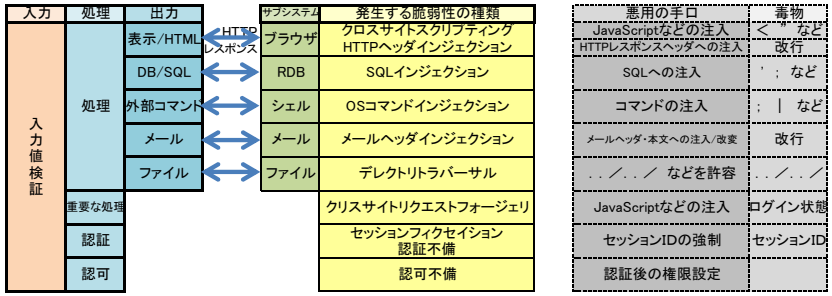
4.2 入力処理とセキュリティ

Webアプリケーションの機能と脆弱性の対応



悪用の手口	毒物
JavaScriptなどの注入	< " など
HTTPレスポンスヘッダへの注入	改行
SQLへの注入	' ; など
コマンドの注入	; | など
メールヘッダ・本文への注入/改変	改行
. . /. . / などを許容	. . /. . /
JavaScriptなどの注入	ログイン状態
セッションIDの強制	セッションID
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	×	Encode::decode
Java	setCharacterEncoding	Stringクラス
ASP.NET	Web.config	×

入力値の妥当性検証

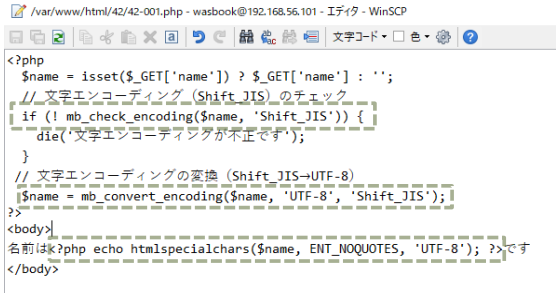
バイナリセーフに注意が必要

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。

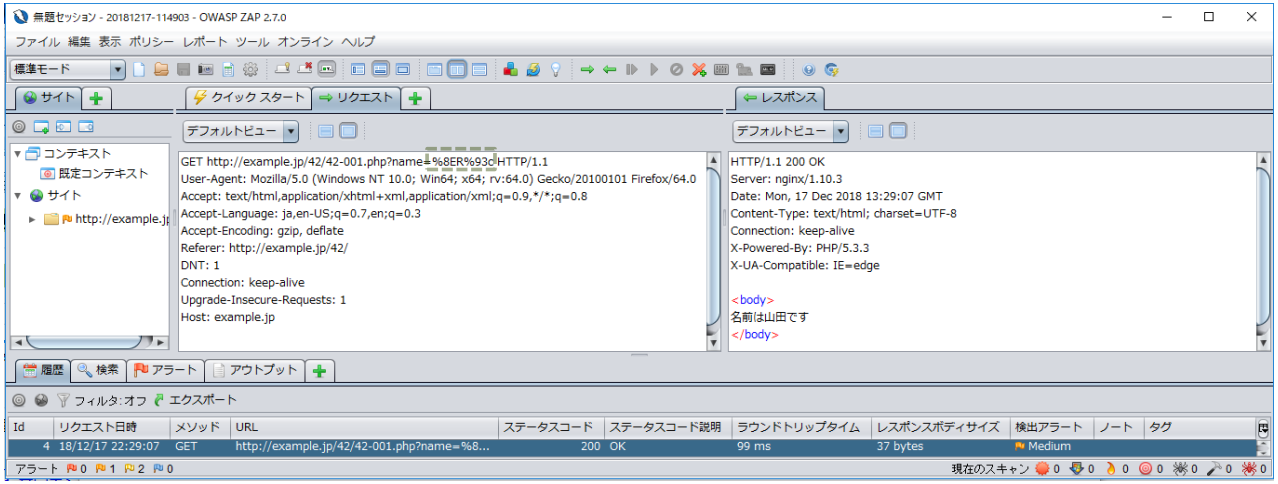
42-001:文字エンコーディングのチェック(正常系)/(不正な文字エンコーディング)

【ブラウザ】

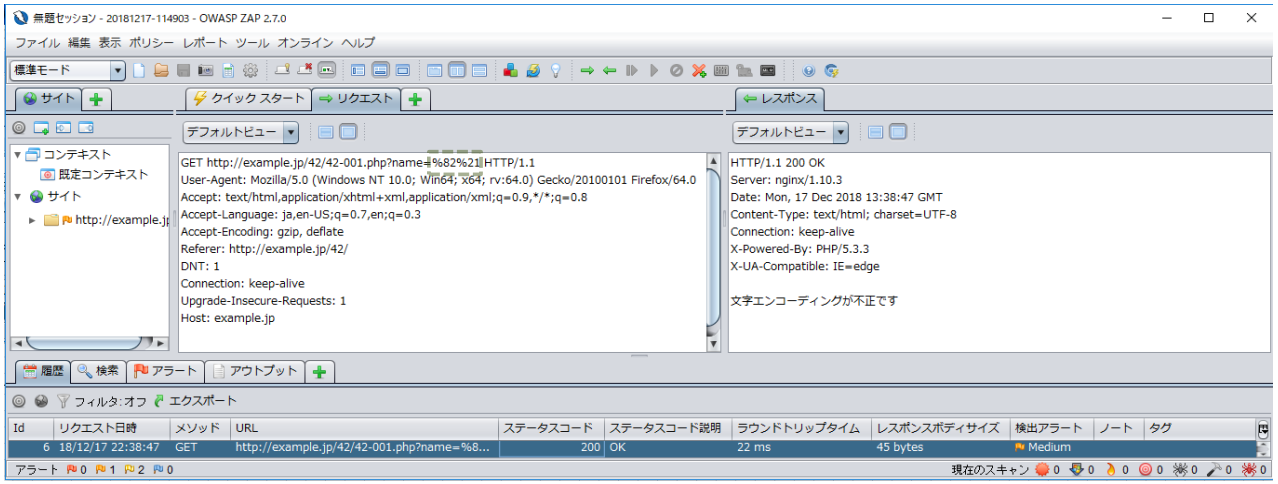
【サーバ： 42/42-001.php 】



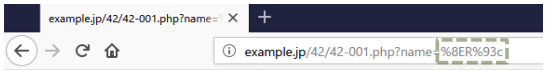
【ブラウザ→サーバ： リクエスト 42/42-001.php → レスポンス 】(正常系)



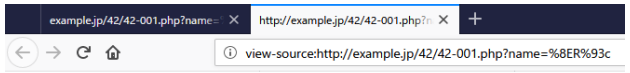
【ブラウザ→サーバ： リクエスト 42/42-001.php → レスポンス 】(不正な文字エンコーディング)



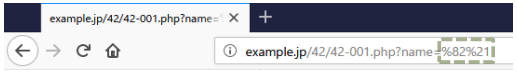
【ブラウザ】



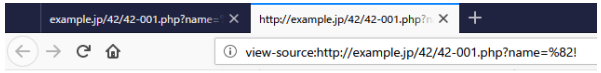
名前は山田です




```
1 <body>
2 名前は山田です
3 </body>
4
```



文字エンコーディングが不正です

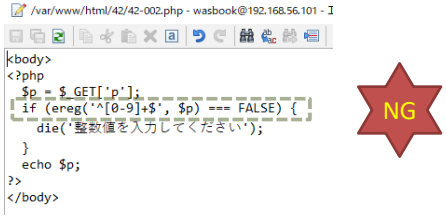



```
1 文字エンコーディングが不正です
```

42-002:ヌルバイト攻撃（正常系）/（XSS）

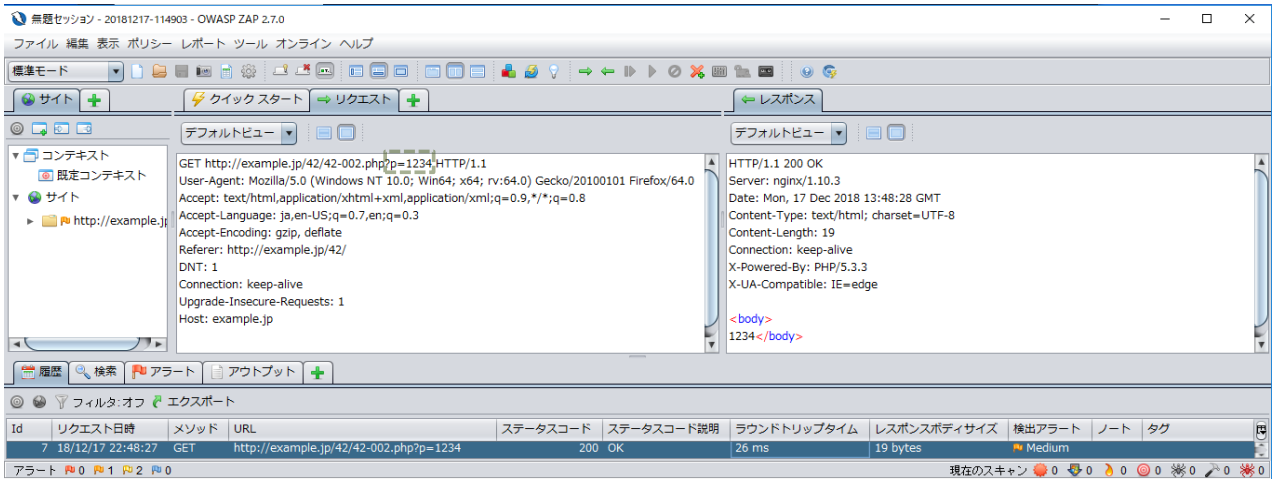
【ブラウザ】

【サーバ： 42/42-002.php 】

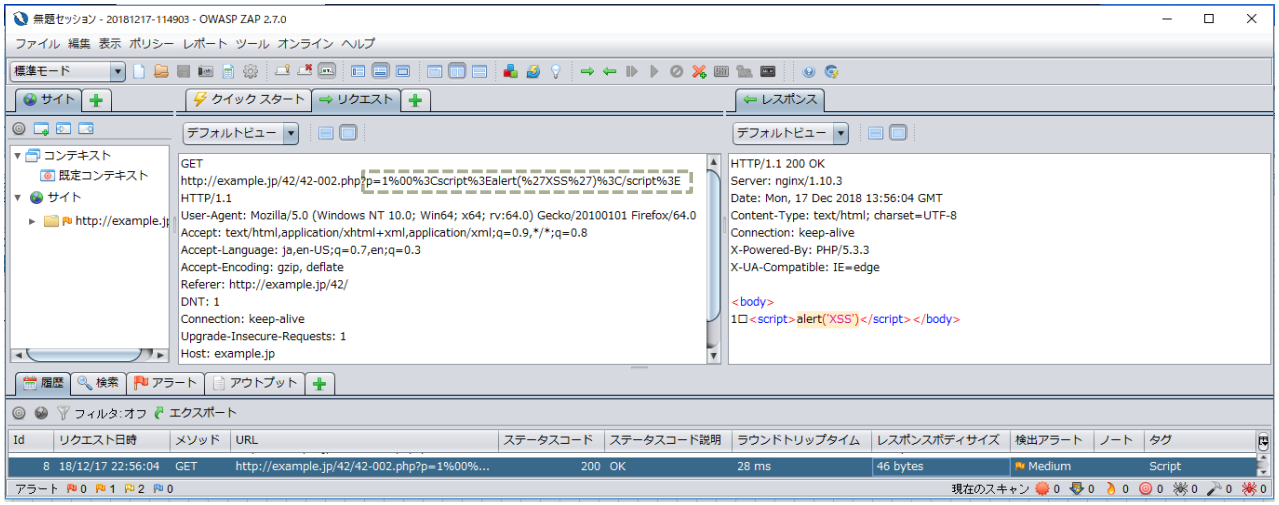


NG

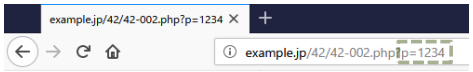
【ブラウザ→サーバ： リクエスト 42/42-002.php → レスポンス 】(正常系)

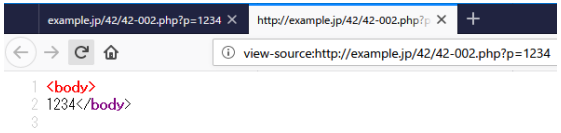


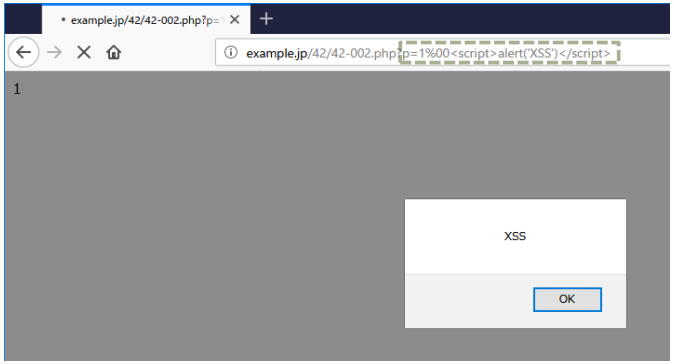
【ブラウザ→サーバ： リクエスト 42/42-002.php → レスポンス 】(XSS)



【ブラウザ】


example.jp/42/42-002.php?p=1234
example.jp/42/42-002.php?p=1234
1234


example.jp/42/42-002.php?p=1234
http://example.jp/42/42-002.php?p
view-source:http://example.jp/42/42-002.php?p=1234
<body>
1234</body>

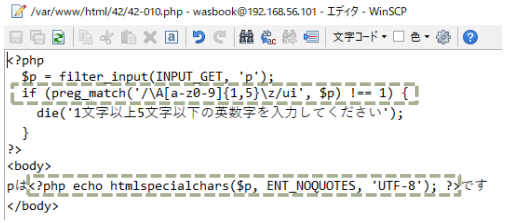

example.jp/42/42-002.php?p=
example.jp/42/42-002.php?p=1%00<script>alert(XSS)</script>
1
XSS
OK

NG

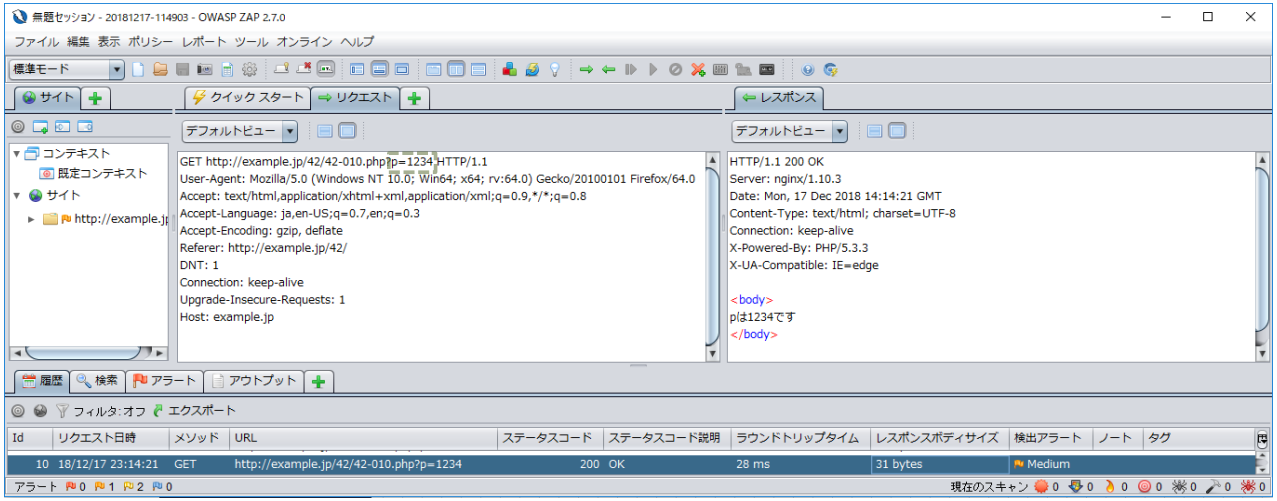
42-010:正規表現による入力値検証の例(1)(preg_match)英数字1文字～5文字(正常系)/(改行込み)

【ブラウザ】

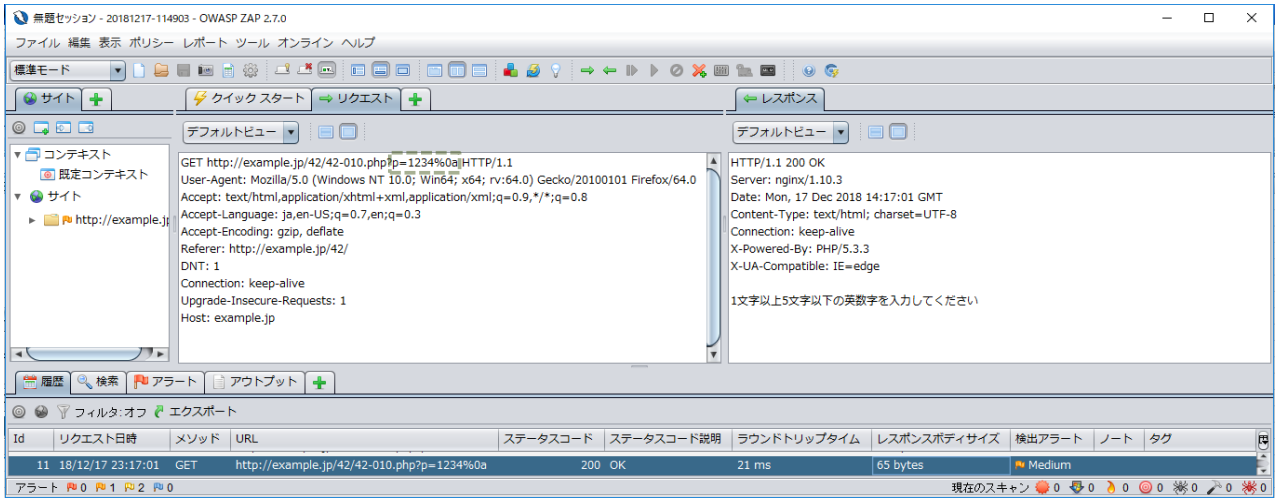
【サーバ： 42/42-010.php 】



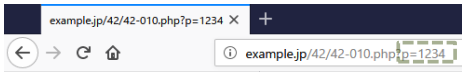
【ブラウザ→サーバ： リクエスト 42/42-010.php → レスポンス 】(正常系)

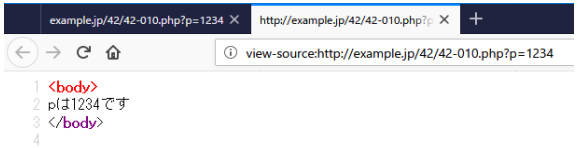


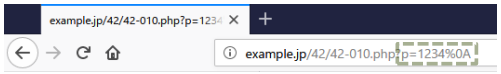
【ブラウザ→サーバ： リクエスト 42/42-010.php → レスポンス 】(改行込み)

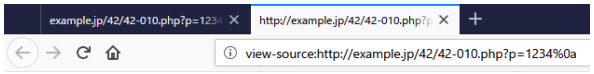


【ブラウザ】


example.jp/42/42-010.php?p=1234
example.jp/42/42-010.php?p=1234
pは1234です


example.jp/42/42-010.php?p=1234
http://example.jp/42/42-010.php?p
view-source:http://example.jp/42/42-010.php?p=1234
<body>
pは1234です
</body>


example.jp/42/42-010.php?p=1234
example.jp/42/42-010.php?p=1234%0A
1文字以上5文字以下の英数字を入力してください


example.jp/42/42-010.php?p=1234
http://example.jp/42/42-010.php?p
view-source:http://example.jp/42/42-010.php?p=1234%0a
1文字以上5文字以下の英数字を入力してください

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】

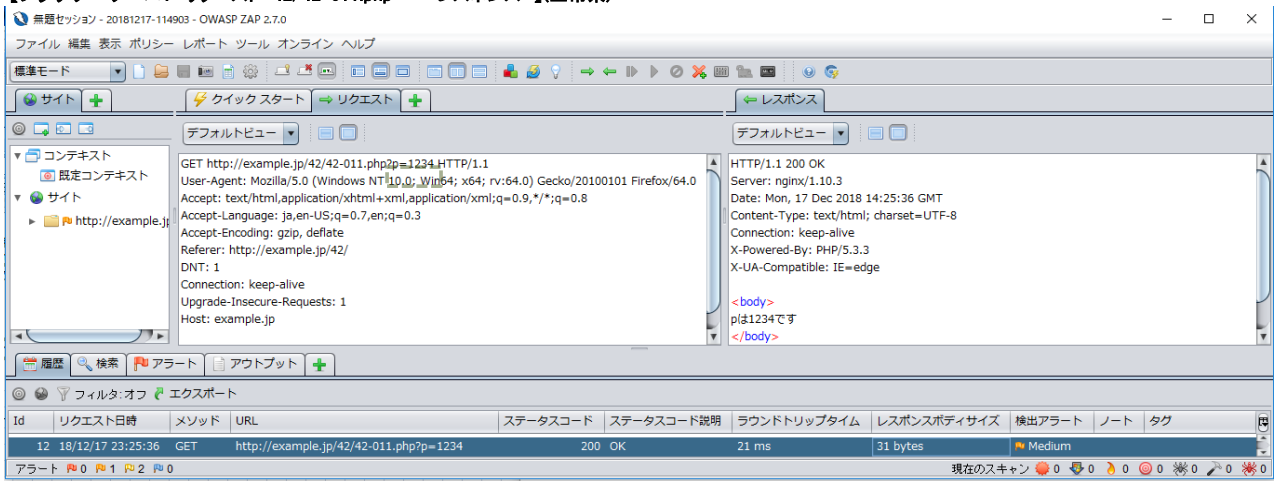
【サーバ： 42/42-011.php 】

/var/www/html/42/42-011.php - wasbook@192.168.56.101 - エディタ - WinSCP

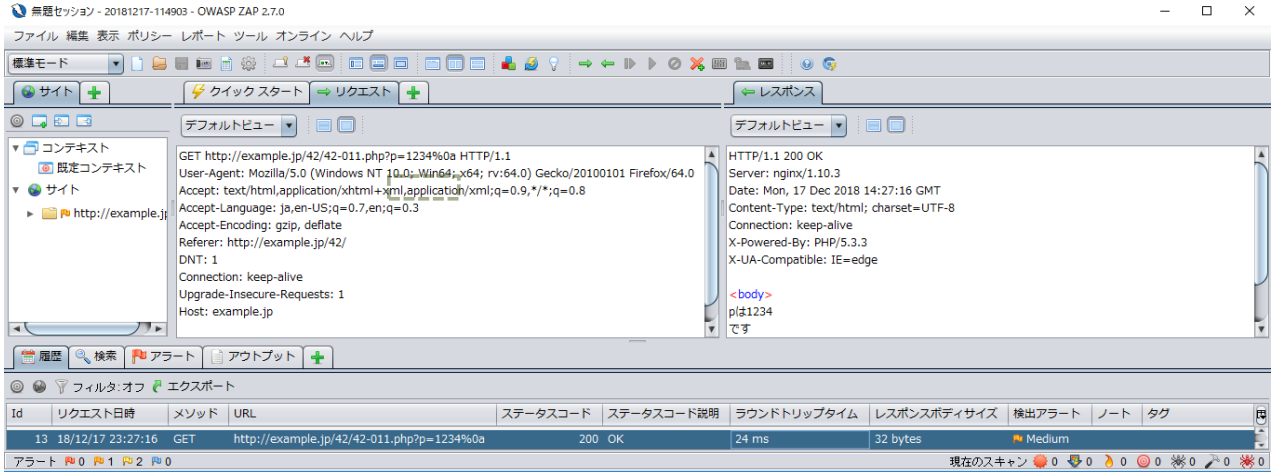
```
<?php
  $p = filter_input(INPUT_GET, 'p');
  if (preg_match('/^[a-z0-9]{1,5}$/ui', $p) !== 1) {
    die('1文字以上5文字以下の英数字を入力してください');
  }
?>
<body>
pは<?php echo htmlspecialchars($p, ENT_NOQUOTES, "UTF-8"); ?>です
</body>
```

注

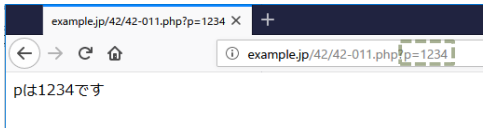
【ブラウザ→サーバ： リクエスト 42/42-011.php → レスポンス 】(正常系)

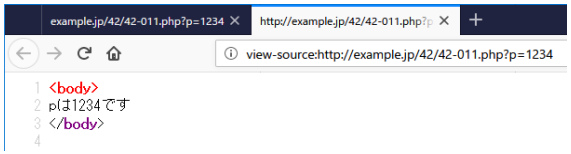


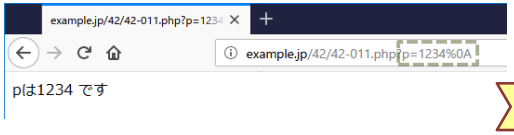
【ブラウザ→サーバ： リクエスト 42/42-011.php → レスポンス 】(改行込み)



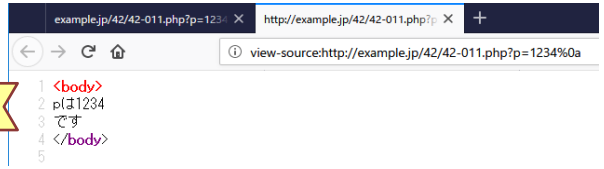
【ブラウザ】


example.jp/42/42-011.php?p=1234
example.jp/42/42-011.php?p=1234
pは1234です


example.jp/42/42-011.php?p=1234
http://example.jp/42/42-011.php?p
view-source:http://example.jp/42/42-011.php?p=1234
<body>
pは1234です
</body>


example.jp/42/42-011.php?p=1234
example.jp/42/42-011.php?p=1234%0A
pは1234 です

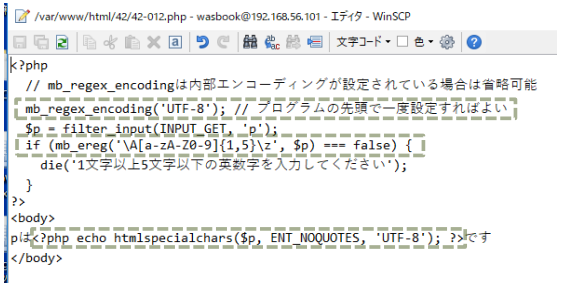
注


example.jp/42/42-011.php?p=1234
http://example.jp/42/42-011.php?p
view-source:http://example.jp/42/42-011.php?p=1234%0a
<body>
pは1234
です
</body>

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】

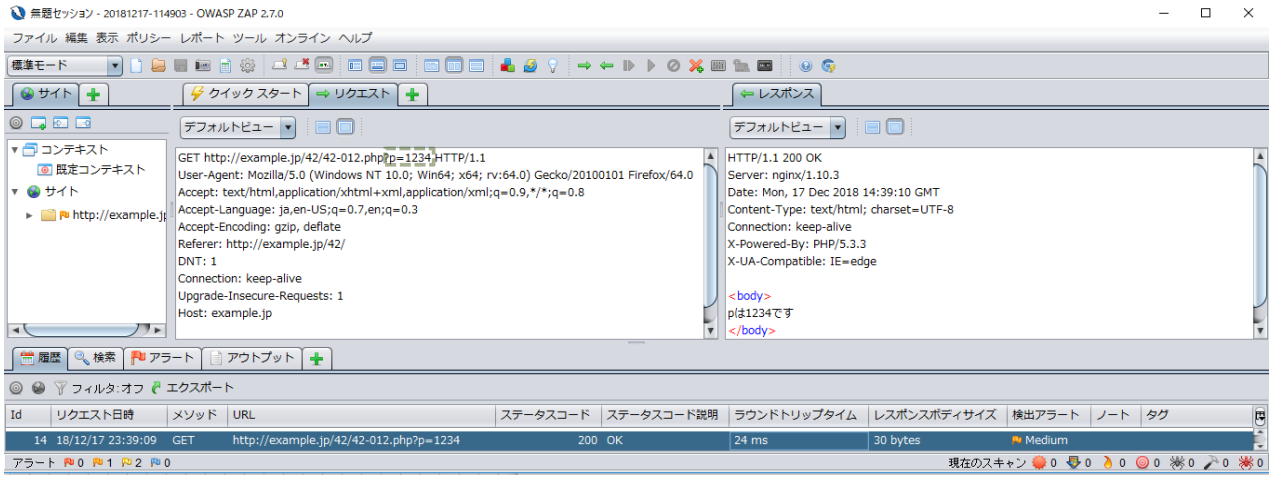
【サーバ： 42/42-012.php 】



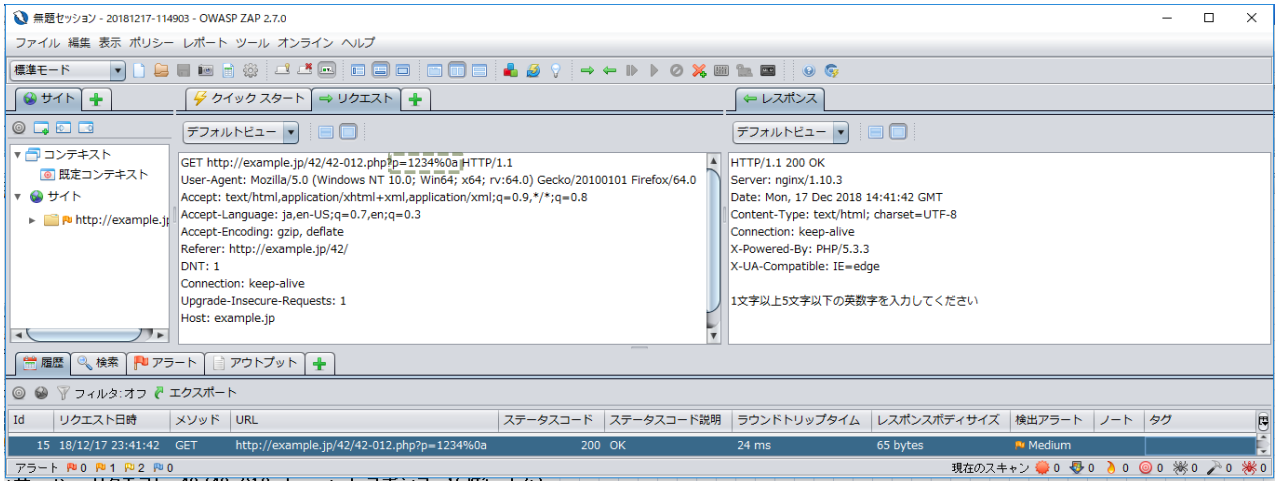
mb_ereg は戻り値として、整数型か論理値を返すので、型を区別する比較演算子「＝＝＝」を用いる

※ ＼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。

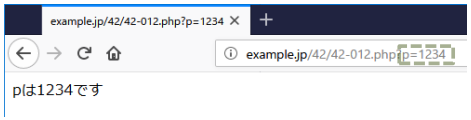
【ブラウザ→サーバ： リクエスト 42/42-012.php → レスポンス 】(正常系)

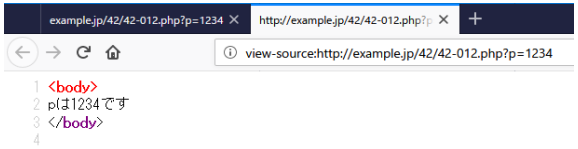


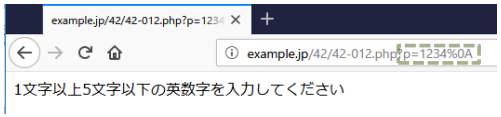
【ブラウザ→サーバ： リクエスト 42/42-012.php → レスポンス 】(改行込み)

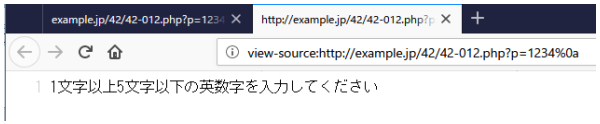


【ブラウザ】


example.jp/42/42-012.php?p=1234
example.jp/42/42-012.php?p=1234
pは1234です


example.jp/42/42-012.php?p=1234
http://example.jp/42/42-012.php?p
view-source:http://example.jp/42/42-012.php?p=1234
<body>
pは1234です
</body>

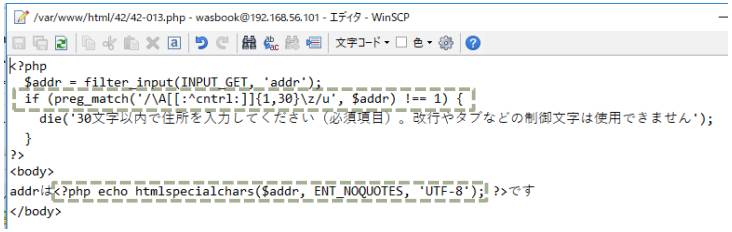

example.jp/42/42-012.php?p=1234
example.jp/42/42-012.php?p=1234%0A
1文字以上5文字以下の英数字を入力してください


example.jp/42/42-012.php?p=1234
http://example.jp/42/42-012.php?p
view-source:http://example.jp/42/42-012.php?p=1234%0a
1文字以上5文字以下の英数字を入力してください

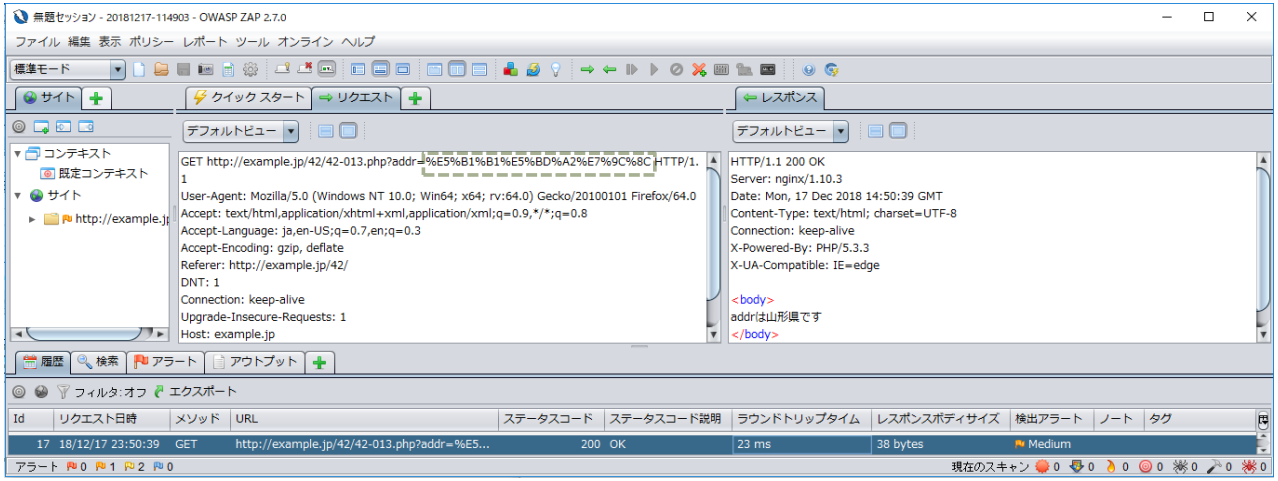
42-013:正規表現による入力値検証の例(2)住所欄(preg_match)(正常系)/(ヌルバイトを含む)

【ブラウザ】

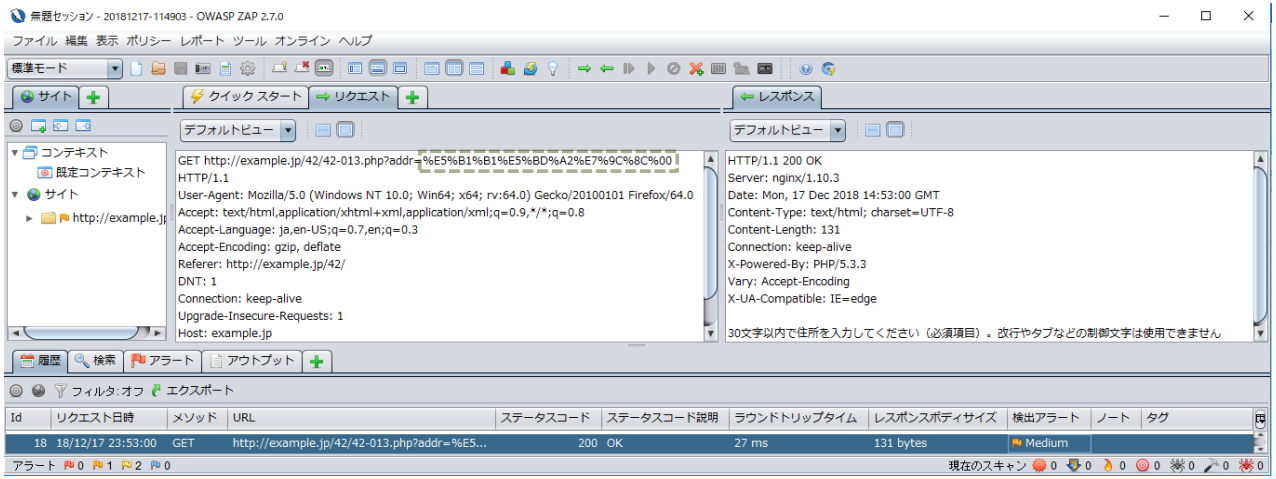
【サーバ： 42/42-013.php 】



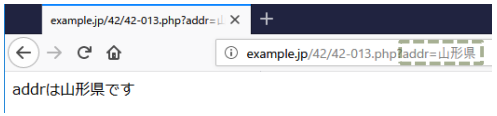
【ブラウザ→サーバ： リクエスト 42/42-013.php → レスポンス 】(正常系)

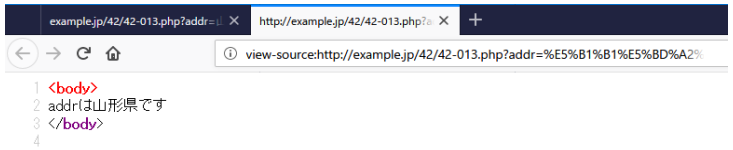


【ブラウザ→サーバ： リクエスト 42/42-013.php → レスポンス 】(ヌルバイトを含む)

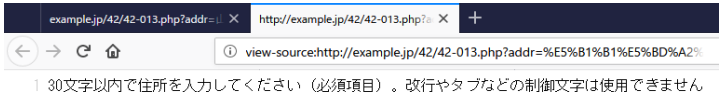


【ブラウザ】


example.jp/42/42-013.php?addr=山形県
addrは山形県です


view-source:http://example.jp/42/42-013.php?addr=%E5%B1%B1%E5%BD%A2%
<body>
addrは山形県です
</body>

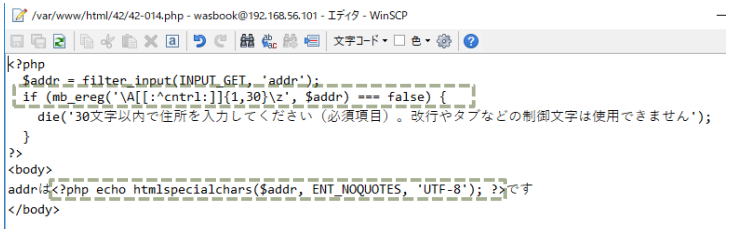

example.jp/42/42-013.php?addr=山形県%00
30文字以内で住所を入力してください（必須項目）。改行やタブなどの制御文字は使用できません


view-source:http://example.jp/42/42-013.php?addr=%E5%B1%B1%E5%BD%A2%
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

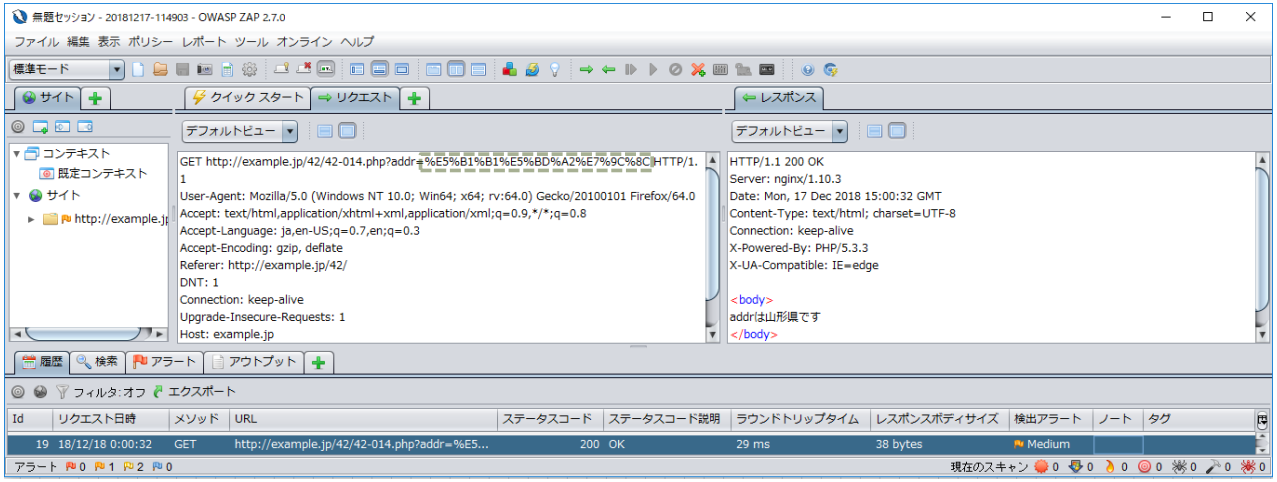
42-014:正規表現による入力値検証の例(2)住所欄(mb_ereg)(正常系)/(ヌルバイトを含む)

【ブラウザ】

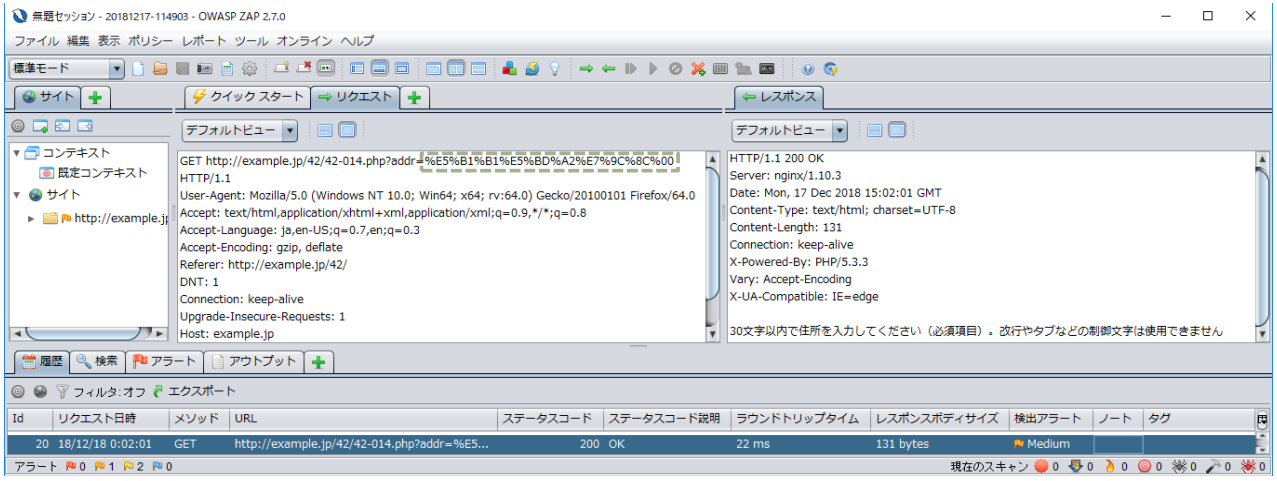
【サーバ： 42/42-014.php 】



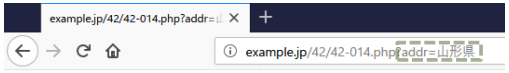
【ブラウザ→サーバ： リクエスト 42/42-014.php → レスポンス 】(正常系)

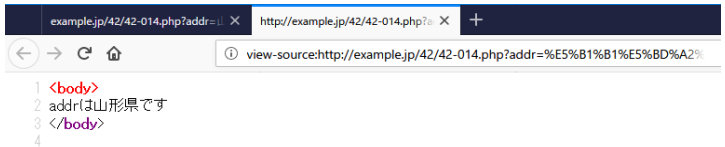


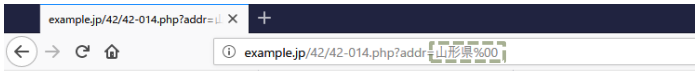
【ブラウザ→サーバ： リクエスト 42/42-014.php → レスポンス 】(ヌルバイトを含む)

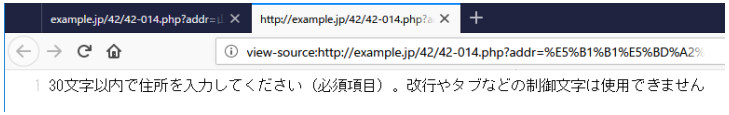


【ブラウザ】


example.jp/42/42-014.php?addr=山形県
addrは山形県です


view-source:http://example.jp/42/42-014.php?addr=%E5%B1%B1%E5%BD%A2%
<body>
addrは山形県です
</body>

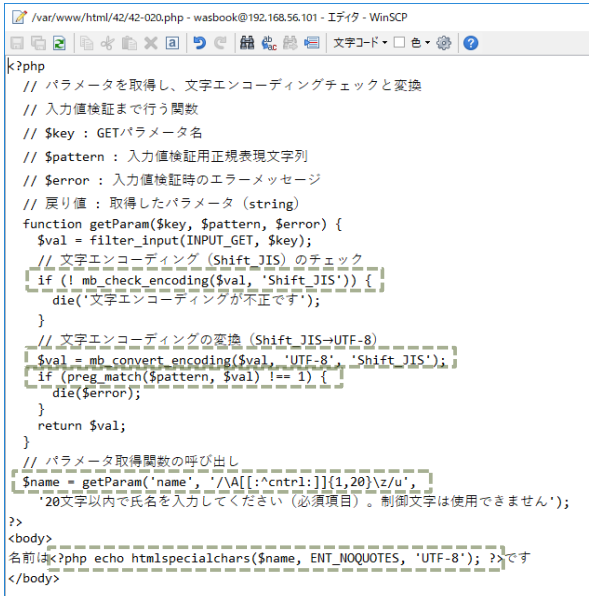

example.jp/42/42-014.php?addr=山形県%00
30文字以内で住所を入力してください（必須項目）。改行やタブなどの制御文字は使用できません


view-source:http://example.jp/42/42-014.php?addr=%E5%B1%B1%E5%BD%A2%
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

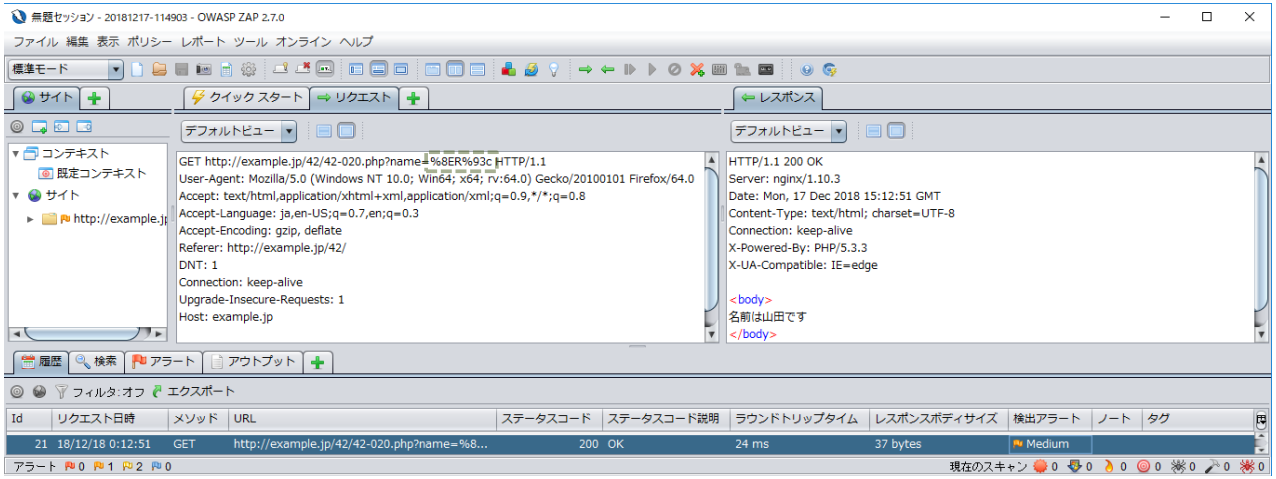
42-020:4.2節のまとめ (改行込み)/(ヌルバイト込み)

【ブラウザ】

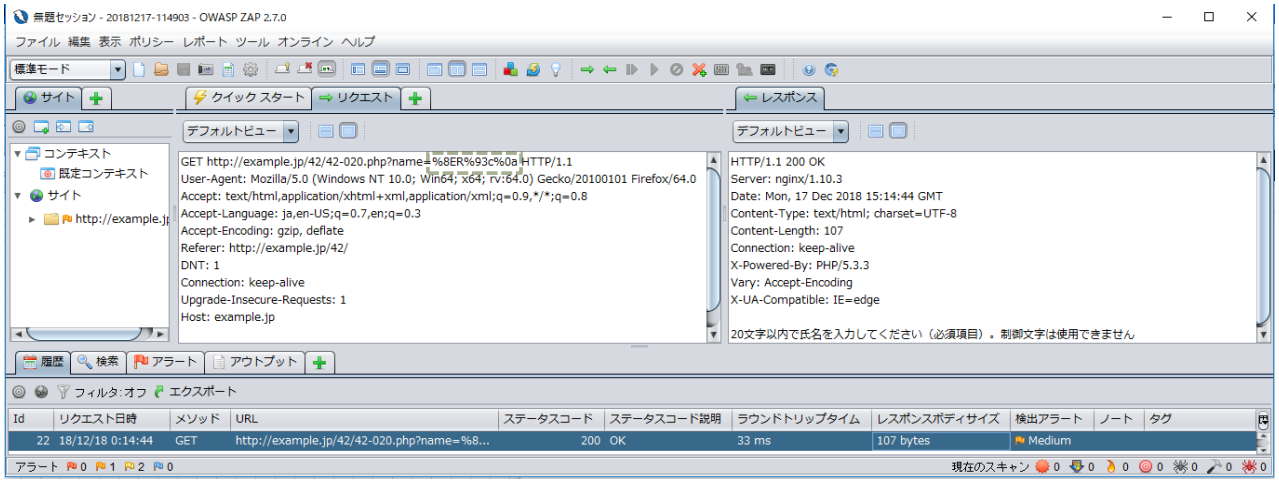
【サーバ： 42/42-020.php 】



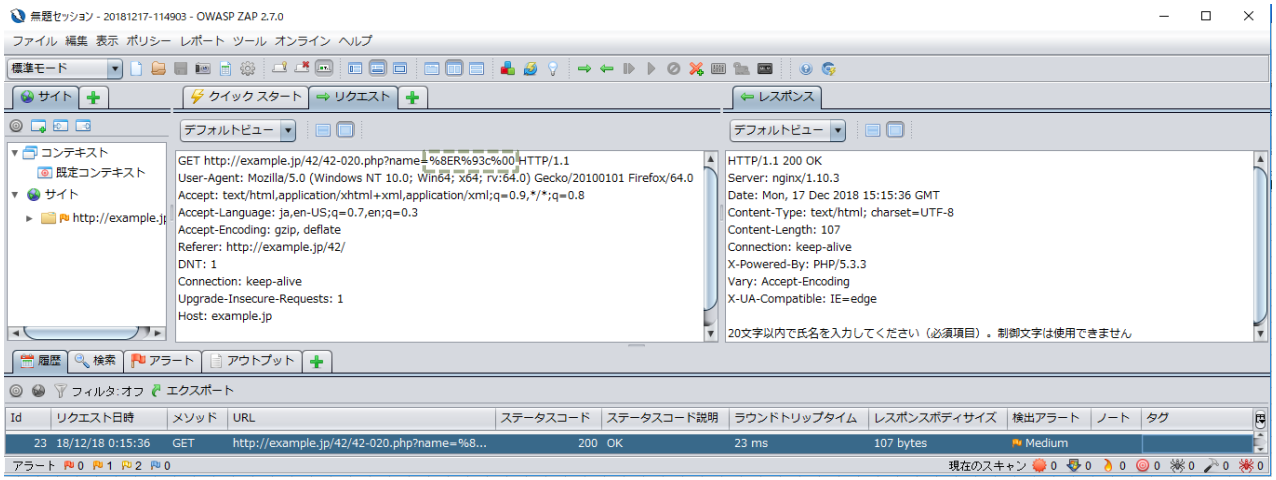
【ブラウザ→サーバ： リクエスト 42/42-020.php → レスポンス 】(正常系)



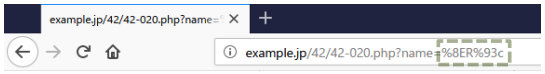
【ブラウザ→サーバ： リクエスト 42/42-020.php → レスポンス 】(改行込み)



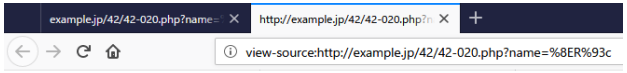
【ブラウザ→サーバ： リクエスト 42/42-020.php → レスポンス 】(ヌルバイト込み)



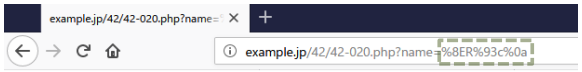
【ブラウザ】



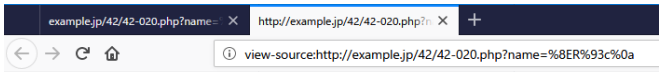
名前は山田です



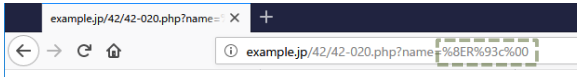

```
1 <body>
2 名前は山田です
3 </body>
4
```



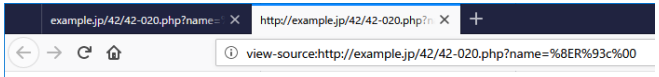
20文字以内で氏名を入力してください（必須項目）。制御文字は使用できません



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



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



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