4b-001d:お問い合わせフォーム(パラメータをエスケープ)

【ブラウザ】

【サーバ: 4b/4b-001d.html 】

/var/www/html/4b/4b-001d.html - wasbook@example.jp - エディタ

```
<body>
<form action="4b-002d.php" method="POST">
お問い合わせをどうぞ<br>
メールアドレス
<input name="mail"><br>
お問い合わせ
<textarea name="inqu" cols="20" rows="3">
</textarea><br>
<input type="submit" value="送信">
</form>
</body>
```

【サーバ: 4b/4b-002d.php 】

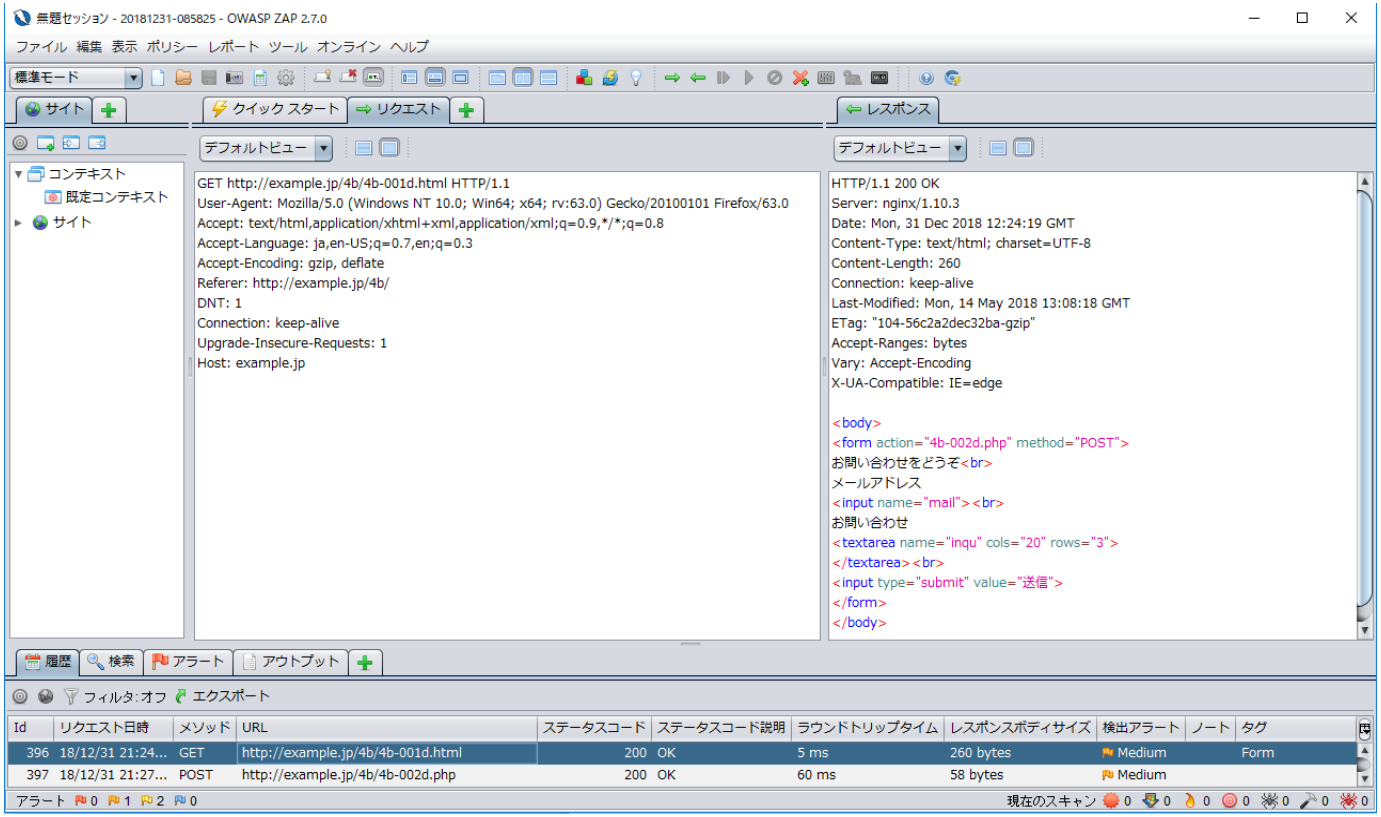
/var/www/html/4b/4b-002d.php - wasbook@example.jp - エディタ - WinSCP

```
<?php
  $mail = $_POST['mail'];
  system('/usr/sbin/sendmail <template.txt ' . escapeshellarg($mail));
?>
<body>
お問い合わせを受け付けました
</body>
```

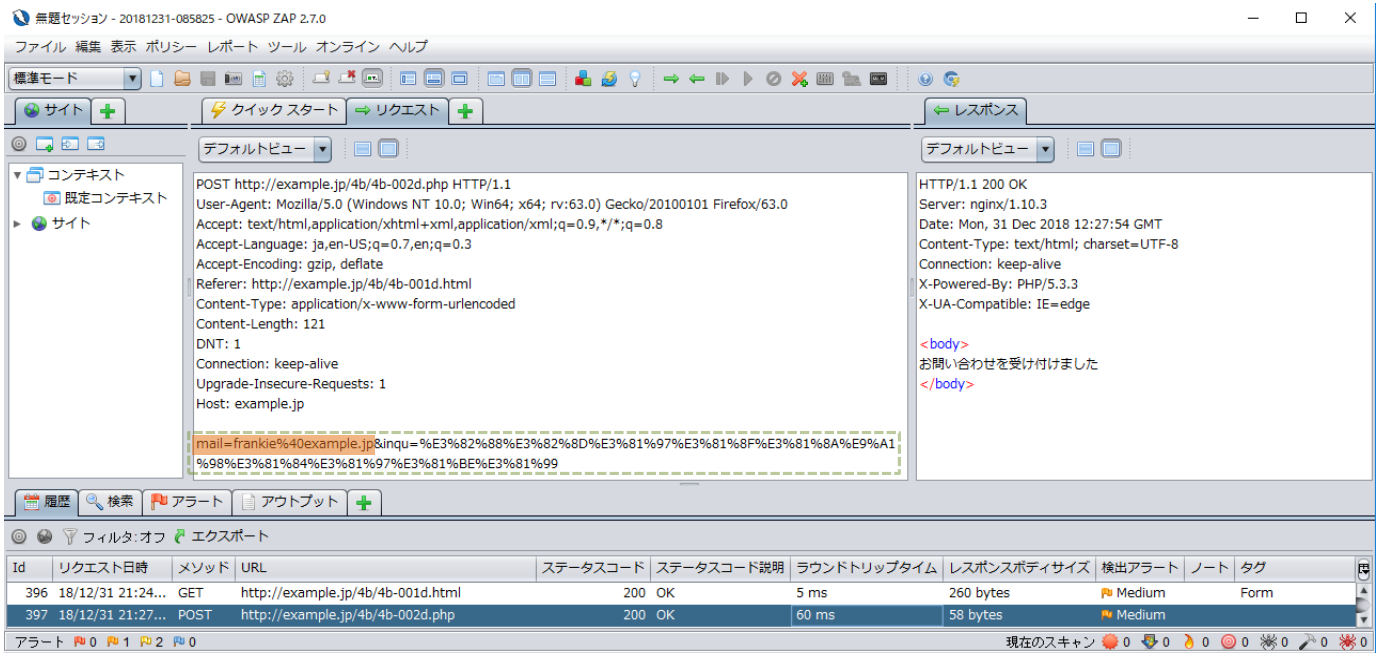
パラメータをエスケープした、OSコマンドインジェクション対策版

OSコマンド・インジェクション脆弱性は対策しましたが、シェルのエスケープルールの複雑性の問題から、escapeshellargを使っていても脆弱性混入の可能性はあります。

【ブラウザ→サーバ： リクエスト 4b/4b-001d.html → レスポンス 】



【ブラウザ→サーバ： リクエスト 4b/4b-002d.php → レスポンス 】

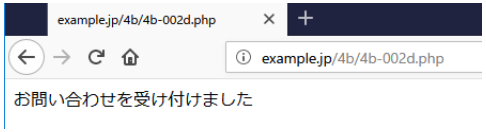


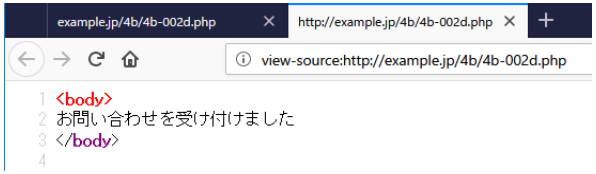
【ブラウザ】


example.jp/4b/4b-001d.html
example.jp/4b/4b-001d.html
お問い合わせをどうぞ
メールアドレス
お問い合わせ
送信


example.jp/4b/4b-001d.html
http://example.jp/4b/4b-001d.html
view-source:http://example.jp/4b/4b-001d.html
<body>
<form action="4b-002d.php" method="POST">
お問い合わせをどうぞ
メールアドレス
<input name="mail">
お問い合わせ
<textarea name="inqu" cols="20" rows="3">
</textarea>
<input type="submit" value="送信">
</form>
</body>


example.jp/4b/4b-001d.html
example.jp/4b/4b-001d.html
お問い合わせをどうぞ
メールアドレス
frankie@example.jp
よろしくお願いします
お問い合わせ
送信


example.jp/4b/4b-002d.php
example.jp/4b/4b-002d.php
お問い合わせを受け付けました


example.jp/4b/4b-002d.php
http://example.jp/4b/4b-002d.php
view-source:http://example.jp/4b/4b-002d.php
<body>
お問い合わせを受け付けました
</body>

4b-001d:お問い合わせフォーム(パラメータをエスケープ)

【ブラウザ】

【サーバ： 4b/4b-001dx.html 】

/var/www/html/4b/4b-001dx.html - wasbook@example.jp - エディタ - WinSCP

```
<body>
<form action="4b-002d.php" method="POST">
お問い合わせをどうぞ<br>
メールアドレス
<input name="mail" value="bob@example.jp;cat /etc/passwd"><br>
お問い合わせ
<textarea name="inqu" cols="20" rows="3">
</textarea><br>
<input type="submit" value="送信">
</form>
</body>
```

【サーバ： 4b/4b-002d.php 】

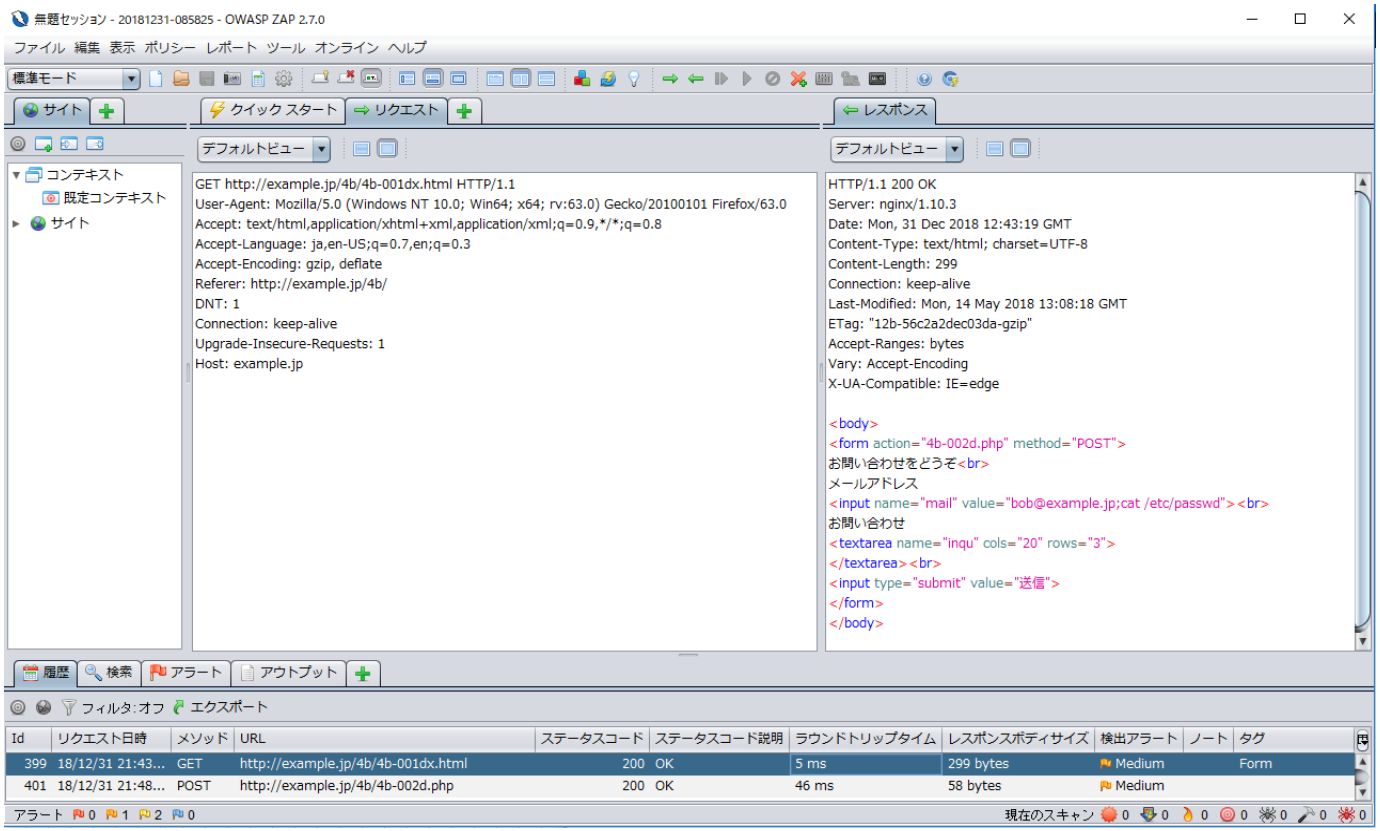
/var/www/html/4b/4b-002d.php - wasbook@example.jp - エディタ - WinSCP

```
<?php
  $mail = $_POST['mail'];
  system('/usr/sbin/sendmail <template.txt ' . escapeshellarg($mail));
?>
<body>
お問い合わせを受け付けました
</body>
```

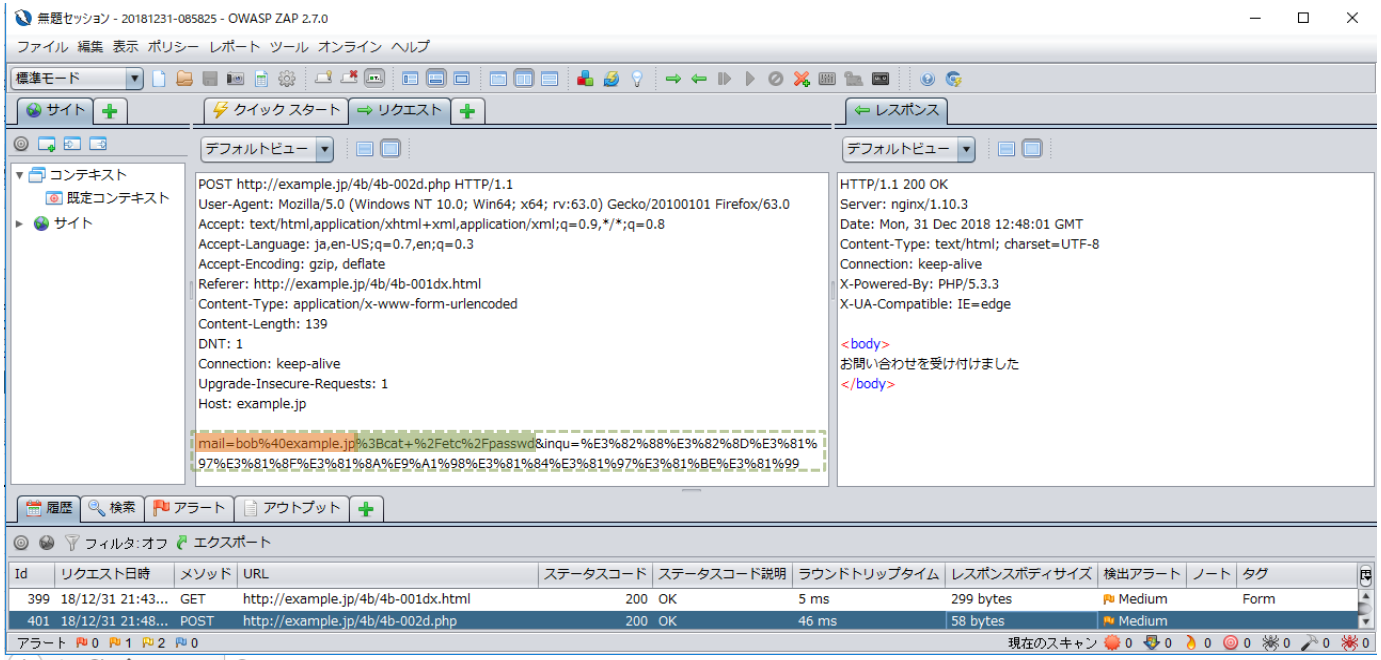
パラメータをエスケープした、OSコマンドインジェクション対策版

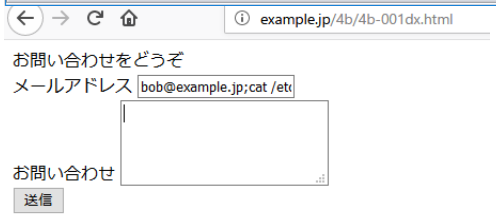
OSコマンド・インジェクション脆弱性は対策しましたが、シェルのエスケープルールの複雑性の問題から、escapeshellargを使っていても脆弱性混入の可能性はあります。

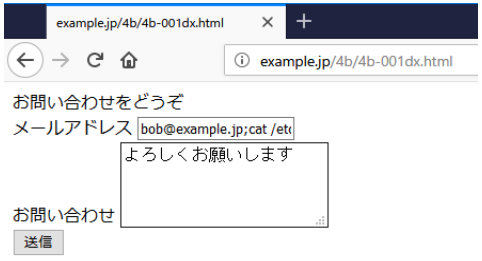
【ブラウザ→サーバ： リクエスト 4b/4b-001dx.html → レスポンス 】

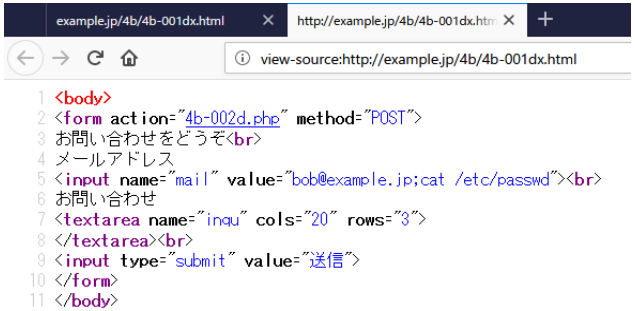


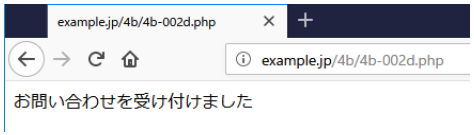
【ブラウザ→サーバ：リクエスト 4b/4b-002d.php → レスポンス 】

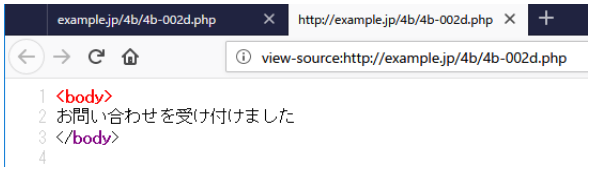






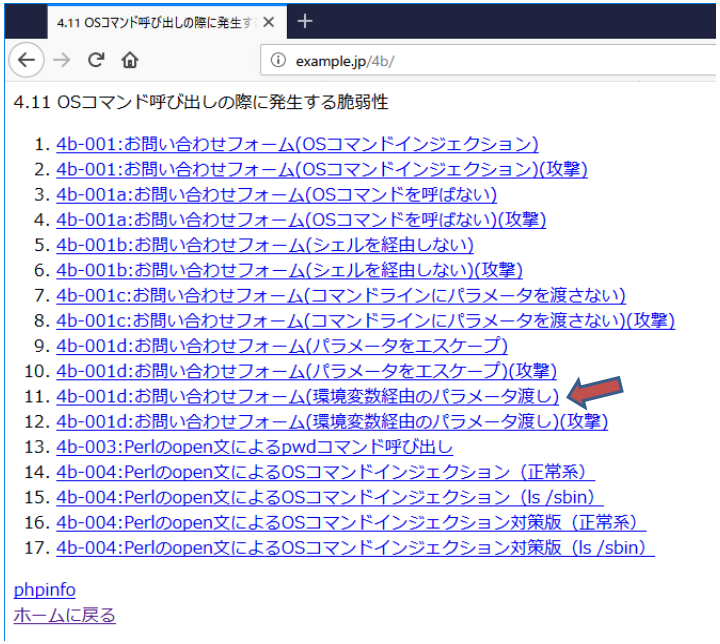



example.jp/4b/4b-002d.php
example.jp/4b/4b-002d.php
お問い合わせを受け付けました

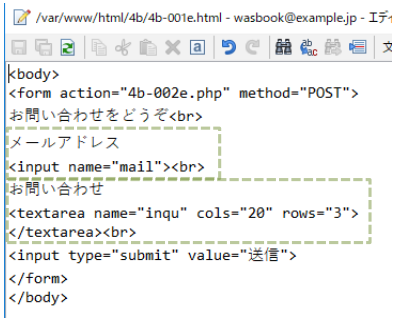

example.jp/4b/4b-002d.php
http://example.jp/4b/4b-002d.php
view-source:http://example.jp/4b/4b-002d.php
<body>
お問い合わせを受け付けました
</body>

4b-001d:お問い合わせフォーム(環境変数経由のパラメータ渡し)

【ブラウザ】



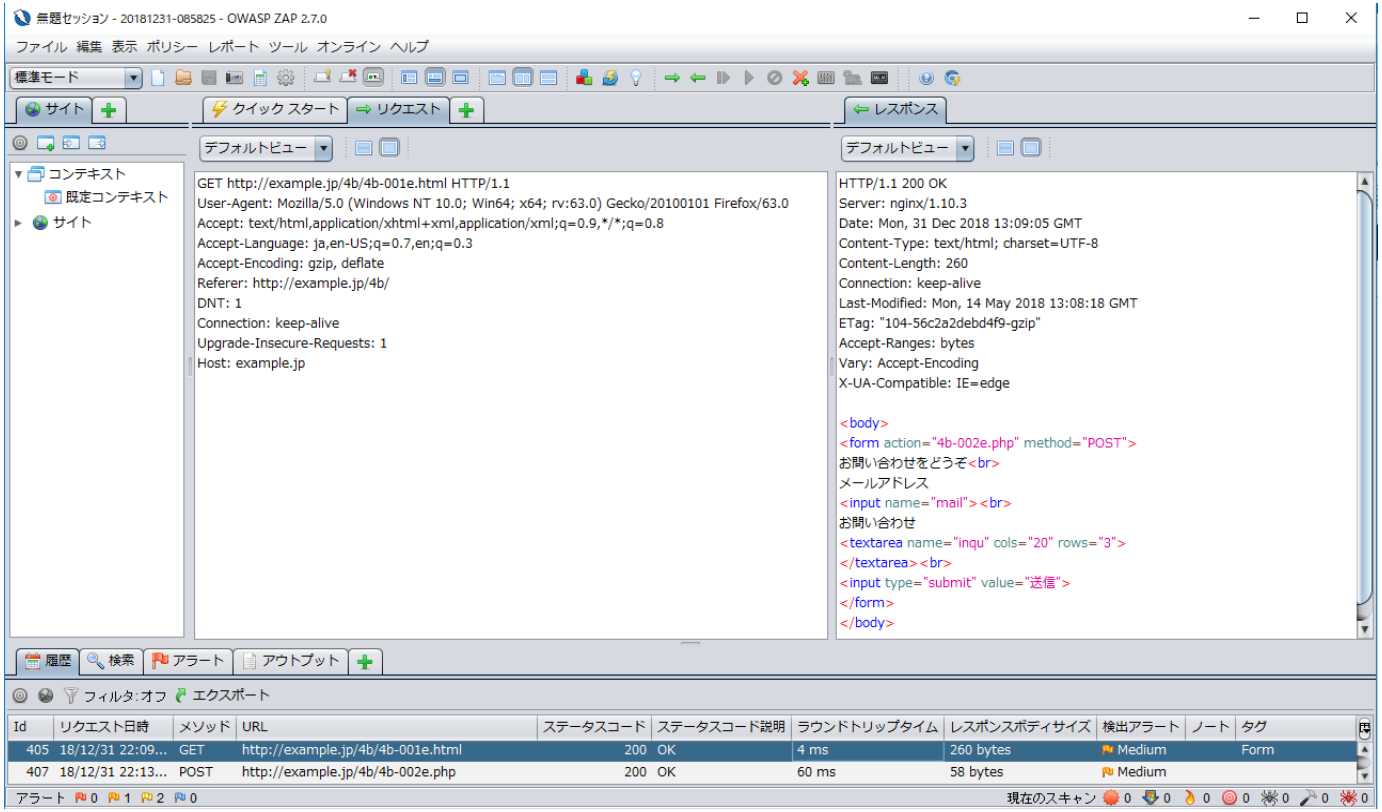
【サーバ: 4b/4b-001e.html 】



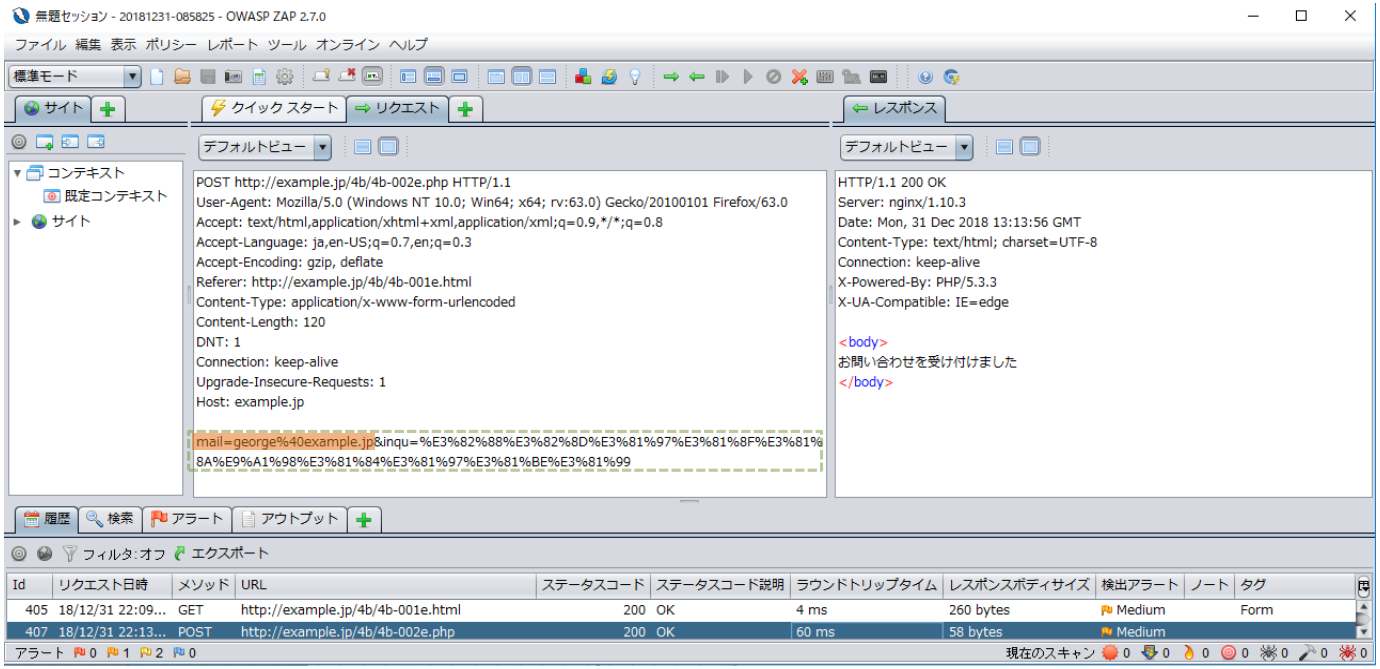
【サーバ: 4b/4b-002e.php 】

環境変数経由のパラメータ渡しのOSコマンドインジェクション対策版

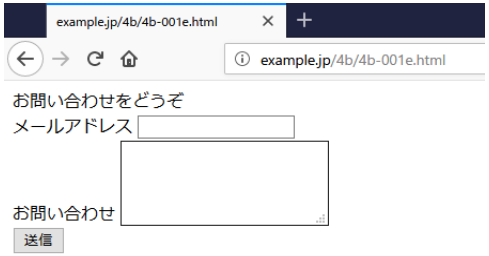
【ブラウザ→サーバ：リクエスト 4b/4b-001e.html → レスポンス 】

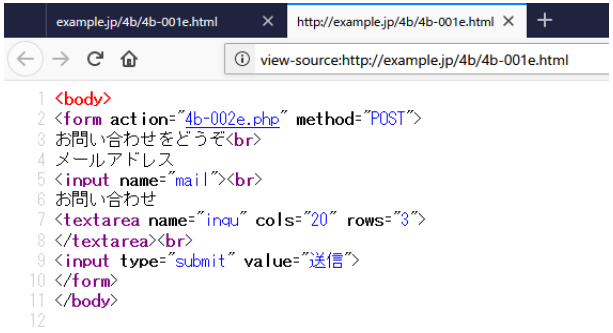


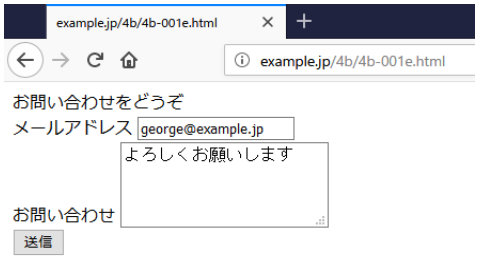
【ブラウザ→サーバ： リクエスト 4b/4b-002e.php → レスポンス 】

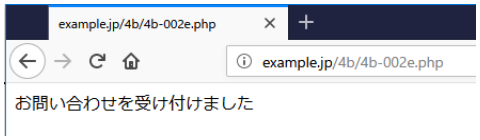


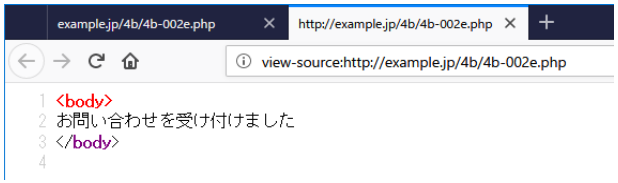
【ブラウザ】


example.jp/4b/4b-001e.html
example.jp/4b/4b-001e.html
お問い合わせをどうぞ
メールアドレス
お問い合わせ
送信


example.jp/4b/4b-001e.html
http://example.jp/4b/4b-001e.html
view-source:http://example.jp/4b/4b-001e.html
<body>
<form action="4b-002e.php" method="POST">
お問い合わせをどうぞ
メールアドレス
<input name="mail">
お問い合わせ
<textarea name="inqu" cols="20" rows="3">
</textarea>
<input type="submit" value="送信">
</form>
</body>


example.jp/4b/4b-001e.html
example.jp/4b/4b-001e.html
お問い合わせをどうぞ
メールアドレス
george@example.jp
お問い合わせ
よろしくお願いします
送信


example.jp/4b/4b-002e.php
example.jp/4b/4b-002e.php
お問い合わせを受け付けました


example.jp/4b/4b-002e.php
http://example.jp/4b/4b-002e.php
view-source:http://example.jp/4b/4b-002e.php
<body>
お問い合わせを受け付けました
</body>

4b-001d:お問い合わせフォーム(パラメータをエスケープ)(攻撃)

【ブラウザ】

【サーバ： 4b/4b-001ex.html 】

/var/www/html/4b/4b-001ex.html - wasbook@example.jp - エディタ - WinSCP

```
<body>
<form action="4b-002e.php" method="POST">
お問い合わせをどうぞ<br>
メールアドレス
<input name="mail" value="bob@example.jp;cat /etc/passwd"><br>
お問い合わせ
<textarea name="inqu" cols="20" rows="3">
</textarea><br>
<input type="submit" value="送信">
</form>
</body>
```

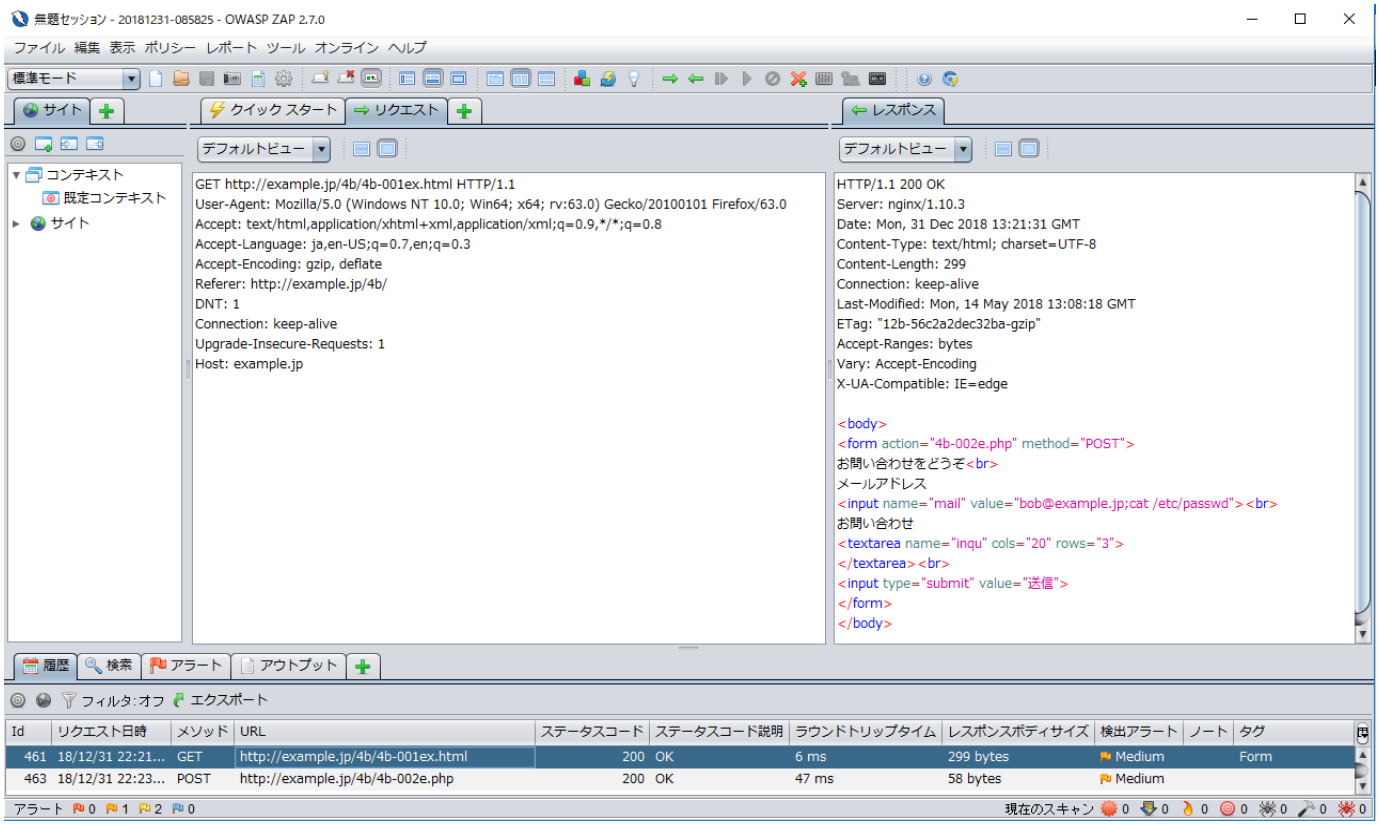
【サーバ： 4b/4b-002e.php 】

/var/www/html/4b/4b-002e.php - wasbook@example.jp - エディタ - WinSCP

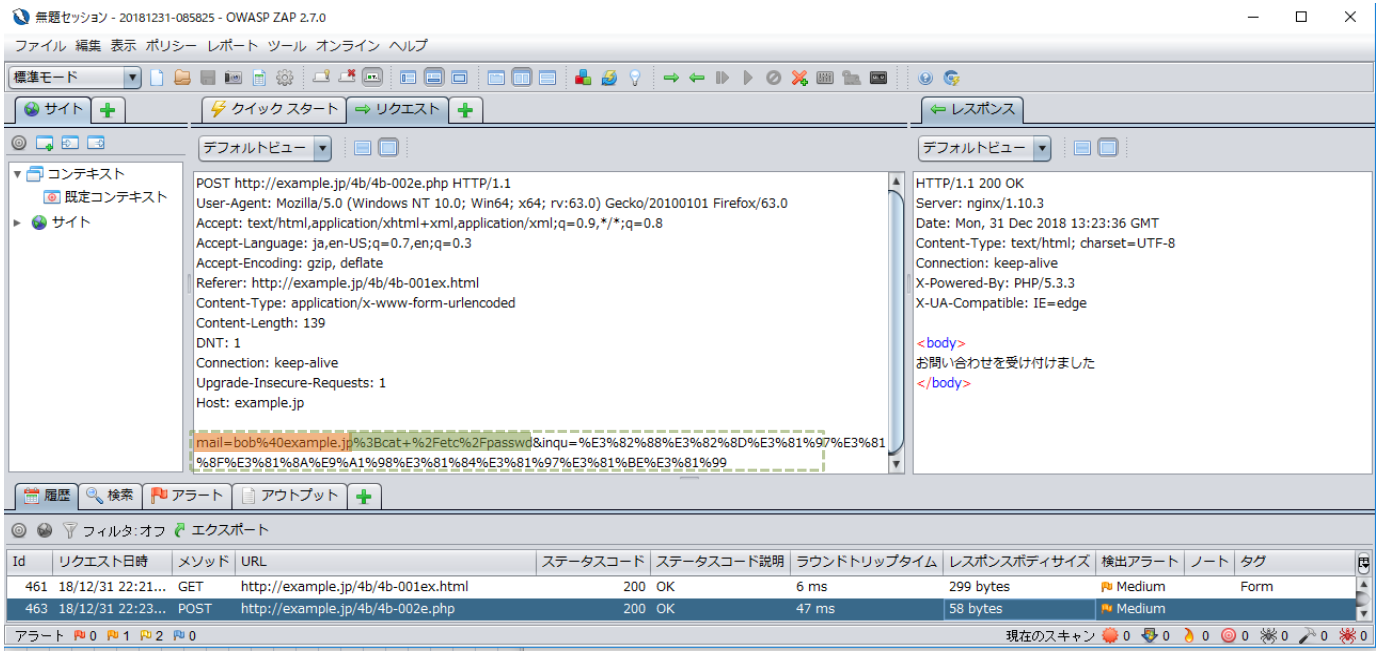
```
<?php
  $mail = filter_input(INPUT_POST, 'mail');

  $descriptorspec = array(0 => array("pipe", "r"));
  $env = array('e_mail' => $mail);

  $process = proc_open('/usr/sbin/sendmail -i "$e_mail"', $descriptorspec, $pipes, getcwd(), $env);

  if (is_resource($process)) {
    fwrite($pipes[0], file_get_contents('template.txt'));
    fclose($pipes[0]);
    proc_close($process);
  }
?>
<body>
お問い合わせを受け付けました
</body>
```

環境変数経由のパラメータ渡しのOSコマンドインジェクション対策版

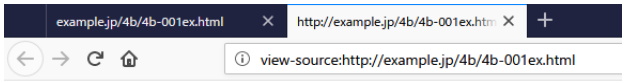
【ブラウザ→サーバ：リクエスト 4b/4b-001ex.html → レスポンス 】



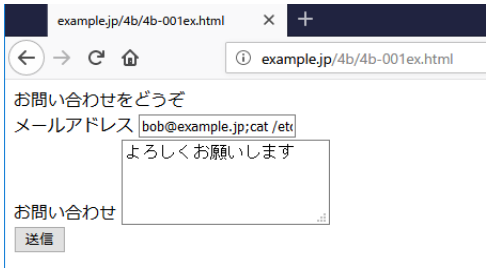
【ブラウザ→サーバ： リクエスト 4b/4b-001e2.php → レスポンス 】

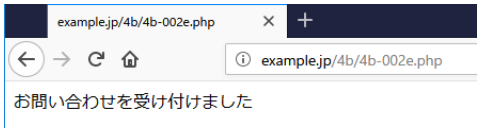


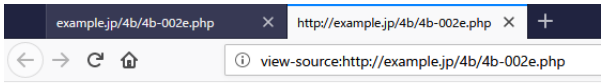
【ブラウザ】




```
<body>
<form action="4b-002e.php" method="POST">
お問い合わせをどうぞ<br>
メールアドレス
<input name="mail" value="bob@example.jp;cat /etc/passwd"><br>
お問い合わせ
<textarea name="inqu" cols="20" rows="3">
</textarea><br>
<input type="submit" value="送信">
</form>
</body>

```





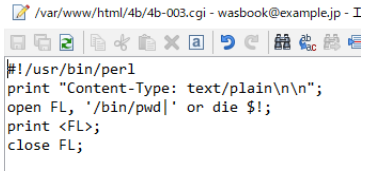



```
<body>
お問い合わせを受け付けました
</body>

```

4b-003:Perlのopen文によるpwdコマンド呼び出し

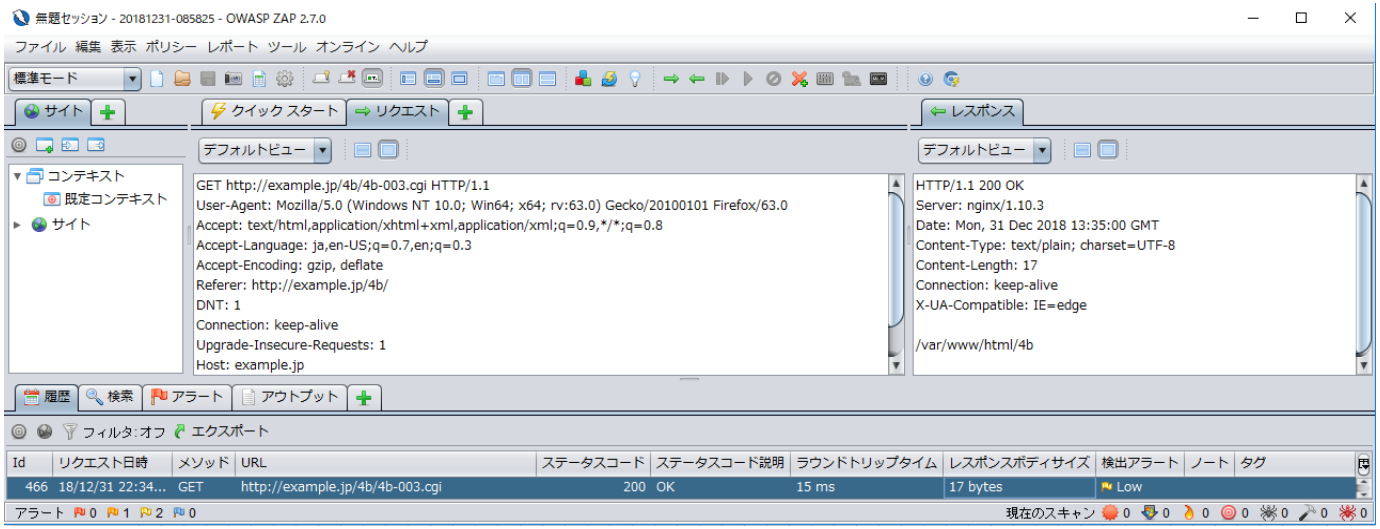
【ブラウザ】

【サーバ： 4b/4b-003.cgi 】

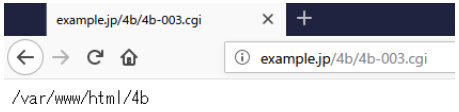


PerlのOPEN関数を用いて、OSコマンドを実行できます

【ブラウザ→サーバ：リクエスト 4b/4b-003.cgi → レスポンス 】



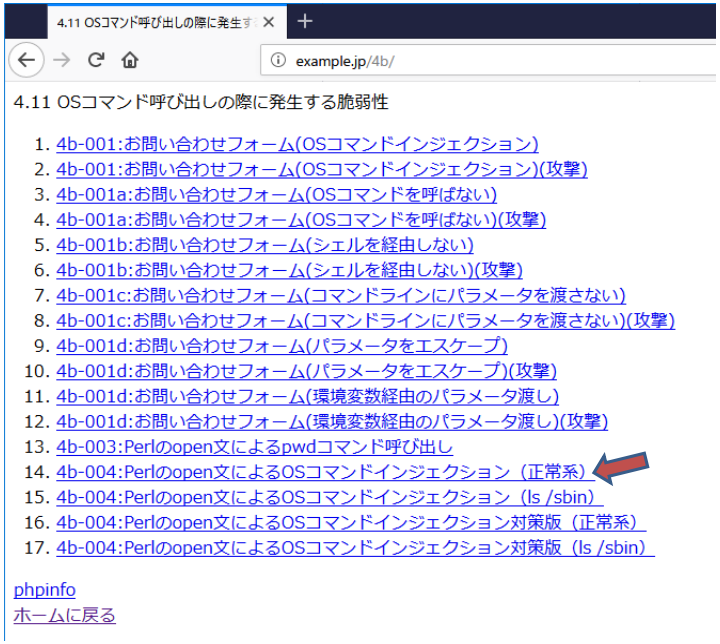
【ブラウザ】



/var/www/html/4b

4b-004:Perlのopen文によるOSコマンドインジェクション（正常系）

【ブラウザ】



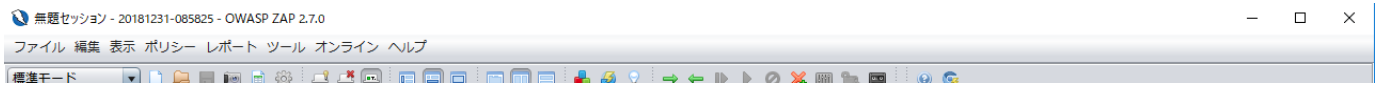
【サーバ： 4b/4b-004.cgi 】

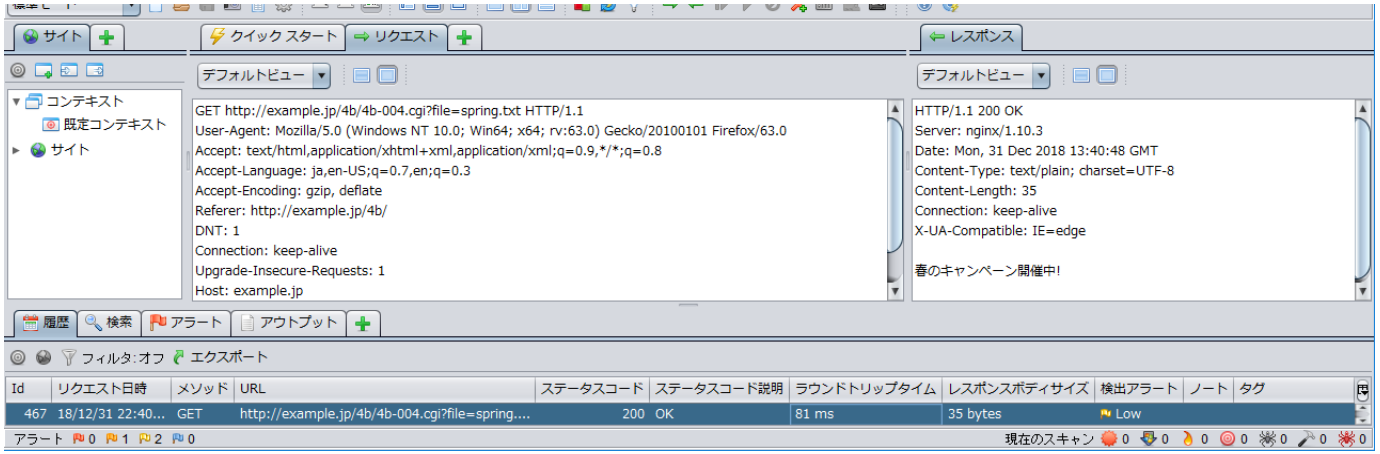
/var/www/html/4b/4b-004.cgi - wasbook@example.jp - エディタ - WinSCP

```
#!/usr/bin/perl
use strict;
use utf8;
use open ':utf8';  # デフォルトの文字コードをUTF-8に
use CGI;

print "Content-Type: text/plain; charset=UTF-8\r\n\r\n";

my $q = new CGI;
my $file = $q->param('file');
open IN, $file or die $!;  # ファイルを開く
print <IN>;        # ファイルの内容を全て表示
close IN;          # ファイルクローズ
```

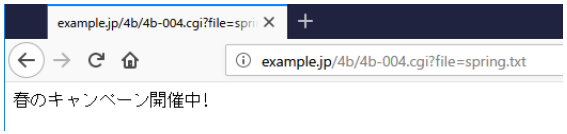
PerlのOPEN関数を用いて、ファイル名を外部から指定できる場合、
ファイル名の前後にパイプ記号「｜」を追加することで、
OSコマンド・インジェクション攻撃できる場合があります

【ブラウザ→サーバ： リクエスト 4b/4b-004.cgi → レスポンス 】




サイト
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
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
GET http://example.jp/4b/4b-004.cgi?file=spring.txt 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/4b/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 31 Dec 2018 13:40:48 GMT
Content-Type: text/plain; charset=UTF-8
Content-Length: 35
Connection: keep-alive
X-UA-Compatible: IE=edge
春のキャンペーン開催中!
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
467
18/12/31 22:40...
GET
http://example.jp/4b/4b-004.cgi?file=spring....
200
OK
81 ms
35 bytes
Low
アラート
現在のスキャン

【ブラウザ】


example.jp/4b/4b-004.cgi?file=spri
example.jp/4b/4b-004.cgi?file=spring.txt
春のキャンペーン開催中!

4b-004:Perlのopen文によるOSコマンドインジェクション（ls /sbin）

【ブラウザ】

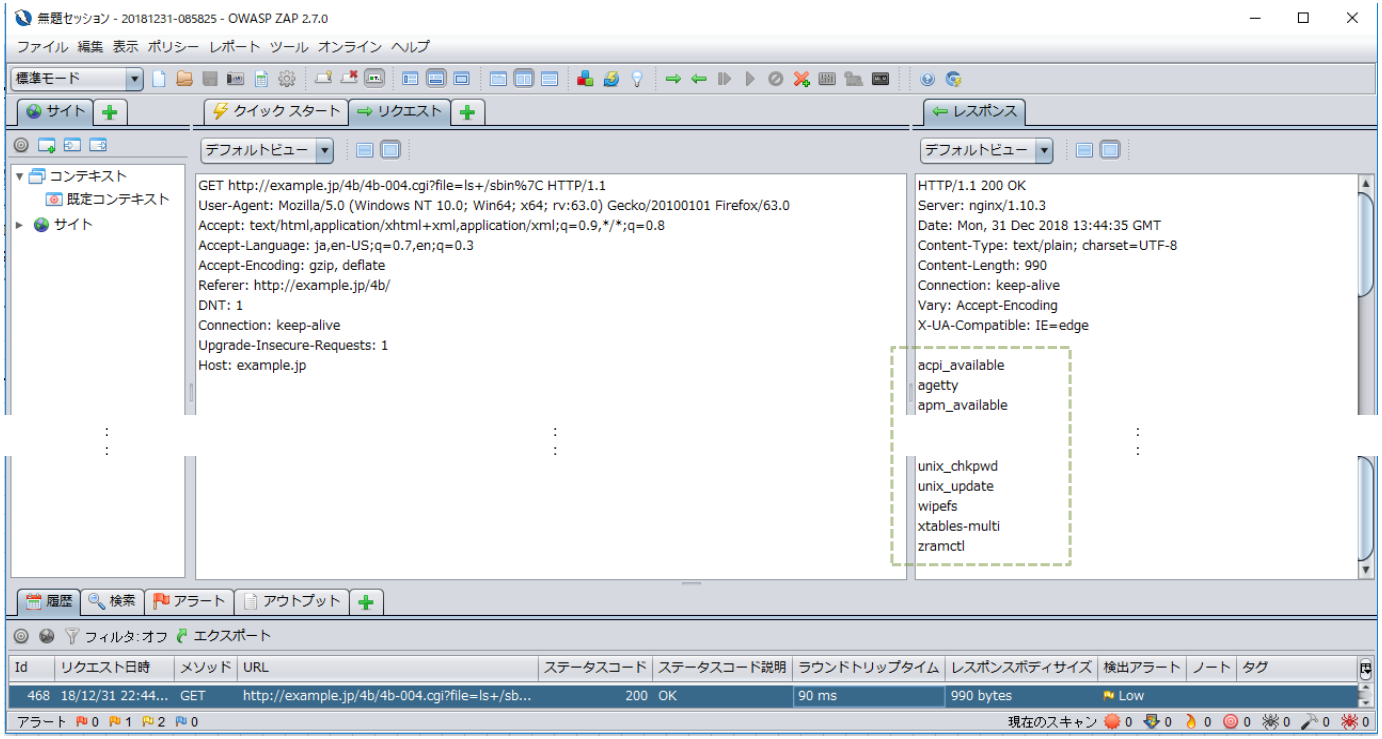
【サーバ： 4b/4b-004.cgi 】

/var/www/html/4b/4b-004.cgi - wasbook@example.jp - エディタ - WinSCP

```
#!/usr/bin/perl
use strict;
use utf8;
use open ':utf8';  # デフォルトの文字コードをUTF-8に
use CGI;

print "Content-Type: text/plain; charset=UTF-8\r\n\r\n";

my $q = new CGI;
my $file = $q->param('file');
open IN, $file or die $!;  # ファイルを開く

print <IN>;        # ファイルの内容を全て表示

close IN;          # ファイルクローズ
```

PerlのOPEN関数を用いて、ファイル名を外部から指定できる場合、
ファイル名の前後にパイプ記号「｜」を追加することで、
OSコマンド・インジェクション攻撃できる場合があります

【ブラウザ→サーバ：リクエスト 4b/4b-004.cgi → レスポンス 】



【ブラウザ】


example.jp/4b/4b-004.cgi?file=ls+/
example.jp/4b/4b-004.cgi?file=ls+/sbin
acpi_available
agetty
apm_available
badblocks
blkdeactivate
blkdiscard
blkid
blockdev
bridge
cfdisk
chcpu
ctrlaltdel
debugfs
depmod
devlink
:
:
:
:
sysctl
tc
telinit
tipc
tune2fs
udevadm
unix_chkpwd
unix_update
wipefs
xtables-multi
zramctl

4b-004:Perlのopen文によるOSコマンドインジェクション対策版（正常系）

【ブラウザ】

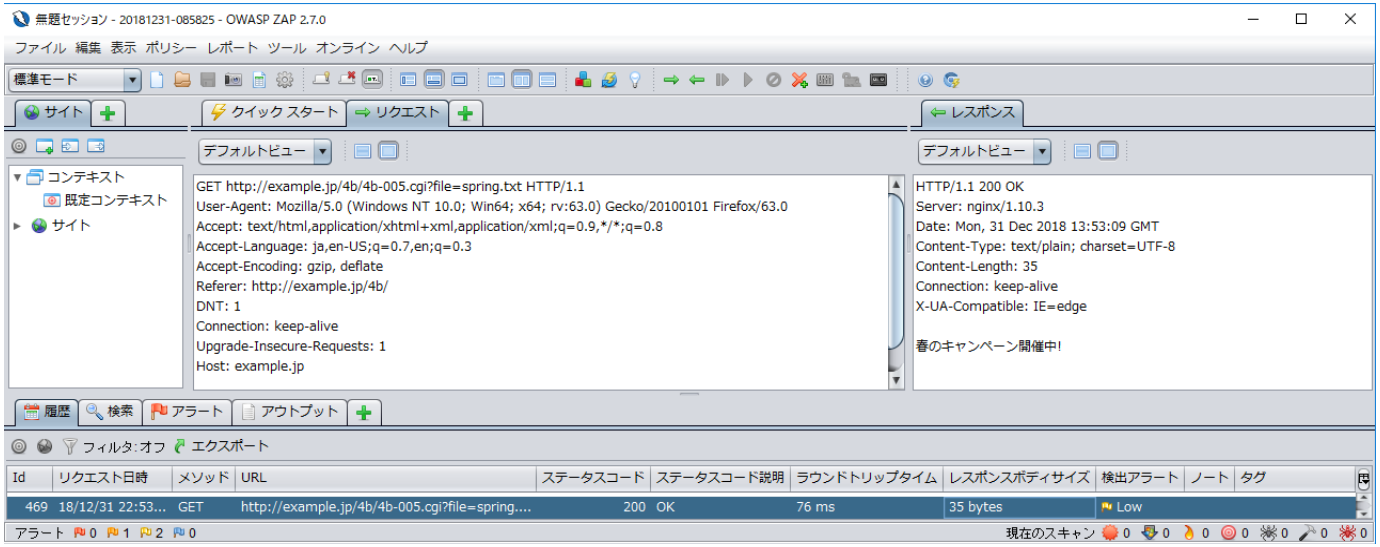
【サーバ： 4b/4b-005.cgi 】

/var/www/html/4b/4b-005.cgi - wasbook@example.jp - エディタ - WinSCP

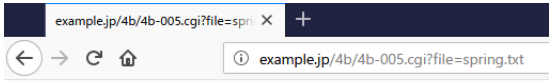
```
#!/usr/bin/perl
# OSコマンドインジェクション脆弱性対処版
# 但し、ディレクトリ・トラバーサル脆弱性は残っています
use strict;
use utf8;
use open ':utf8';  # デフォルトの文字コードをUTF-8に
use CGI;

print "Content-Type: text/plain; charset=UTF-8\r\n\r\n";

my $q = new CGI;
my $file = $q->param('file');
open IN, '<', $file or die $!;  # ファイルを開く
print <IN>;       # ファイルの内容を全て表示
close IN;         # ファイルクローズ
```

Perlのopen文によるOSコマンドインジェクション対策版

【ブラウザ→サーバ：リクエスト 4b/4b-005.cgi → レスポンス 】



【ブラウザ】



春のキャンペーン開催中!

4b-004:Perlのopen文によるOSコマンドインジェクション対策版（ls /sbin）

【ブラウザ】

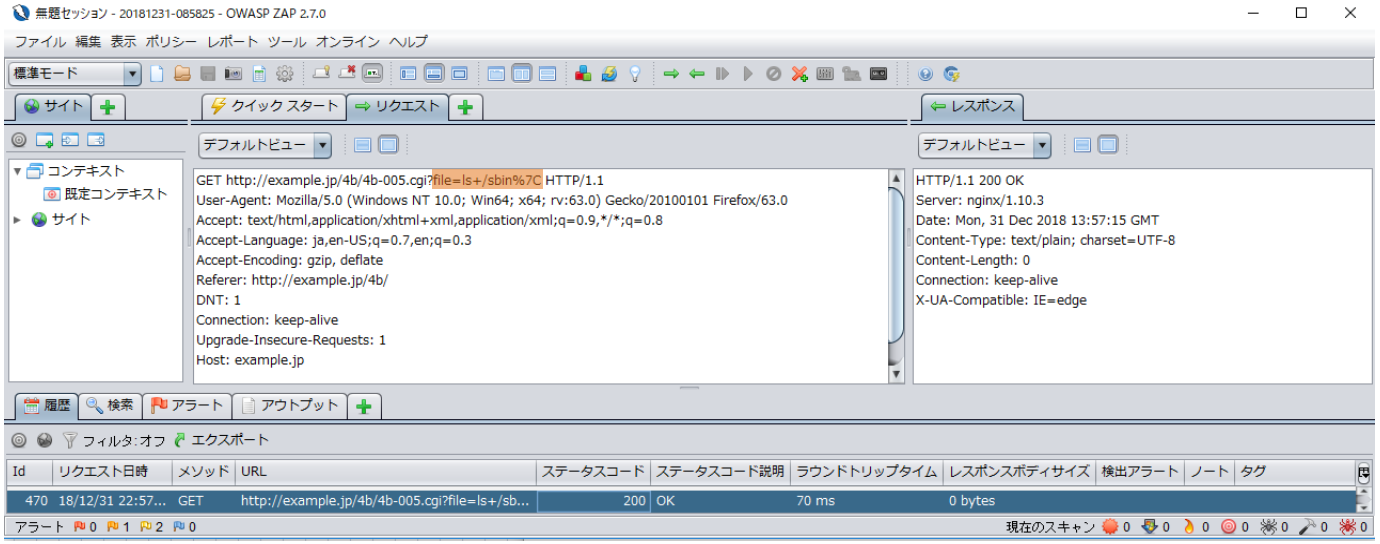
【サーバ： 4b/4b-005.cgi 】

/var/www/html/4b/4b-005.cgi - wasbook@example.jp - エディタ - WinSCP

```
#!/usr/bin/perl
# OSコマンドインジェクション脆弱性対処版
# 但し、ディレクトリ・トラバーサル脆弱性は残っています
use strict;
use utf8;
use open ':utf8';  # デフォルトの文字コードをUTF-8に
use CGI;

print "Content-Type: text/plain; charset=UTF-8\r\n\r\n";

my $q = new CGI;
my $file = $q->param('file');
open IN, '<', $file or die $!;  # ファイルを開く
print <IN>;       # ファイルの内容を全て表示
close IN;         # ファイルクローズ
```

Perlのopen文によるOSコマンドインジェクション対策版

【ブラウザ→サーバ：リクエスト 4b/4b-005.cgi → レスポンス 】



【ブラウザ】