="radio" name="sender" value="男">男<BR>
9 <input type="submit" value="確認">
10 </form>
11 </body></html>
12
```

個人情報入力

example.jp/31/31-002.php

氏名

メールアドレス

性別  女  男

```
1 <html>
2 <head><title>個人情報入力</title></head>
3 <body>
4 <form action="31-003.php" method="POST">
5 氏名<input type="text" name="name"><BR>
6 メールアドレス<input type="text" name="mail"><BR>
7 性別<input type="radio" name="sender" value="女">女
8 <input type="radio" name="sender" value="男">男<BR>
9 <input type="submit" value="確認">
10 </form>
11 </body></html>
12
```

<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
k?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>




</form>
</body></html>
```

ブラウザで入力したタブの内容を、\$\_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】

The screenshot displays the Burp Suite interface with the following details:

- Request (left pane):**

```
POST http://example.jp/31/31-003.php HTTP/1.1
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/31/31-002.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 102
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp

name=%E5%BE%A1%E5%AD%90%E6%9F%B4%E3%80%80%E5%8D%9A%E4%B9%8B&mail=miko3193@gmail.com&gender=%E7%94%B7
```
- Response (right pane):**

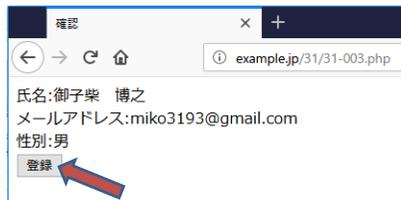
```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Thu, 13 Dec 2018 12:40:29 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 410
Connection: keep-alive
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<html>
<head><title>確認</title></head>
<body>
<form action="/31-004.php" method="POST">
  氏名:御子葉 博之<BR>
  メールアドレス:miko3193@gmail.com<BR>
  性別:男<BR>
  <input type="hidden" name="name" value="御子葉 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>
```

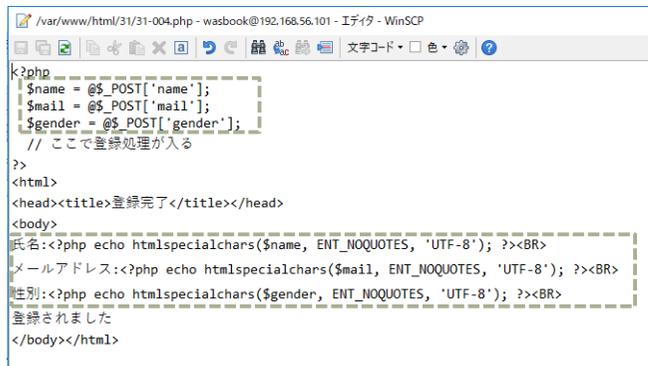
At the bottom, a table summarizes the request:

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
7	18/12/13 21:40:29	POST	http://example.jp/31/31-003.php	200	OK	31 ms	410 bytes	Medium		Form, Hidden

## 【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】

無題セッション - 20181213-211246 - 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

クイック スタート リクエスト レスポンス

コンテキスト: デフォルトビュー

POST http://example.jp/31/31-004.php HTTP/1.1  
 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  
 Upgrade-Insecure-Requests: 1  
 Host: example.jp

レスポンス: デフォルトビュー

HTTP/1.1 200 OK  
 Server: nginx/1.10.3  
 Date: Thu, 13 Dec 2018 12:44:52 GMT  
 Content-Type: text/html; charset=UTF-8  
 Content-Length: 182  
 Connection: keep-alive  
 Vary: Accept-Encoding  
 X-UA-Compatible: IE=edge

氏名:御子柴 博之<BR>  
 メールアドレス:miko3193@gmail.com<BR>  
 性別:男<BR>  
 登録されました  
 </body></html>

ブラウザで入力した<input>タブの内容が、パーセントエンコーディングされた形でメッセージボディ

name=%E5%BE%A1%E5%AD%A0%E6%9F%B4%E3%80%80%E5%8D%9A%E4%B9%8B&mail=miko3193%40gmail.com&gender=%E7%94%B7

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
8	18/12/13 21:44:51	POST	http://example.jp/31/31-004.php	200	OK	5 ms	182 bytes	Medium		

アラート 0 1 2 0

現在のスキャン 0 0 0 0 0 0 0 0 0 0

【ブラウザ】

登録完了

example.jp/31/31-004.php

氏名:御子柴 博之  
 メールアドレス:miko3193@gmail.com  
 性別:男  
 登録されました

登録完了

http://example.jp/31/31-004.php

view-source:http://example.jp/31/31-004.php

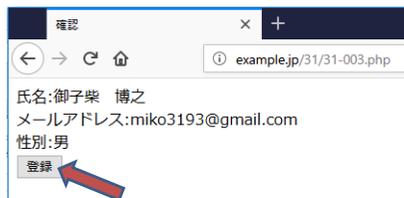
```

1 <html>
2 <head><title>登録完了</title></head>
3 <body>
4 氏名:御子柴 博之<BR>
5 メールアドレス:miko3193@gmail.com<BR>
6 性別:男<BR>
7 登録されました
8 </body></html>
9

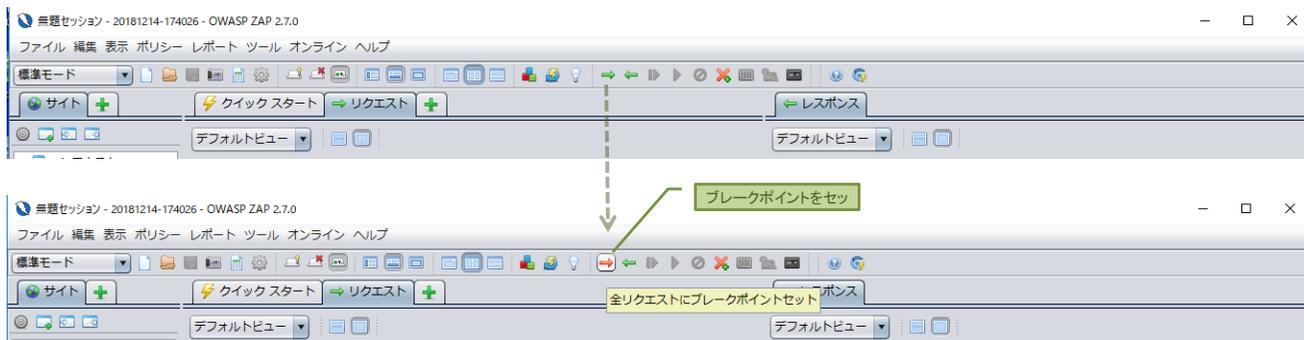
```

## hiddenパラメータの書き換え

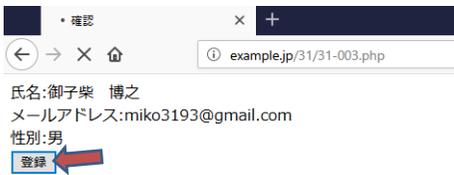
### 【ブラウザ】



### 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

標準モード

メソッド: POST URL: http://example.jp/31/31-004.php

Header: デフォルトビュー

Body: タブビュー

パラメータ名 | 値

name	御子柴 博之
mail	miko3193@gmail.com
gender	男

レスポンス

```
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<BR>
メールアドレス:miko3193@gmail.com<BR>
性別<BR>
<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>
```

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/31/31-004.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 0 0 0

無題セッション - 20181214-174026 - OWASP ZAP 2.7.0

標準モード

メソッド: POST URL: http://example.jp/31/31-004.php

Header: デフォルトビュー

Body: タブビュー

レスポンス: 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: http://example.jp

パラメータ名: name, mail, gender

値: 綾小路 さゆり, ayanokouji@sayuri.com, 女

ブラウザ利用者がOWASP ZAPなどのツールを使えば、hiddenパラメータ

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: 御子樂 博之 <BR>

メールアドレス: miko3193@gmail.com <BR>

性別: 男 <BR>

<input type="hidden" name="name" value="御子樂 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>

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 0

無題セッション - 20181214-174026 - OWASP ZAP 2.7.0

標準モード

メソッド: POST URL: http://example.jp/31/31-003.php

Header: デフォルトビュー

Body: タブビュー

レスポンス: デフォルトビュー

サブミットして次のブレークポイントへ移動

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 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  
http://example.jp

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Fri, 14 Dec 2018 09:08:04 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 188
Connection: keep-alive
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<html>
<head><title>登録完了</title></head>
<body>
  氏名:綾小路 さゆり<BR>
  メールアドレス:ayanokouji@sayuri.com<BR>
  性別:女<BR>
  登録されました
</body></html>

```

パラメータ名 値

パラメータ名	値

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
0	18/12/14 17:41:10	POST	http://example.jp/31/31-003.php	200	OK	24 ms	410 bytes	Medium		Form, Hidden
7	18/12/14 18:08:03	POST	http://example.jp/31/31-004.php	200	OK	10 ms	188 bytes	Medium		

アラート 0 1 2 0

現在のスキャン 0 0 0 0 0 0 0 0 0 0

【ブラウザ】

登録完了

example.jp/31/31-004.php

氏名:綾小路 さゆり  
 メールアドレス:ayanokouji@sayuri.com  
 性別:女  
 登録されました

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



```
k?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】



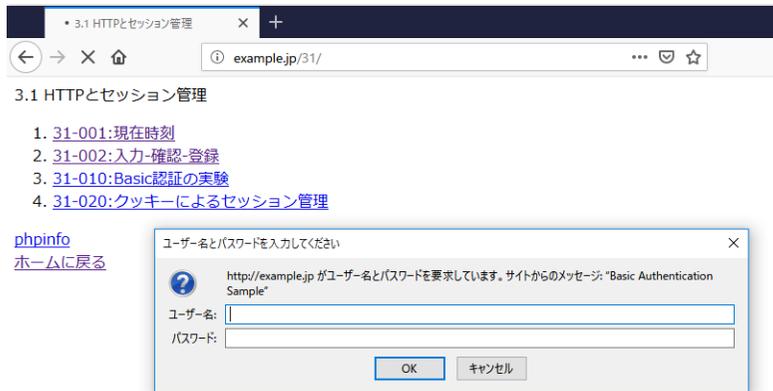
3.1 HTTPとセッション管理

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](#)



```
1 <html>
2 <head><title>3.1 HTTPとセッション管理</title></head>
3 <body>
4 3.1 HTTPとセッション管理
5 <ol>
6 <li><a href="31-001.php">31-001: 現在時刻</a></li>
7 <li><a href="31-002.php">31-002: 入力-確認-登録</a></li>
8 <li><a href="31-010.php">31-010: Basic認証の実験</a></li>
9 <li><a href="31-020.php">31-020: クッキーによるセッション管理</a></li>
10 </ol>
11 <a href="phpinfo.php">phpinfo</a><br>
12 <a href="/">ホームに戻る</a>
13 </body>
14 </html>
15
```



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】

The screenshot shows the OWASP ZAP interface. The left pane shows the context tree with 'http://example.jp' selected. The main pane displays the request and response details.

**Request:**

```
GET http://example.jp/31-010.php HTTP/1.1
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/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp
```

**Response:**

```
HTTP/1.1 401 Unauthorized
Server: nginx/1.10.3
Date: Thu, 13 Dec 2018 13:46:23 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 45
Connection: keep-alive
WWW-Authenticate: Basic realm="Basic Authentication Sample"
X-UA-Compatible: IE=edge

ユーザ名とパスワードが必要です
```

The bottom pane shows a table of request logs:

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
9	18/12/13 22:46:22	GET	http://example.jp/31-010.php	401	Unauthorized	10 ms	45 bytes	Low		

## (認証OKの場合)

### 【ブラウザ】

The browser window shows the URL 'example.jp/31/'. The page content includes a list of links:

- 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

A modal dialog titled 'ユーザー名とパスワードを入力してください' is displayed. It contains the following text:

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: userA

パスワード: ●●●●●●●●

Buttons: OK, キャンセル

The browser window shows the URL 'example.jp/31/31-010.php'. The page content displays a successful login message:

認証しました  
ユーザ名:userA  
パスワード:passwordB

The browser window shows the URL 'http://example.jp/31/31-010.php'. The page content displays the source code of the login page:

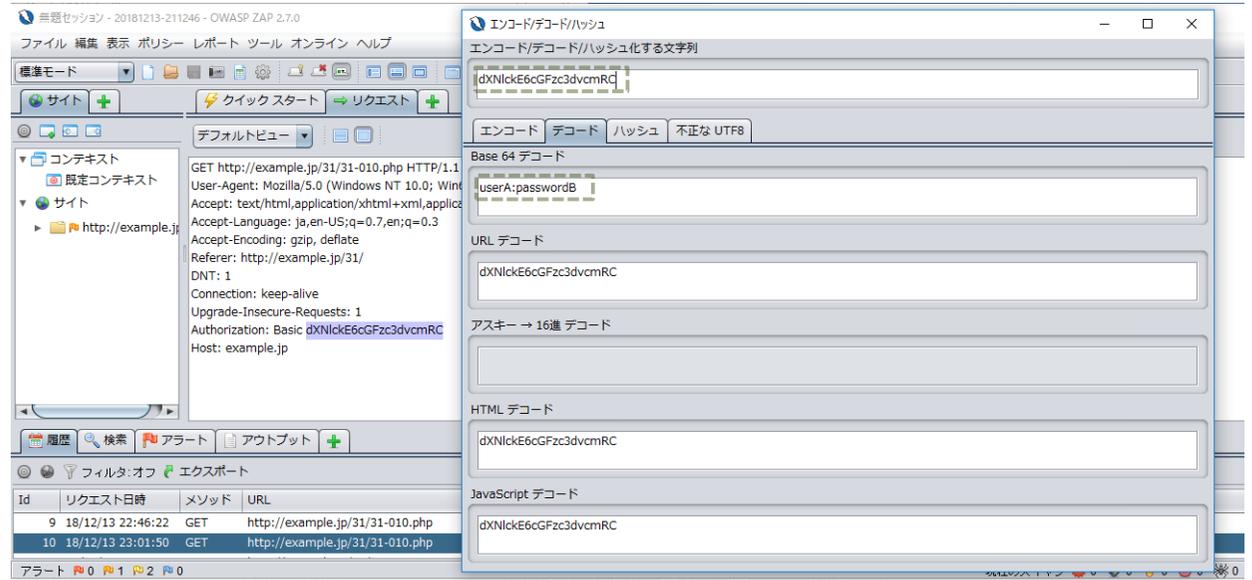
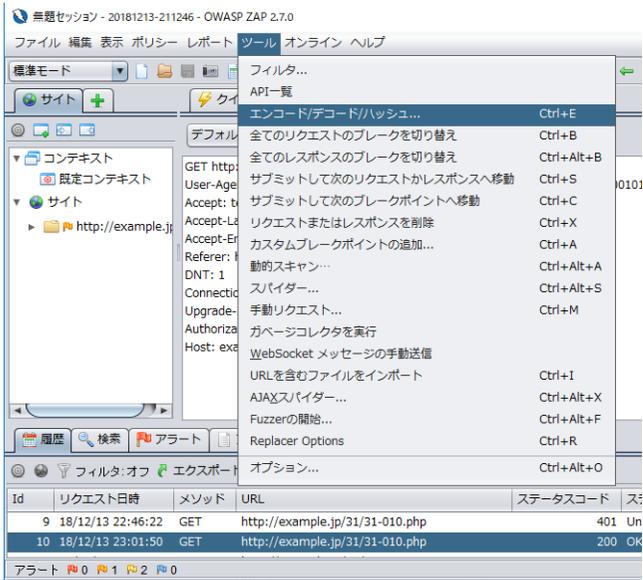
```
1 <body>
2 認証しました<BR>
3 ユーザ名:userA<BR>
4 パスワード:passwordB <BR>
5 </body>
6
```

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

The screenshot displays the OWASP ZAP 2.7.0 interface. The top toolbar includes buttons for 'クイックスタート' (Quick Start), 'リクエスト' (Request), and 'レスポンス' (Response). The main window is split into two panes: 'リクエスト' (Request) on the left and 'レスポンス' (Response) on the right. The request pane shows a GET request to http://example.jp/31/31-010.php with various headers including User-Agent, Accept, Accept-Language, Accept-Encoding, Referer, DNT, Connection, Upgrade-Insecure-Requests, and Authorization: Basic dXNlckE6cGFzc3dvcmRC. The response pane shows an HTTP/1.1 200 OK response with headers for Server, Date, Content-Type, Content-Length, Connection, Vary, and X-UA-Compatible, followed by an HTML body containing a confirmation message in Japanese: '認証しました<BR>ユーザー:userA<BR>パスワード:passwordB <BR>'. Below the main panes is a table of request history.

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
9	18/12/13 22:46:22	GET	http://example.jp/31/31-010.php	401	Unauthorized	10 ms	45 bytes	Low		
10	18/12/13 23:01:50	GET	http://example.jp/31/31-010.php	200	OK	6 ms	92 bytes	Medium		

Alerts: 0 1 2 0  
現在のスキャン: 0 0 0 0 0 0 0 0



## (認証NGの場合)

### 【ブラウザ】

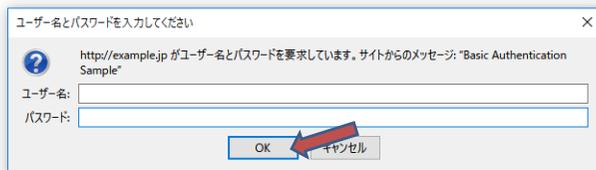


#### 3.1 HTTPとセッション管理

- 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](#)



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】

The screenshot displays the OWASP ZAP 2.7.0 interface. The top-left pane shows the site tree with 'http://example.jp' selected. The middle-left pane shows the request details for 'GET http://example.jp/31/31-010.php HTTP/1.1'. The middle-right pane shows the response details for 'HTTP/1.1 401 Unauthorized'. The bottom pane shows a table of request logs.

**Request Details:**

```
GET http://example.jp/31/31-010.php HTTP/1.1
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/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Authorization: Basic Og==
Host: example.jp
```

**Response Details:**

```
HTTP/1.1 401 Unauthorized
Server: nginx/1.10.3
Date: Thu, 13 Dec 2018 14:13:51 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 45
Connection: keep-alive
WWW-Authenticate: Basic realm="Basic Authentication Sample"
X-UA-Compatible: IE=edge

ユーザ名とパスワードが必要です
```

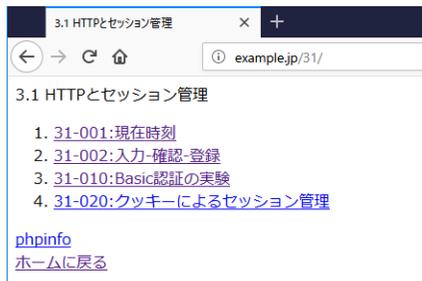
**Request Log Table:**

ID	Request Time	Method	URL	Status Code	Status Code Description	Round Trip Time	Response Body Size	Alert	Note	Tag
17	18/12/13 23:13:50	GET	http://example.jp/31/31-010.php	401	Unauthorized	5 ms	45 bytes	Low		

Alerts: 0 | Current Scans: 0

## 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】

The screenshot shows the OWASP ZAP interface. On the left, the request details are visible: GET http://example.jp/31-020.php HTTP/1.1, 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/, DNT: 1, Connection: keep-alive, Upgrade-Insecure-Requests: 1, Host: example.jp.

The response details on the right show: HTTP/1.1 200 OK, Server: nginx/1.10.3, Date: Fri, 14 Dec 2018 14:15:27 GMT, Content-Type: text/html; charset=UTF-8, Content-Length: 279, Connection: keep-alive, Set-Cookie: PHPSESSID=em949q5n1bfmues287ocaehb05; path=/, Expires: Thu, 19 Nov 1981 08:52:00 GMT, Cache-Control: no-store, no-cache, must-revalidate, Pragma: no-cache, Vary: Accept-Encoding, X-UA-Compatible: IE=edge.

The response body contains HTML code for a login form:
 

```
<html>
<head><title>ログインしてください</title></head>
<body>
<form action="31-021.php" method="POST">
ユーザー名<input type="TEXT" name="ID"><BR>
パスワード<input type="PASSWORD" name="PWD"><BR>
<input type="SUBMIT" value="ログイン">
</form>
</body>
</html>
```

A callout box with a green background and a speech bubble points to the Set-Cookie header in the response, containing the text: "サーバからクライアントに、Set-cookie でクッキーを送信しています".

At the bottom, a table shows the request details:

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
1	18/12/15 10:40:39	GET	http://example.jp/31-020.php	200	OK	17 ms	279 bytes	Medium		Form, Password,...

【ブラウザ】

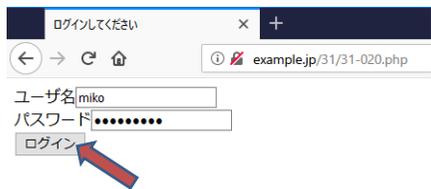
The browser window shows the URL "example.jp/31-020.php". The page content is a login form with the following elements:

- ログインしてください
- ユーザー名
- パスワード
- ログイン

The browser window shows the source code for "view-source:http://example.jp/31-020.php". The code is as follows:

```

1 <html>
2 <head><title>ログインしてください</title></head>
3 <body>
4 <form action="31-021.php" method="POST">
5 ユーザー名<input type="TEXT" name="ID"><BR>
6 パスワード<input type="PASSWORD" name="PWD"><BR>
7 <input type="SUBMIT" value="ログイン">
8 </form>
9 </body>
10 </html>
11
```



```
1 <html>
2 <head><title>ログインしてください</title></head>
3 <body>
4 <form action="31-021.php" method="POST">
5   ユーザー名<input type="TEXT" name="ID"><BR>
6   パスワード<input type="PASSWORD" name="PWD"><BR>
7   <input type="SUBMIT" value="ログイン">
8 </form>
9 </body>
10 </html>
11
```

### 【サーバ: 31/31-021.php】

```
/var/www/html/31/31-021.php - wasbook@192.168.56.101 - エディタ - WinSCP
k?php
session_start(); // セッションの開始
$id = $_POST['ID'];
$password = $_POST['PWD'];
// IDとパスワードのどちらかが空の場合はログイン失敗
if ($id == '' || $password == '') {
    die('ログイン失敗');
}
$_SESSION['ID'] = $id;
}
}
<html>
<head><title>ログイン</title></head>
<body>
ログイン成功しました
<a href="31-022.php">プロフィール</a>
</body>
</html>
```

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】

無害セッション - 20181215-103244 - 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 レスポンス

コンテキスト  
既定コンテキスト  
サイト  
http://example.jp

デフォルトビュー

```
POST http://example.jp/31/31-021.php HTTP/1.1
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/31/31-020.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 21
DNT: 1
Connection: keep-alive
Cookie: PHPSESSID=em949q5n1bfhues287ocae8b05
Upgrade-Insecure-Requests: 1
Host: example.jp
ID=miko&PWD=cat165721
```

クライアントからサーバに、Cookieでクッキー値を送信しています

デフォルトビュー

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Sat, 15 Dec 2018 02:00:33 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 146
Connection: keep-alive
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
X-UA-Compatible: IE=edge
```

```
<html>
<head><title>ログイン</title></head>
<body>
ログイン成功しました
<a href="/31-022.php">プロフィール</a>
</body>
</html>
```

履歴 検索 アラート アウトプット

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
4	18/12/15 11:00:33	POST	http://example.jp/31/31-021.php	200	OK	12 ms	146 bytes	Medium		

アラート 0 1 3 0 現在のスキャン 0 0 0 0 0 0 0 0

## 【ブラウザ】

ログイン

example.jp/31/31-021.php

ログイン成功しました [プロフィール](#)

ログイン

view-source:http://example.jp/31/31-021.php

```
1 <html>
2 <head><title>ログイン</title></head>
3 <body>
4 ログイン成功しました
5 <a href="/31-022.php">プロフィール</a>
6 </body>
7 </html>
8
```

## 【サーバ: 31/31-022.php 】

```
#!/var/www/html/31/31-022.php - wasbook@192.168.56.101 - エディタ - WinSCP
k?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_QUOTES, '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 】

The screenshot shows the browser's developer tools with the 'Network' tab selected. A request and response are visible for the URL http://example.jp/31/31-022.php. The request headers include User-Agent, Accept, Accept-Language, Accept-Encoding, Referer, and DNT. The response headers include HTTP/1.1 200 OK, Server, Date, Content-Type, Content-Length, Connection, Expires, Cache-Control, Pragma, Vary, and X-UA-Compatible. The response body contains HTML code with the user ID 'miko' displayed.

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
5	18/12/15 11:10:28	GET	http://example.jp/31/31-022.php	200	OK	6 ms	93 bytes	Medium		

## 【ブラウザ】

