

4.2 入力処理とセキュリティ

Webアプリケーションの機能と脆弱性の対応

入力	処理	出力	サブシステム	発生する脆弱性の種類	悪用の手口	毒物
入力値検証	処理	表示/HTML	ブラウザ	クロスサイトスクリプティング HTTPヘッダインジェクション	JavaScriptなどの注入 HTTPレスポンスヘッダへの注入	< など 改行
		DB/SQL	RDB	SQLインジェクション	SQLへの注入	; など
		外部コマンド	シェル	OSコマンドインジェクション	コマンドの注入	; など
		メール	メール	メールヘッダインジェクション	メールヘッダ・本文への注入/改変	改行
		ファイル	ファイル	デレトリトラバーサル	JavaScriptなどの注入	ログイン状態
重要な処理			クリスサイトリクエストフォージェリ	セッションIDの強制	セッションID	
認証			セッションフィクセーション 認証不備	セッションIDの強制	セッションID	
認可			認可不備	認証後の権限設定		

入力処理

文字エンコーディングの妥当性検証
文字エンコーディングの変換(必要時)

言語	自動変換	スクリプトに記述
PHP	php.iniなど	mb_convert_encoding
Perl	x	Encode::decode
Java	setCharacterEncoding	Stringクラス
ASP.NET	Web.config	x

入力値の妥当性検証

バイナリセーフに注意が必要

制御文字の中でも特に「%00」などで表すことができるNULL文字を、Webアプリケーションに渡される文字列の中に混入することで、意図しない動作を引き起こす可能性がある。
バイナリセーフの場合、NULL文字も文字として扱われるが、非バイナリセーフの場合、NULL文字を「文字列の終了」と見なしてしまうため、以降の文字を扱えず、この挙動の差から問題が発生する。
また、NULL文字だけではなく改行(%0d%0a)のような制御文字で問題が発生する場合も多く存在する。このような問題が起きないようにするために、制御文字が混入することを想定し、バイナリセーフか否かを意識する必要がある。

42-001:文字エンコーディングのチェック(正常系)/(不正な文字エンコーディング)

【ブラウザ】



4.2 入力処理とセキュリティ

- 42-001:文字エンコーディングのチェック(正常系)
- 42-001:文字エンコーディングのチェック(不正な文字エンコーディング)
- 42-002:ヌルバイト攻撃 (正常系)
- 42-002:ヌルバイト攻撃 (XSS)
- 42-010:正規表現による入力値検証の例(1) (preg_match) 英数字1文字~5文字(正常系)
- 42-010:正規表現による入力値検証の例(1) (preg_match) 英数字1文字~5文字(改行込み)
- 42-011:正規表現による入力値検証の例(1) (^と\$使用) 英数字1文字~5文字(正常系)
- 42-011:正規表現による入力値検証の例(1) (^と\$使用) 英数字1文字~5文字(改行込み)
- 42-012:正規表現による入力値検証の例(1) (mb_ereg) 英数字1文字~5文字(正常系)
- 42-012:正規表現による入力値検証の例(1) (mb_ereg) 英数字1文字~5文字(改行込み)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(正常系)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(ヌルバイトを含む)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(正常系)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(ヌルバイトを含む)
- 42-020:4.2節のまとめ
- 42-020:4.2節のまとめ(改行込み)
- 42-020:4.2節のまとめ(ヌルバイト込み)

[phpinfo](#)

[ホームに戻る](#)

【サーバ: 42/42-001.php 】

`/var/www/html/42/42-001.php - wasbook@192.168.56.101 - エディタ - WinSCP`

```
<?php
$name = isset($_GET['name']) ? $_GET['name'] : '';
// 文字エンコーディング (Shift_JIS) のチェック
if (! mb_check_encoding($name, 'Shift_JIS')) {
    die('文字エンコーディングが不正です');
}
// 文字エンコーディングの変換 (Shift_JIS→UTF-8)
$name = mb_convert_encoding($name, 'UTF-8', 'Shift_JIS');
?>
<body>
名前:php echo htmlspecialchars($name, ENT_QUOTES, 'UTF-8'); ?>です
</body>
```



【ブラウザ→サーバ: リクエスト 42/42-001.php → レスポンス】(正常系)

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

- Request:**

```
GET http://example.jp/42/42-001.php?name=%8ER%93 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
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/42/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp
```
- Response:**

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 17 Dec 2018 13:29:07 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
X-UA-Compatible: IE=edge

<body>
名前は山田です
</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
4	18/12/17 22:29:07	GET	http://example.jp/42/42-001.php?name=%8...	200	OK	99 ms	37 bytes	Medium		

【ブラウザ→サーバ: リクエスト 42/42-001.php → レスポンス】(不正な文字エンコーディング)

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

- Request:**

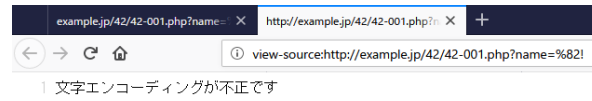
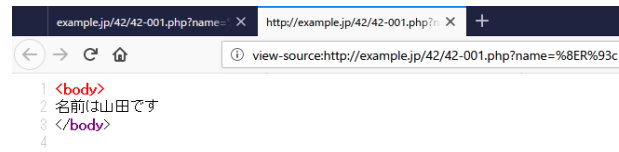
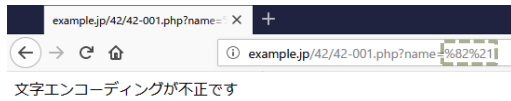
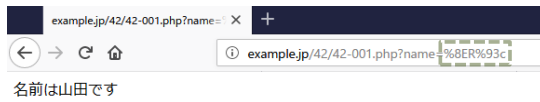
```
GET http://example.jp/42/42-001.php?name=%82%2 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://example.jp/42/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp
```
- Response:**

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 17 Dec 2018 13:38:47 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
X-UA-Compatible: IE=edge

文字エンコーディングが不正です
```
- 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
6	18/12/17 22:38:47	GET	http://example.jp/42/42-001.php?name=%8...	200	OK	22 ms	45 bytes	Medium		

【ブラウザ】



42-002:ヌルバイト攻撃(正常系)/(XSS)

【ブラウザ】



4.2 入力処理とセキュリティ

- 42-001:文字エンコーディングのチェック(正常系)
- 42-001:文字エンコーディングのチェック(不正な文字エンコーディング)
- 42-002:ヌルバイト攻撃(正常系)
- 42-002:ヌルバイト攻撃(XSS)
- 42-010:正規表現による入力値検証の例(1)(preg_match) 英数字1文字~5文字(正常系)
- 42-010:正規表現による入力値検証の例(1)(preg_match) 英数字1文字~5文字(改行込み)
- 42-011:正規表現による入力値検証の例(1)(^と\$使用) 英数字1文字~5文字(正常系)
- 42-011:正規表現による入力値検証の例(1)(^と\$使用) 英数字1文字~5文字(改行込み)
- 42-012:正規表現による入力値検証の例(1)(mb_ereg) 英数字1文字~5文字(正常系)
- 42-012:正規表現による入力値検証の例(1)(mb_ereg) 英数字1文字~5文字(改行込み)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(正常系)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(ヌルバイトを含む)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(正常系)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(ヌルバイトを含む)
- 42-020:4.2節のまとめ
- 42-020:4.2節のまとめ(改行込み)
- 42-020:4.2節のまとめ(ヌルバイト込み)

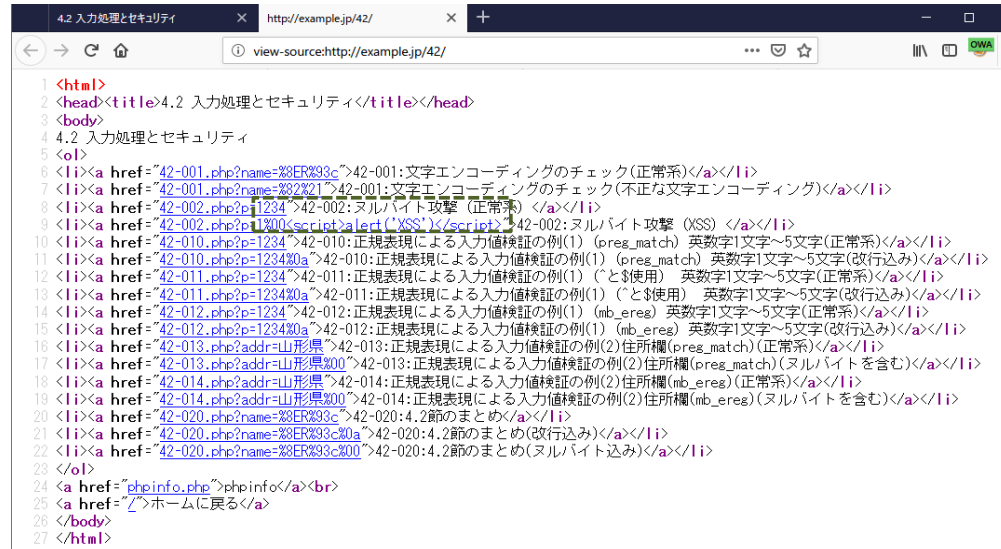
[phpinfo](#)

[ホームに戻る](#)

【サーバ: 42/42-002.php】

/var/www/html/42/42-002.php - wasbook@192.168.56.101 - 1

```
body>
<?php
    $p = $_GET['p'];
    if (ereg("^[0-9]+$", $p) === FALSE) {
        die('整数値を入力してください');
    }
    echo $p;
?>
</body>
```



【ブラウザ→サーバ: リクエスト 42/42-002.php → レスポンス】(正常系)

The screenshot shows the OWASP ZAP interface with a normal request and response. The request is a GET to http://example.jp/42/42-002.php?p=1234. The response is a 200 OK from nginx/1.10.3, with a content type of text/html and a body containing the text "1234".

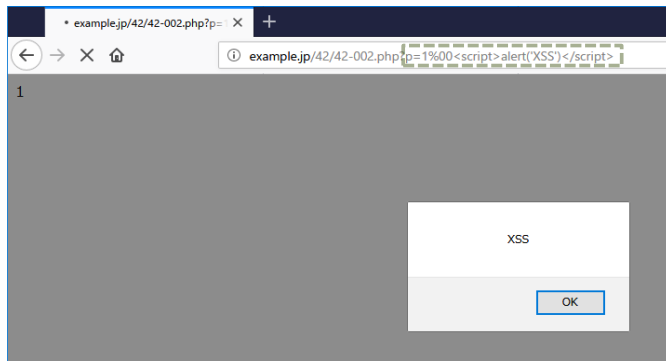
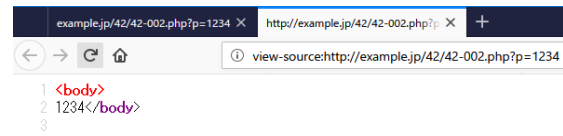
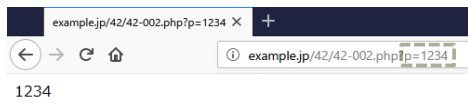
Id	リクエスト日時	メソッド	URL	ステータスコード	ステータスコード説明	ラウンドトリップタイム	レスポンスボディサイズ	検出アラート	ノート	タグ
7	18/12/17 22:48:27	GET	http://example.jp/42/42-002.php?p=1234	200	OK	26 ms	19 bytes	Medium		

【ブラウザ→サーバ: リクエスト 42/42-002.php → レスポンス】(XSS)

The screenshot shows the OWASP ZAP interface with an XSS request and response. The request is a GET to http://example.jp/42/42-002.php?p=1%00%3Cscript%3Ealert(%27XSS%27)%3C/script%3E. The response is a 200 OK from nginx/1.10.3, with a content type of text/html and a body containing the text "1<script>alert('XSS')</script></body>".

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	18/12/17 22:56:04	GET	http://example.jp/42/42-002.php?p=1%00%3Cscript%3Ealert(%27XSS%27)%3C/script%3E	200	OK	28 ms	46 bytes	Medium	Script	

【ブラウザ】



42-010:正規表現による入力値検証の例(1)(preg_match)英数字1文字~5文字(正常系)/(改行込み)

【ブラウザ】



4.2 入力処理とセキュリティ

- 42-001:文字エンコーディングのチェック(正常系)
- 42-001:文字エンコーディングのチェック(不正な文字エンコーディング)
- 42-002:ヌルバイト攻撃 (正常系)
- 42-002:ヌルバイト攻撃 (XSS)
- 42-010:正規表現による入力値検証の例(1) (preg_match) 英数字1文字~5文字(正常系)
- 42-010:正規表現による入力値検証の例(1) (preg_match) 英数字1文字~5文字(改行込み)
- 42-011:正規表現による入力値検証の例(1) (^と\$使用) 英数字1文字~5文字(正常系)
- 42-011:正規表現による入力値検証の例(1) (^と\$使用) 英数字1文字~5文字(改行込み)
- 42-012:正規表現による入力値検証の例(1) (mb_ereg) 英数字1文字~5文字(正常系)
- 42-012:正規表現による入力値検証の例(1) (mb_ereg) 英数字1文字~5文字(改行込み)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(正常系)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(ヌルバイトを含む)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(正常系)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(ヌルバイトを含む)
- 42-020:4.2節のまとめ
- 42-020:4.2節のまとめ(改行込み)
- 42-020:4.2節のまとめ(ヌルバイト込み)

[phpinfo](#)

[ホームに戻る](#)

【サーバ: 42/42-010.php】

```
/var/www/html/42/42-010.php - wasbook@192.168.56.101 - エディタ - WinSCP  
k?php  
$p = filter_input(INPUT_GET, 'p');  
if (preg_match('/^[a-z0-9]{1,5}$/i', $p) != 1) {  
    die('1文字以上5文字以下の英数字を入力してください');  
}  
>>  
<body>  
pisk?php echo htmlspecialchars($p, ENT_QUOTES, 'UTF-8'); >>です  
</body>
```



【ブラウザ→サーバ: リクエスト 42/42-010.php → レスポンス】(正常系)

無題セッション - 20181217-114903 - 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

```
GET http://example.jp/42/42-010.php?ip=1234 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
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/42/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp
```

デフォルトビュー

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 17 Dec 2018 14:14:21 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
X-UA-Compatible: IE=edge

<body>
pは1234です
</body>
```

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
10	18/12/17 23:14:21	GET	http://example.jp/42/42-010.php?ip=1234	200	OK	28 ms	31 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

【ブラウザ→サーバ: リクエスト 42/42-010.php → レスポンス】(改行込み)

無題セッション - 20181217-114903 - 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

```
GET http://example.jp/42/42-010.php?ip=1234%0a HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
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/42/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp
```

デフォルトビュー

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 17 Dec 2018 14:17:01 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
X-UA-Compatible: IE=edge

1文字以上5文字以下の英数字を入力してください
```

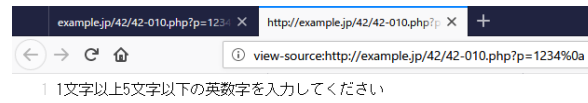
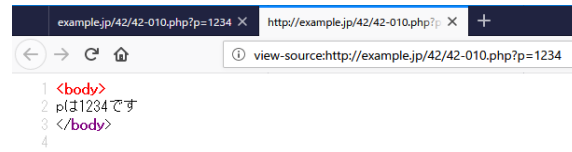
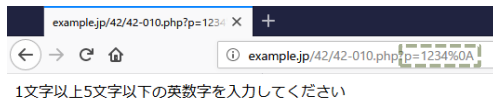
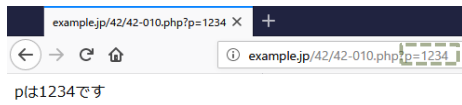
履歴 検索 アラート アウトプット +

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
11	18/12/17 23:17:01	GET	http://example.jp/42/42-010.php?ip=1234%0a	200	OK	21 ms	65 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

【ブラウザ】



42-011:正規表現による入力値検証の例(1)(^と\$使用) 英数字1文字~5文字(正常系)

【ブラウザ】



4.2 入力処理とセキュリティ

- 42-001:文字エンコーディングのチェック(正常系)
- 42-001:文字エンコーディングのチェック(不正な文字エンコーディング)
- 42-002:ヌルバイト攻撃 (正常系)
- 42-002:ヌルバイト攻撃 (XSS)
- 42-010:正規表現による入力値検証の例(1) (preg_match) 英数字1文字~5文字(正常系)
- 42-010:正規表現による入力値検証の例(1) (preg_match) 英数字1文字~5文字(改行込み)
- 42-011:正規表現による入力値検証の例(1) (^と\$使用) 英数字1文字~5文字(正常系)
- 42-011:正規表現による入力値検証の例(1) (^と\$使用) 英数字1文字~5文字(改行込み)
- 42-012:正規表現による入力値検証の例(1) (mb_ereg) 英数字1文字~5文字(正常系)
- 42-012:正規表現による入力値検証の例(1) (mb_ereg) 英数字1文字~5文字(改行込み)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(正常系)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(ヌルバイトを含む)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(正常系)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(ヌルバイトを含む)
- 42-020:4.2節のまとめ
- 42-020:4.2節のまとめ(改行込み)
- 42-020:4.2節のまとめ(ヌルバイト込み)

[phpinfo](#)

[ホームに戻る](#)



【サーバ: 42/42-011.php 】

```
#!/var/www/html/42/42-011.php - wasbook@192.168.56.101 - エディタ - WinSCP  
k?php  
$p = filter_input(INPUT_GET, 'p');  
if(preg_match("/[a-z0-9]{1,5}$/i", $p) != 1) {  
die("1文字以上5文字以下の英数字を入力してください");  
}  
?>  
<body>  
pは?php echo htmlspecialchars($p, ENT_QUOTES, "UTF-8"); ?>です  
</body>
```



【ブラウザ→サーバ: リクエスト 42/42-011.php → レスポンス 】(正常系)

無題セッション - 20181217-114903 - OWASP ZAP 2.7.0

デフォルトビュー

コンテキスト: GET http://example.jp/42/42-011.php?p=1234 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
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/42/
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: Mon, 17 Dec 2018 14:25:36 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
X-UA-Compatible: IE=edge

<body>
pは1234です
</body>

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
12	18/12/17 23:25:36	GET	http://example.jp/42/42-011.php?p=1234	200	OK	21 ms	31 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

【ブラウザ→サーバ: リクエスト 42/42-011.php → レスポンス 】(改行込み)

無題セッション - 20181217-114903 - OWASP ZAP 2.7.0

デフォルトビュー

コンテキスト: GET http://example.jp/42/42-011.php?p=1234%0a HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
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/42/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp

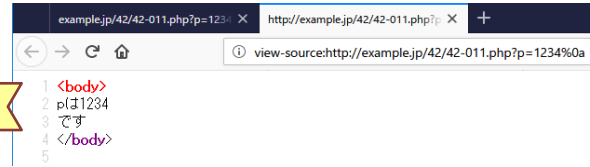
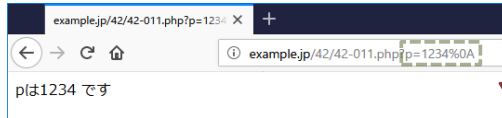
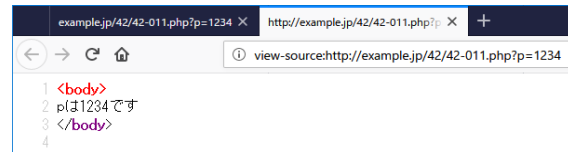
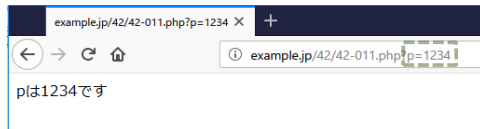
レスポンス: HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 17 Dec 2018 14:27:16 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
X-UA-Compatible: IE=edge

<body>
pは1234
です
</body>

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
13	18/12/17 23:27:16	GET	http://example.jp/42/42-011.php?p=1234%0a	200	OK	24 ms	32 bytes	Medium		

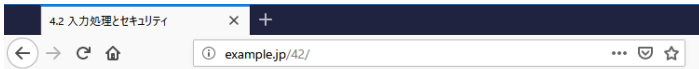
アラート 0 1 2 0 現在のスキャン 0 0 0 0 0 0 0 0

【ブラウザ】



42-012:正規表現による入力値検証の例(1)(mb_ereg)英数字1文字~5文字(正常系)

【ブラウザ】



4.2 入力処理とセキュリティ

- 42-001:文字エンコーディングのチェック(正常系)
- 42-001:文字エンコーディングのチェック(不正な文字エンコーディング)
- 42-002:ヌルバイト攻撃 (正常系)
- 42-002:ヌルバイト攻撃 (XSS)
- 42-010:正規表現による入力値検証の例(1) (preg_match) 英数字1文字~5文字(正常系)
- 42-010:正規表現による入力値検証の例(1) (preg_match) 英数字1文字~5文字(改行込み)
- 42-011:正規表現による入力値検証の例(1) (^とs使用) 英数字1文字~5文字(正常系)
- 42-011:正規表現による入力値検証の例(1) (^とs使用) 英数字1文字~5文字(改行込み)
- 42-012:正規表現による入力値検証の例(1) (mb_ereg) 英数字1文字~5文字(正常系)
- 42-012:正規表現による入力値検証の例(1) (mb_ereg) 英数字1文字~5文字(改行込み)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(正常系)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(ヌルバイトを含む)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(正常系)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(ヌルバイトを含む)
- 42-020:4.2節のまとめ
- 42-020:4.2節のまとめ(改行込み)
- 42-020:4.2節のまとめ(ヌルバイト込み)

[phpinfo](#)

[ホームに戻る](#)

【サーバ: 42/42-012.php 】

```
/var/www/html/42/42-012.php - wasbook@192.168.56.101 - エディタ - WinSCP
k?php
// mb_regex_encodingは内部エンコーディングが設定されている場合は省略可能
mb_regex_encoding('UTF-8'); // プログラムの先頭で一度設定すればよい
$_p = filter_input(INPUT_GET, 'p');
if (mb_ereg('\A[a-zA-Z0-9]{1,5}z', $_p) === false) {
    die("1文字以上5文字以下の英数字を入力してください");
}
?>
<body>
pは?php echo htmlspecialchars($_p, ENT_QUOTES, 'UTF-8'); ?>;
</body>
```



mb_ereg は戻り値として、整数型か論理値を返すので、型を区別する比較演算子「===」を用いる

※ \d \w の扱いに注意

\d は数字にマッチ \w は英数字とアンダースコアにマッチしますが、\d \w は全角文字にもマッチします。たとえば\d は全角の数字にもマッチします。(Unicodeの場合) [a-z0-9]などの文字クラスの指定のほうが安全です。

【ブラウザ→サーバ: リクエスト 42/42-012.php → レスポンス】(正常系)

無害セッション - 20181217-114903 - 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

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

GET http://example.jp/42/42-012.php?p=1234 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
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/42/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 17 Dec 2018 14:39:10 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
X-UA-Compatible: IE=edge

<body>
pは1234です
</body>

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
14	18/12/17 23:39:09	GET	http://example.jp/42/42-012.php?p=1234	200	OK	24 ms	30 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

【ブラウザ→サーバ: リクエスト 42/42-012.php → レスポンス】(改行込み)

無害セッション - 20181217-114903 - 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

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

GET http://example.jp/42/42-012.php?p=1234%0a HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
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/42/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 17 Dec 2018 14:41:42 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
X-UA-Compatible: IE=edge

1文字以上5文字以下の英数字を入力してください

履歴 検索 アラート アウトプット

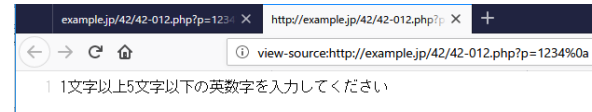
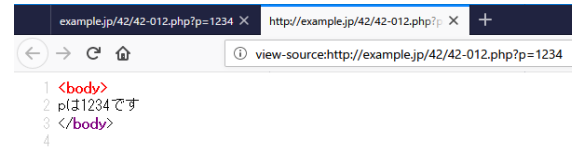
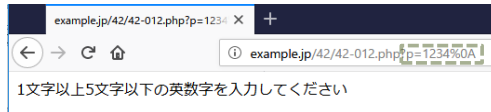
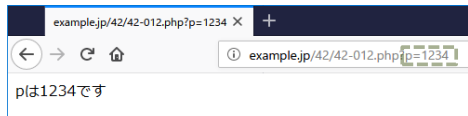
フィルタ: オフ エクスポート

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
15	18/12/17 23:41:42	GET	http://example.jp/42/42-012.php?p=1234%0a	200	OK	24 ms	65 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

【ブラウザ】



42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(正常系)/(ヌルバイトを含む)

【ブラウザ】



4.2 入力処理とセキュリティ

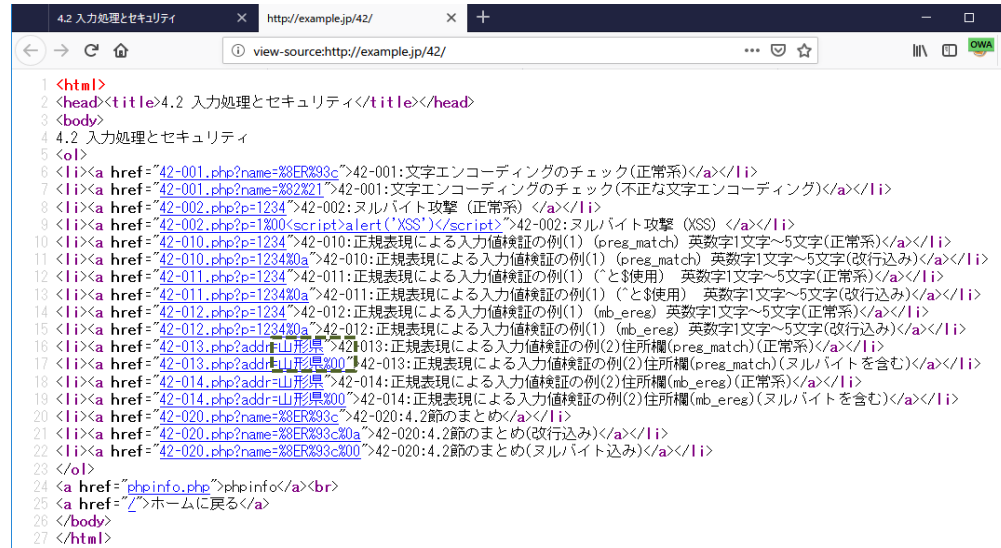
- 42-001:文字エンコーディングのチェック(正常系)
- 42-001:文字エンコーディングのチェック(不正な文字エンコーディング)
- 42-002:ヌルバイト攻撃 (正常系)
- 42-002:ヌルバイト攻撃 (XSS)
- 42-010:正規表現による入力値検証の例(1) (preg_match) 英数字1文字~5文字(正常系)
- 42-010:正規表現による入力値検証の例(1) (preg_match) 英数字1文字~5文字(改行込み)
- 42-011:正規表現による入力値検証の例(1) (^と\$使用) 英数字1文字~5文字(正常系)
- 42-011:正規表現による入力値検証の例(1) (^と\$使用) 英数字1文字~5文字(改行込み)
- 42-012:正規表現による入力値検証の例(1) (mb_ereg) 英数字1文字~5文字(正常系)
- 42-012:正規表現による入力値検証の例(1) (mb_ereg) 英数字1文字~5文字(改行込み)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(正常系)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(ヌルバイトを含む)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(正常系)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(ヌルバイトを含む)
- 42-020:4.2節のまとめ
- 42-020:4.2節のまとめ(改行込み)
- 42-020:4.2節のまとめ(ヌルバイト込み)

[phpinfo](#)

[ホームに戻る](#)

【サーバ: 42/42-013.php】

```
/var/www/html/42/42-013.php - wasbook@192.168.56.101 - エディタ - WinSCP
k?php
  $addr = filter_input(INPUT_GET, 'addr');
  if (preg_match("/^[A[^\^cntrl:]]{1,30}\z/u", $addr) != 1) {
    die("30文字以内で住所を入力してください(必須項目)。改行やタブなどの制御文字は使用できません");
  }
?>
<body>
addrは<?php echo htmlspecialchars($addr, ENT_QUOTES, 'UTF-8');>です
</body>
```



【ブラウザ→サーバ: リクエスト 42/42-013.php → レスポンス】(正常系)

The screenshot shows the OWASP ZAP interface with a request and response view. The request is a GET request to http://example.jp/42/42-013.php with a URL-encoded address parameter. The response is a 200 OK status with headers including Server: nginx/1.10.3, Date: Mon, 17 Dec 2018 14:50:39 GMT, and Content-Type: text/html; charset=UTF-8. The response body contains the text: <body> addrは山形県です </body>.

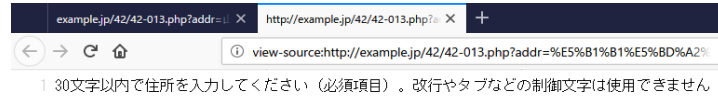
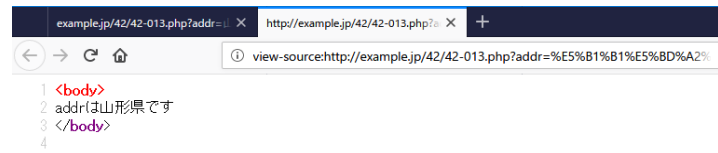
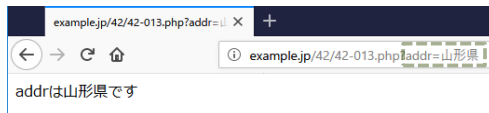
Id	リクエスト日時	メソッド	URL	ステータスコード	ステータスコード説明	ラウンドトリップタイム	レスポンスボディサイズ	検出アラート	ノート	タグ
17	18/12/17 23:50:39	GET	http://example.jp/42/42-013.php?addr=%E5...	200	OK	23 ms	38 bytes	Medium		

【ブラウザ→サーバ: リクエスト 42/42-013.php → レスポンス】(ヌルバイトを含む)

The screenshot shows the OWASP ZAP interface with a request and response view. The request is a GET request to http://example.jp/42/42-013.php with a URL-encoded address parameter containing null bytes. The response is a 200 OK status with headers including Server: nginx/1.10.3, Date: Mon, 17 Dec 2018 14:53:00 GMT, and Content-Length: 131. The response body contains the text: 30文字以内で住所を入力してください (必須項目)。改行やタブなどの制御文字は使用できません.

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
18	18/12/17 23:53:00	GET	http://example.jp/42/42-013.php?addr=%E5...	200	OK	27 ms	131 bytes	Medium		

【ブラウザ】



42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(正常系)/(ヌルバイトを含む)

【ブラウザ】



4.2 入力処理とセキュリティ

- 42-001:文字エンコーディングのチェック(正常系)
- 42-001:文字エンコーディングのチェック(不正な文字エンコーディング)
- 42-002:ヌルバイト攻撃 (正常系)
- 42-002:ヌルバイト攻撃 (XSS)
- 42-010:正規表現による入力値検証の例(1) (preg_match) 英数字1文字~5文字(正常系)
- 42-010:正規表現による入力値検証の例(1) (preg_match) 英数字1文字~5文字(改行込み)
- 42-011:正規表現による入力値検証の例(1) (^と\$使用) 英数字1文字~5文字(正常系)
- 42-011:正規表現による入力値検証の例(1) (^と\$使用) 英数字1文字~5文字(改行込み)
- 42-012:正規表現による入力値検証の例(1) (mb_ereg) 英数字1文字~5文字(正常系)
- 42-012:正規表現による入力値検証の例(1) (mb_ereg) 英数字1文字~5文字(改行込み)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(正常系)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(ヌルバイトを含む)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(正常系)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(ヌルバイトを含む)
- 42-020:4.2節のまとめ
- 42-020:4.2節のまとめ(改行込み)
- 42-020:4.2節のまとめ(ヌルバイト込み)

[phpinfo](#)

[ホームに戻る](#)

【サーバ: 42/42-014.php 】

```
/var/www/html/42/42-014.php - wasbook@192.168.56.101 - エディタ - WinSCP
k?php
$filter_input(INPUT_GET, 'addr');
if (mb_ereg('\A[!:"\']{1,30}\z', $addr) === false) {
    die('30文字以内で住所を入力してください (必須項目)。改行やタブなどの制御文字は使用できません');
}
?>
<body>
addr: k?php echo htmlspecialchars($addr, ENT_QUOTES, 'UTF-8'); ?>
</body>
```



【ブラウザ→サーバ: リクエスト 42/42-014.php → レスポンス】(正常系)

The screenshot shows the OWASP ZAP interface with a request and response view. The request is a GET request to `http://example.jp/42/42-014.php?addr=%E5%B1%B1%E5%BD%A2%E7%9C%8C`. The response is an HTTP/1.1 200 OK from nginx/1.10.3, with a content type of `text/html; charset=UTF-8` and a body containing `addrは山形県です`.

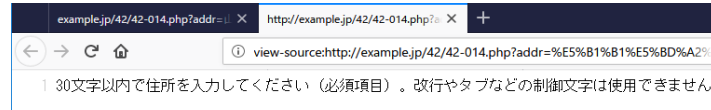
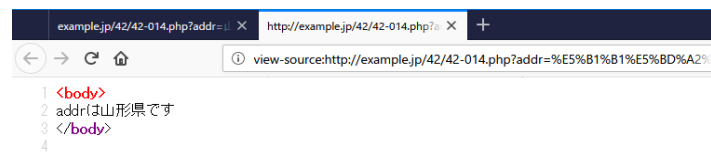
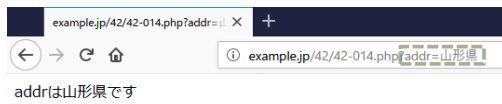
Id	リクエスト日時	メソッド	URL	ステータスコード	ステータスコード説明	ラウンドトリップタイム	レスポンスボディサイズ	検出アラート	ノート	タグ
19	18/12/18 0:00:32	GET	http://example.jp/42/42-014.php?addr=%E5...	200	OK	29 ms	38 bytes	Medium		

【ブラウザ→サーバ: リクエスト 42/42-014.php → レスポンス】(ヌルバイトを含む)

The screenshot shows the OWASP ZAP interface with a request and response view. The request is a GET request to `http://example.jp/42/42-014.php?addr=%E5%B1%B1%E5%BD%A2%E7%9C%8C%00`. The response is an HTTP/1.1 200 OK from nginx/1.10.3, with a content type of `text/html; charset=UTF-8` and a body containing a message in Japanese: `30文字以内で住所を入力してください(必須項目)。改行やタブなどの制御文字は使用できません`.

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
20	18/12/18 0:02:01	GET	http://example.jp/42/42-014.php?addr=%E5...	200	OK	22 ms	131 bytes	Medium		

【ブラウザ】



42-020:4.2節のまとめ(改行込み)/(ヌルバイト込み)

【ブラウザ】



4.2 入力処理とセキュリティ

- 42-001:文字エンコーディングのチェック(正常系)
- 42-001:文字エンコーディングのチェック(不正な文字エンコーディング)
- 42-002:ヌルバイト攻撃(正常系)
- 42-002:ヌルバイト攻撃(XSS)
- 42-010:正規表現による入力値検証の例(1)(preg_match) 英数字1文字~5文字(正常系)
- 42-010:正規表現による入力値検証の例(1)(preg_match) 英数字1文字~5文字(改行込み)
- 42-011:正規表現による入力値検証の例(1)(^と\$使用) 英数字1文字~5文字(正常系)
- 42-011:正規表現による入力値検証の例(1)(^と\$使用) 英数字1文字~5文字(改行込み)
- 42-012:正規表現による入力値検証の例(1)(mb_ereg) 英数字1文字~5文字(正常系)
- 42-012:正規表現による入力値検証の例(1)(mb_ereg) 英数字1文字~5文字(改行込み)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(正常系)
- 42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(ヌルバイトを含む)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(正常系)
- 42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(ヌルバイトを含む)
- 42-020:4.2節のまとめ
- 42-020:4.2節のまとめ(改行込み)
- 42-020:4.2節のまとめ(ヌルバイト込み)

[phpinfo](#)

[ホームに戻る](#)

【サーバ: 42/42-020.php】

```
#!/var/www/html/42/42-020.php - wasbook@192.168.56.101 - エディタ - WinSCP
k?php
// パラメータを取得し、文字エンコーディングチェックと変換
// 入力値検証まで行う関数
// $key : GETパラメータ名
// $pattern : 入力値検証用正規表現文字列
// $error : 入力値検証時のエラーメッセージ
// 戻り値 : 取得したパラメータ (string)
function getParam($key, $pattern, $error) {
    $val = filter_input(INPUT_GET, $key);
    // 文字エンコーディング (Shift_JIS) のチェック
    if (!mb_check_encoding($val, 'Shift_JIS')) {
        die('文字エンコーディングが不正です');
    }
    // 文字エンコーディングの変換 (Shift_JIS→UTF-8)
    $val = mb_convert_encoding($val, 'UTF-8', 'Shift_JIS');
    if (preg_match($pattern, $val) != 1) {
        die($error);
    }
    return $val;
}
// パラメータ取得関数の呼び出し
$name = getParam('name', '/A[[:^cntrl:]]{1,20}\z/u',
    '20文字以内で氏名を入力してください(必須項目)。制御文字は使用できません');
?>
<body>
名前:<?php echo htmlspecialchars($name, ENT_QUOTES, 'UTF-8'); ?>です
</body>
```



【ブラウザ→サーバ: リクエスト 42/42-020.php → レスポンス】(正常系)

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

- Request:** GET http://example.jp/42/42-020.php?name=%8E%93c HTTP/1.1
- Response:** HTTP/1.1 200 OK
- Response Body:** <body> 名前は山田です </body>
- 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
21	18/12/18 0:12:51	GET	http://example.jp/42/42-020.php?name=%8...	200	OK	24 ms	37 bytes	Medium		

【ブラウザ→サーバ: リクエスト 42/42-020.php → レスポンス】(改行込み)

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

- Request:** GET http://example.jp/42/42-020.php?name=%8E%93c%0a HTTP/1.1
- Response:** HTTP/1.1 200 OK
- Response Body:** 20文字以内で氏名を入力してください(必須項目)。制禁文字は使用できません
- 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
22	18/12/18 0:14:44	GET	http://example.jp/42/42-020.php?name=%8...	200	OK	33 ms	107 bytes	Medium		

【ブラウザ→サーバ: リクエスト 42/42-020.php → レスポンス】(ヌルバイト込み)

The screenshot shows the OWASP ZAP interface. The left pane displays the request details for a GET request to `http://example.jp/42/42-020.php?name=%8ER%93c%00`. The right pane displays the response details, which is a 200 OK status with headers including `Server: nginx/1.10.3`, `Date: Mon, 17 Dec 2018 15:15:36 GMT`, and `Content-Type: text/html; charset=UTF-8`. The response body contains the text "20文字以内で氏名を入力してください(必須項目)。制御文字は使用できません".

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
23	18/12/18 0:15:36	GET	http://example.jp/42/42-020.php?name=%8...	200	OK	23 ms	107 bytes	Medium		

【ブラウザ】

The browser address bar shows the URL `http://example.jp/42/42-020.php?name=%8ER%93c`. Below the address bar, the text "名前は山田です" is displayed.

The browser view-source page shows the HTML body content:

```

1 <body>
2 名前は山田です
3 </body>
4

```

The browser address bar shows the URL `http://example.jp/42/42-020.php?name=%8ER%93c%0a`. Below the address bar, the text "20文字以内で氏名を入力してください(必須項目)。制御文字は使用できません" is displayed.

The browser view-source page shows the HTML body content:

```

1 20文字以内で氏名を入力してください(必須項目)。制御文字は使用できません

```

The browser address bar shows the URL `http://example.jp/42/42-020.php?name=%8ER%93c%00`. Below the address bar, the text "20文字以内で氏名を入力してください(必須項目)。制御文字は使用できません" is displayed.

The browser view-source page shows the HTML body content:

```

1 20文字以内で氏名を入力してください(必須項目)。制御文字は使用できません

```