

## 4.16 Web APIにまつわる問題

### JSON (JavaScript Object Notation) とは

JSONはJavaScriptから生まれたデータ記述フォーマットで、真偽値、数値、文字列、null値の組み合わせを持ったハッシュが配列かその両方の、ウェブ関連ではポピュラーなデータフォーマットです。JavaScriptとの違いを挙げると、JavaScriptでは文字列リテラルを囲うのはシングルクォートでもダブルクォートでも構いませんが、JSONではダブルクォートのみです。また、JSONではプロパティ名にもダブルクォートが必須です。JavaScriptにはNaNやundefinedなどの値がありますが、JSONには存在しません。JSONはJavaScriptの曖昧さを排除して、データ記述言語としての精度と、解析のし易さを確保しています。JavaScriptにネイティブなJSONサポート (JSON.parse, JSON.stringify) が定義されました。特にJSON.stringifyメソッドを使用すれば手軽にJavaScriptのオブジェクトからJSONを出力することができます。

サーバ側のJSON ⇄ オブジェクト (PHP)	JavaScript側のJSON ⇄ オブジェクト
json_encode PHP配列・オブジェクトからJSON文字列を生成	JSON.stringify JavaScriptのオブジェクトからJSON文字列を生成
json_decode JSON文字列をPHP配列に変換	JSON.parse JSON文字列をJavaScriptのオブジェクトに変換

### JSONP (JSON with Padding) とは

Paddingは(本来は不要なものの)付け足しという意味です。(JSONPはJSONではないので、JSONPはJSONとしてパースできません) JSONPはクロスドメインの制限を超えてデータをやり取りするために使用されます。CORSができる前に異なるオリジンのサーバからデータを取得する方法の1つがJSONPでした。JSONPはXMLHttpRequestではなく、script要素を用いて外部のJavaScriptを直接実行することにより、データを取得します。JSON文字列をそのままscript要素で受け取ることができないので、関数呼び出しの形でデータを生成します。

### JSONエスケープ不備による脆弱性

Web APIにおいてJSON文字列生成時のエスケープ処理に不備があると、意図しないJavaScriptがJSONデータに混入する場合があります。JSONデコードにeval関数を使っている場合や、JSONPのようにscript要素でJSON文字列を呼び込んでいる場合には、不正なJavaScriptの実行に至ります。

#### JSONエスケープ不備・脆弱性の影響

Webサイト利用者のブラウザ上でのJavaScriptの実行

#### JSONエスケープ不備・脆弱性の対策

- ①文字列連結によるJSONデータ生成をやめ、信頼できるライブラリを用いて、JSONを生成する
- ②eval関数ではなく、JSON.parseなどの安全なAPIでJSONを解釈する

### JSON直接閲覧によるXSS脆弱性

JSONを返すWeb APIは、通常XMLHttpRequestによるアクセスを想定したのですが、APIが返すレスポンスデータをブラウザで直接閲覧させることで、攻撃が可能になることがあります。

#### JSON直接閲覧によるXSS脆弱性の影響

Webサイト利用者のブラウザ上でのJavaScriptの実行

#### JSON直接閲覧によるXSS脆弱性の対策

- ①MIMEタイプを正しく設定する
- ②レスポンスヘッダ X-Content-Type-Options:nosniff を出力する
- ③<、> をUnicodeエスケープする

PHPの場合、json\_encodeのオプションパラメータで、エスケープが可能

オプション	エスケープ対象文字	エスケープ結果
JSON_HEX_TAG	< >	\\U003C \\U003E
JSON_HEX_AMP	&	\\U0026
JSON_HEX_APOS	'	\\U0027
JSON_HEX_QUOT	"	\\U0022

- ④XMLHttpRequestなどCORS対応の機能だけから呼び出せるようにする  
XMLHttpRequestからのリクエストには、「X-Requested-With: XMLHttpRequest」が付与され、script要素からのアクセスには付かないので、その違いによって、不正なアクセスをチェックできます

## JSONPのコールバック関数名によるXSS脆弱性

JSONPはJSONデータを引数とするコールバック関数呼び出しの形式を取るが、このコールバック関数名を外部から指定できるように構築するケースが多い  
これにより、コールバック関数名によるクロスサイトスクリプティングが可能になる場合があります

### JSONPのコールバック関数名によるXSS脆弱性の影響

Webサイト利用者のブラウザ上でのJavaScriptの実行

### JSONPのコールバック関数名によるXSS脆弱性の対策

- ①コールバック関数名の文字種と文字数を制限する ※ 英数字とアンダースコア「\_」のみに制限しても問題ないと思われる。<>を使えなくする。
- ②MIMEタイプを正しく設定する ※ JSONPはJSONではなく、JavaScriptなので、text/javascriptですが、MIMEタイプを指定しないと、text/htmlと解釈され、script要素がJavaScriptとして解釈される  
※ text/javascriptであれば、script要素は解釈されず、実行されることはありません

## Web APIのクロスサイトフォージェリ脆弱性

Web APIでもCSRF脆弱性は混入します。

### Web APIのクロスサイトフォージェリ脆弱性をつくHTTP通信経路について

- ①GETリクエスト
- ②HTTPフォームでMIMEタイプが(text/plain、application/x-www-form-urlencoded、multipart/form-data)であることを想定していて、MIMEタイプをチェックしていない場合
- ③プリフライトリクエストを必要としないシンプルなりクエストで、MIMEタイプをチェックしていない場合
- ④プリフライトリクエストを必要とするリクエストだが、プリフライトリクエストに対する処理が不適切

### Web APIのクロスサイトフォージェリ脆弱性の対策

- ①CSRFトークン(セッション変数にトークンを保持)
- ②2重送信クッキー 問題点: 以下①~③が原因で、セッションフィクセーションされる可能性がある ①クッキーモンスターバグ悪用のリスクがある ②サイトにXSS脆弱性がある場合がある ③HTTP強制による漏洩リスクがある
- ③XMLHttpRequestなどCORS対応の機能だけから呼び出せるようにする  
XMLHttpRequestからのリクエストには、「X-Request-With: XMLHttpRequest」が付与され、script要素からのアクセスには付かないので、その違いによって、不正なアクセスをチェックできます  
問題点: Adobe Flash Player に過去、カスタムリクエストヘッダを許可なく送信できる脆弱性があった(2014年7月に修正済み)

## JSONハイジャックの脆弱性

異サイトの工夫によって、JSONデータを異サイトのJavaScriptで読み出す手法がJSONハイジャックの脆弱性として、考案されてきて、ブラウザ側の対策が取られてきました  
今後も新しい手法が見つかることが考えられるので、以下の対策が必要です

### JSONハイジャックによるXSS脆弱性の対策

- ①レスポンスヘッダ X-Content-Type-Options:nosniff を出力する
- ②XMLHttpRequestなどCORS対応の機能だけから呼び出せるようにする  
XMLHttpRequestからのリクエストには、「X-Request-With: XMLHttpRequest」が付与され、script要素からのアクセスには付かないので、その違いによって、不正なアクセスをチェックできます

## CORSの検証不備

オリジンの制限をかけないのは情報漏洩のリスクになる NG  
オリジンを指定する OK


Access-Control-Allow-Origin: *
Access-Control-Allow-Origin: example.jp

CORS規定では、XMLHttpRequestリクエストのWithCredentialsプロパティを指定した場合、Access-Control-Allow-Originヘッダにオリジンを指定しなければならないが、手抜きが多い

## 基礎

### 4g-002 :JSONによる時計

#### 【ブラウザ】

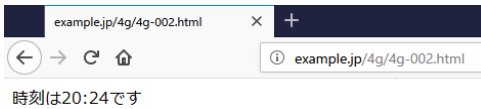


4.16 Web APIにまつわる問題

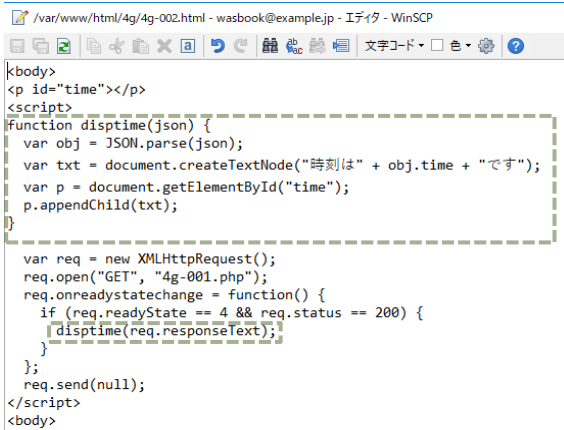
- 基礎
  - 1. [4g-002 :JSONによる時計](#)
  - 2. [4g-004 :JSONPによる時計](#)
  - 3. [4g-005 :JSONPによる時計\(jQuery\)](#)
- JSONエスケープの問題
  - 1. [4g-007 :JSONエスケープの不備\(正常系\)](#)
  - 2. [4g-007 :JSONエスケープの不備\(XSS攻撃\)](#)
  - 3. [4g-007a:JSONエスケープの対策版\(XSS攻撃\)](#)
- JSON直接閲覧によるXSS
  - 1. [4g-011 :JSON直接閲覧](#)
  - 2. [4g-011b:JSON直接閲覧\(XSS攻撃\)](#)
  - 3. [4g-011a:JSON直接閲覧\(Content-Type設定版\)\(IE9以前ならXSS可能\)](#)
  - 4. [4g-011b:JSON直接閲覧\(X-Content-Type-Options: nosniff設定版\)\(IE7以前ならXSS可能\)](#)
  - 5. [4g-011c:JSON直接閲覧\(タグ等もエスケープ\)\(IE7でもJavaScriptは実行されない\)](#)
  - 6. [4g-011d:JSON直接閲覧\(X-Requested-Withヘッダの確認; 403 Forbiddenとなる\)](#)
  - 7. [4g-012 :4g-011d.phpの呼び出し\(jQuery利用\)](#)
- JSONPコールバック関数名によるXSS
  - 1. [4g-016 :JSONPコールバック関数名によるXSS\(HTMLからの正常系利用\)](#)
  - 2. [4g-015 :JSONPコールバック関数名によるXSS\(攻撃でない直接閲覧\)](#)
  - 3. [4g-015 :JSONPコールバック関数名によるXSS\(XSS攻撃\)](#)
  - 4. [4g-015a:JSONPコールバック関数名によるXSS\(対策版\)](#)
- Web APIのCSRF脆弱性
  - 1. [4g-022 :CSRF脆弱なメールアドレス変更](#)
  - 2. [4g-921 :CSRF攻撃](#)
  - 3. [4g-022a:CSRF対策済みのメールアドレス変更](#)
  - 4. [4g-921a:CSRF攻撃](#)
  - 5. [4g-022b:CSRF対策済みのメールアドレス変更](#)
  - 6. [4g-921b:CSRF攻撃](#)
  - 7. [4g-022c:CSRF対策済みのメールアドレス変更](#)
  - 8. [4g-921c:CSRF攻撃](#)
  - 9. [4g-921d:CSRF攻撃\(2\)](#)
- JSONハイジャック
  - 1. [4g-030 :JSONのサンプル](#)
  - 2. [4g-930 :JSONをscript要素で読み出そうとする](#)
  - 3. [4g-931 :JSONハイジャック\(古いFirefoxで再現\)](#)
- JSONPの不適切な利用
  - 1. [4g-040 :ログイン\(api.example.net\)](#)
  - 2. [4g-042 :個人情報の表示\(正常系\)](#)
  - 3. [4g-042 :震サイトから個人情報の表示](#)

```
1 <html>
2 <head><title>4.16 Web APIにまつわる問題</title></head>
3 <body>
4 4.16 Web APIにまつわる問題
5 <ul>
6 <li>基礎</li>
7 <ol>
8 <li><a href="4g-002.html">4g-002 :JSONによる時計</a></li>
9 <li><a href="4g-004.html">4g-004 :JSONPによる時計</a></li>
10 <li><a href="4g-005.html">4g-005 :JSONPによる時計(jQuery)</a></li>
11 </ol>
12 <li>JSONエスケープの問題</li>
13 <ol>
14 <li><a href="4g-007.html#1">4g-007 :JSONエスケープの不備(正常系)</a></li>
15 <li><a href="4g-007.html#1&quot;%2balert(document.domain)%2b&quot;">4g-007 :JSONエスケープの不備(XSS攻撃)</a></li>
16 <li><a href="4g-007a.html#1&quot;%2balert(document.domain)%2b&quot;">4g-007a:JSONエスケープの対策版(XSS攻撃)</a></li>
17 </ol>
18 <li>JSON直接閲覧によるXSS</li>
19 <ol>
20 <li><a href="4g-011.php?zip=1">4g-011 :JSON直接閲覧</a></li>
21 <li><a href="4g-011.php?zip=1&imgsrc=1+onerror=alert(document.domain)&et;">4g-011 :JSON直接閲覧(XSS攻撃)</a></li>
22 <li><a href="4g-011a.php/a.html?zip=1&imgsrc=1+onerror=alert(document.domain)&et;">4g-011a:JSON直接閲覧(Content-Type設定版)(IE9以前ならXSS可能)</a></li>
23 <li><a href="4g-011b.php/a.html?zip=1&imgsrc=1+onerror=alert(document.domain)&et;">4g-011b:JSON直接閲覧(X-Content-Type-Options: nosniff設定版)(IE7以前ならXSS可能)</a></li>
24 <li><a href="4g-011c.php/a.html?zip=1&imgsrc=1+onerror=alert(document.domain)&et;">4g-011c:JSON直接閲覧(タグ等もエスケープ)(IE7でもJavaScriptは実行されない)</a></li>
25 <li><a href="4g-011d.php/a.html?zip=1&imgsrc=1+onerror=alert(document.domain)&et;">4g-011d:JSON直接閲覧(X-Requested-Withヘッダの確認; 403 Forbiddenとなる)</a></li>
26 <li><a href="4g-012.html#1">4g-012 :4g-011d.phpの呼び出し(jQuery利用)</a></li>
27 </ol>
28 <li>JSONPコールバック関数名によるXSS</li>
29 <ol>
30 <li><a href="4g-016.html">4g-016 :JSONPコールバック関数名によるXSS(HTMLからの正常系利用)</a></li>
31 <li><a href="http://api.example.net/4g/4g-015.php?callback=display_time">4g-015 :JSONPコールバック関数名によるXSS(攻撃でない直接閲覧)</a></li>
32 <li><a href="http://api.example.net/4g/4g-015.php?callback=%3Cscript%3Ealert(document.domain)%3C/script%3E">4g-015 :JSONPコールバック関数名によるXSS(XSS攻撃)</a></li>
33 <li><a href="http://api.example.net/4g/4g-015a.php?callback=%3Cscript%3Ealert(document.domain)%3C/script%3E">4g-015a:JSONPコールバック関数名によるXSS(対策版)</a></li>
34 </ol>
35 <li>Web APIのCSRF脆弱性</li>
36 <ol>
37 <li><a href="4g-022.html">4g-022 :CSRF脆弱なメールアドレス変更</a></li>
38 <li><a href="http://trap.example.com/4g/4g-921.html">4g-921 :CSRF攻撃</a></li>
39 <li><a href="4g-022a.html">4g-022a:CSRF対策済みのメールアドレス変更</a></li>
40 <li><a href="http://trap.example.com/4g/4g-921a.html">4g-921a:CSRF攻撃</a></li>
41 <li><a href="4g-022b.html">4g-022b:CSRF対策済みのメールアドレス変更</a></li>
42 <li><a href="http://trap.example.com/4g/4g-921b.html">4g-921b:CSRF攻撃</a></li>
43 <li><a href="4g-022c.html">4g-022c:CSRF対策済みのメールアドレス変更</a></li>
44 <li><a href="http://trap.example.com/4g/4g-921c.html">4g-921c:CSRF攻撃</a></li>
45 <li><a href="http://trap.example.com/4g/4g-921d.html">4g-921d:CSRF攻撃(2)</a></li>
46 </ol>
47 <li>JSONハイジャック</li>
48 <ol>
49 <li><a href="4g-030.json">4g-030 :JSONのサンプル</a></li>
50 <li><a href="http://trap.example.com/4g/4g-930.html">4g-930 :JSONをscript要素で読み出そうとする</a></li>
51 <li><a href="http://trap.example.com/4g/4g-931.html">4g-931 :JSONハイジャック(古いFirefoxで再現)</a></li>
52 </ol>
53 <li>JSONPの不適切な利用</li>
54 <ol>
55 <li><a href="http://api.example.net/4g/4g-040.php">4g-040 :ログイン(api.example.net)</a></li>
56 <li><a href="4g-042.html">4g-042 :個人情報の表示(正常系)</a></li>
57 <li><a href="http://trap.example.com/4g/4g-042.html">4g-042 :震サイトから個人情報の表示</a></li>
58 </ol>
59 </ul>
60 <a href="phpinfo.php">phpinfo</a><br>
61 <a href="/">ホームに戻る</a>
62 </body>
63 </html>
```

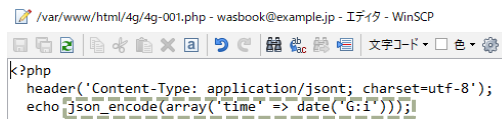
[phpinfo](#)  
[ホームに戻る](#)



#### 【サーバ: 4g/4g-002.html】



#### 【サーバ: 4g/4g-001.php】



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

The screenshot displays the OWASP ZAP interface. The top toolbar includes buttons for 'クイックスタート', 'リクエスト', and 'レスポンス'. The main window is split into two panes: 'リクエスト' (Request) on the left and 'レスポンス' (Response) on the right. The request pane shows a GET request to http://example.jp/4g/4g-002.html with headers including User-Agent, Accept, Accept-Language, Accept-Encoding, Referer, DNT, Connection, Upgrade-Insecure-Requests, and Host. The response pane shows an HTTP/1.1 200 OK response with headers including Server, Date, Content-Type, Content-Length, Connection, Last-Modified, ETag, Accept-Ranges, Vary, and X-UA-Compatible. The response body contains HTML tags and a JavaScript function named 'disptime' that takes a JSON object and appends the current time to a text node with the ID 'time'. Below the function is an XMLHttpRequest object that sends a GET request to 4g-001.php and calls the 'disptime' function with its response text.

```

GET http://example.jp/4g/4g-002.html 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/4g/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Sun, 06 Jan 2019 11:24:09 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 469
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "1d5-56c2a2dea6d98-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body>
<p id="time"></p>
<script>
function disptime(json) {
var obj = JSON.parse(json);
var txt = document.createTextNode("時刻は" + obj.time + "です");
var p = document.getElementById("time");
p.appendChild(txt);
}

var req = new XMLHttpRequest();
req.open("GET", "4g-001.php");
req.onreadystatechange = function() {
if (req.readyState == 4 && req.status == 200) {
disptime(req.responseText);
}
};
req.send(null);
</script>
<body>
  
```

Id	リクエスト日時	メソ...	URL	ステータスコ...	ステータスコード...	ラウンドトリップ...	レスポンスボディサ...	検出アラ...	ノート	タグ
68	19/01/06 20:2...	GET	http://example.jp/4g/4g-002.html	200	OK	7 ms	469 bytes	Medium		Script
71	19/01/06 20:2...	GET	http://example.jp/4g/4g-001.php	200	OK	24 ms	16 bytes	Low		JSON

アラート 0 1 2 0

現在のスキャン 0 0 0 0 0 0

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

無題セッション - 20190105-221150 - 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

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/4g/4g-001.php HTTP/1.1  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0  
Accept: \*/\*  
Accept-Language: ja,en-US;q=0.7,en;q=0.3  
Accept-Encoding: gzip, deflate  
Referer: http://example.jp/4g/4g-002.html  
DNT: 1  
Connection: keep-alive  
Host: example.jp

HTTP/1.1 200 OK  
Server: nginx/1.10.3  
Date: Sun, 06 Jan 2019 11:24:09 GMT  
Content-Type: application/json; charset=utf-8  
Connection: keep-alive  
X-Powered-By: PHP/5.3.3  
X-UA-Compatible: IE=edge

{\"time\":\"20:24\"}

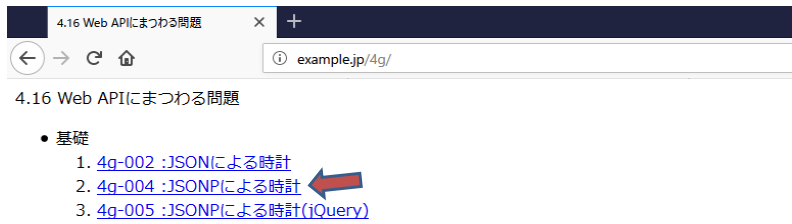
履歴 検索 アラート アウトプット +

フィルタ: オフ エクスポート

Id	リクエスト日時	メソ...	URL	ステータスコ...	ステータスコード...	ラウンドトリップ...	レスポンスボディサ...	検出アラ...	ノート	タグ
68	19/01/06 20:2...	GET	http://example.jp/4g/4g-002.html	200	OK	7 ms	469 bytes	Medium		Script
71	19/01/06 20:2...	GET	http://example.jp/4g/4g-001.php	200	OK	24 ms	16 bytes	Low		JSON

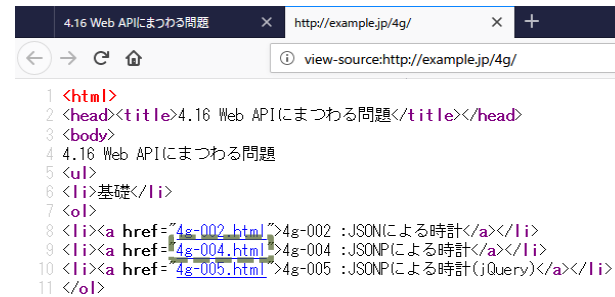
アラート 0 1 2 0 現在のスキャン 0 0 0 0 0 0

## 4g-004 :JSONPによる時計

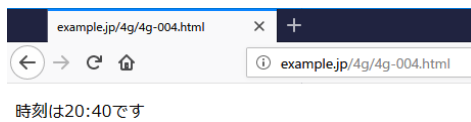


4.16 Web APIにまつわる問題

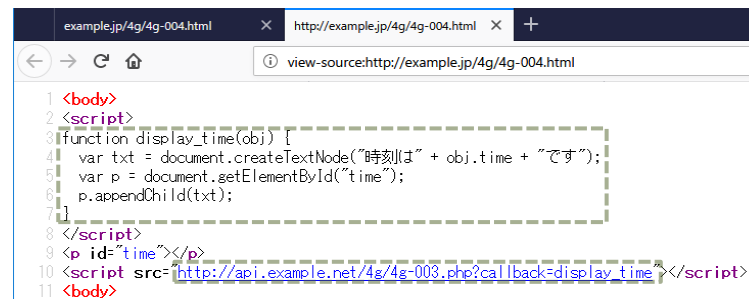
- 基礎
- 1. 4g-002 :JSONによる時計
- 2. 4g-004 :JSONPによる時計
- 3. 4g-005 :JSONPによる時計(jQuery)



```
1 <html>
2 <head><title>4.16 Web APIにまつわる問題</title></head>
3 <body>
4 4.16 Web APIにまつわる問題
5 <ul>
6 <li>基礎</li>
7 </ul>
8 <li><a href="/4g-002.html">4g-002 :JSONによる時計</a></li>
9 <li><a href="/4g-004.html">4g-004 :JSONPによる時計</a></li>
10 <li><a href="/4g-005.html">4g-005 :JSONPによる時計(jQuery)</a></li>
11 </ol>
```

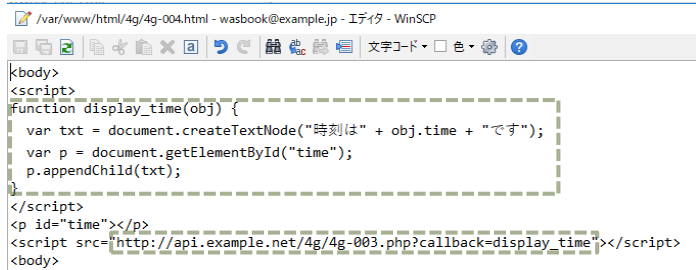


時刻は20:40です



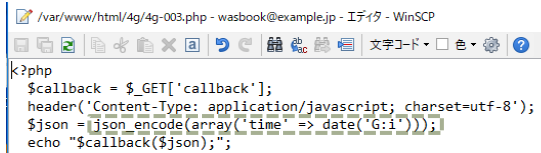
```
1 <body>
2 <script>
3 function display_time(obj) {
4   var txt = document.createTextNode("時刻は" + obj.time + "です");
5   var p = document.getElementById("time");
6   p.appendChild(txt);
7 }
8 </script>
9 <p id="time"></p>
10 <script src="/api.example.net/4g/4g-003.php?callback=display_time"></script>
11 </body>
```

### 【サーバ: 4g/4g-004.html】



```
function display_time(obj) {
  var txt = document.createTextNode("時刻は" + obj.time + "です");
  var p = document.getElementById("time");
  p.appendChild(txt);
}
<script>
<p id="time"></p>
<script src="/api.example.net/4g/4g-003.php?callback=display_time"></script>
```

### 【サーバ: 4g/4g-003.php】



```
$callback = $_GET['callback'];
header('Content-Type: application/javascript; charset=utf-8');
$json = json_encode(array('time' => date('G:i')));
echo "$callback($json)";
```

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

無題セッション - 20190105-221150 - 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

デフォルトビュー

```
GET http://example.jp/4g/4g-004.html 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/4g/
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: Sun, 06 Jan 2019 11:40:44 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 302
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "12e-56c2a2de98337-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body>
<script>
function display_time(obj) {
var txt = document.createTextNode("時刻は" + obj.time + "です");
var p = document.getElementById("time");
p.appendChild(txt);
}
</script>
<p id="time"></p>
<script src="http://api.example.net/4g/4g-003.php?callback=display_time"></script>
</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...	レスポンスボディサイズ	検出アラート	ノート	タグ
74	19/01/06 20:40:41	GET	http://example.jp/4g/4g-004.html	200	OK	6 ms	302 bytes	Medium		Script
75	19/01/06 20:40:42	GET	http://api.example.net/4g/4g-003.php?callback=displa...	200	OK	39 ms	31 bytes	Low		

アラート 0 1 3 0

現在のスキャン 0 0 0 0 0 0 0 0



## 【ブラウザ→APIサーバ: リクエスト 4g/4g-003.php → レスポンス】

The screenshot displays the OWASP ZAP 2.7.0 interface. The top menu bar includes 'ファイル', '編集', '表示', 'ポリシー', 'レポート', 'ツール', 'オンライン', and 'ヘルプ'. The main toolbar contains buttons for '標準モード', 'サイト', 'クイックスタート', 'リクエスト', and 'レスポンス'. The left sidebar shows a tree view with 'コンテキスト' and 'サイト'. The central pane is split into two sections: 'リクエスト' and 'レスポンス'. The request section shows a GET request to 'http://api.example.net/4g/4g-003.php?callback=display\_time' with various headers. The response section shows an 'HTTP/1.1 200 OK' status with headers and a JSON body: 'display\_time({"time": "20:40"})'. The bottom pane shows a table of request logs.

Id	リクエスト日時	メソッド	URL	ステータスコード	ステータスコード...	ラウンドトリップ...	レスポンスボディ...	検出アラ...	ノート	タグ
74	19/01/06 20:40:41	GET	http://example.jp/4g/4g-004.html	200	OK	6 ms	302 bytes	Medium		Script
75	19/01/06 20:40:42	GET	http://api.example.net/4g/4g-003.php?callback=display_time	200	OK	39 ms	31 bytes	Low		

アラート: 0 1 3 0

現在のスキャン: 0 0 0 0 0 0 0 0

## 4g-005 :JSONPによる時計(jQuery)

4.16 Web APIにまつわる問題

- 基礎
- 1. [4g-002 :JSONによる時計](#)
- 2. [4g-004 :JSONPによる時計](#)
- 3. [4g-005 :JSONPによる時計\(jQuery\)](#)

```
1 <html>
2 <head><title>4.16 Web APIにまつわる問題</title></head>
3 <body>
4 4.16 Web APIにまつわる問題
5 <ul>
6 <li>基礎</li>
7 </ul>
8 <li><a href="4g-002.html">4g-002 :JSONによる時計</a></li>
9 <li><a href="4g-004.html">4g-004 :JSONPによる時計</a></li>
10 <li><a href="4g-005.html">4g-005 :JSONPによる時計(jQuery)</a></li>
11 </ol>
```

時刻は20:55です

```
1 <body>
2 <script src="../js/jquery-3.2.1.min.js"></script>
3 <script>
4 function display_time(obj) {
5   $("#time").text("時刻は" + obj.time + "です");
6 }
7
8 $.ajax({
9   url: "http://api.example.net/4g/4g-003.php",
10  dataType: "jsonp",
11  jsonpCallback: "display_time"
12 });
13 </script>
14 <p id="time"></p>
15 </body>
```

### 【サーバ: 4g/4g-005.html】

```
<body>
<script src="../js/jquery-3.2.1.min.js"></script>
<script>
function display_time(obj) {
  $("#time").text("時刻は" + obj.time + "です");
}

$.ajax({
  url: "http://api.example.net/4g/4g-003.php",
  dataType: "jsonp",
  jsonpCallback: "display_time"
});
</script>
<p id="time"></p>
</body>
```

### 【サーバ: 4g/4g-003.php】

```
<?php
$callback = $_GET['callback'];
header('Content-Type: application/javascript; charset=utf-8');
$json = json_encode(array('time' => date('G:i')));
echo "$callback($json)";
```

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

The screenshot displays the OWASP ZAP 2.7.0 interface. The top menu includes 'ファイル', '編集', '表示', 'ポリシー', 'レポート', 'ツール', 'オンライン', and 'ヘルプ'. The main workspace is split into two panes: 'コンテキスト' (Context) on the left and 'デフォルトビュー' (Default View) on the right. The 'コンテキスト' pane shows a tree view with '既定コンテキスト' (Default Context) and 'サイト' (Site). The 'デフォルトビュー' pane shows the request and response details for a GET request to 'http://example.jp/4g/4g-005.html'.

**Request Details:**

```

GET http://example.jp/4g/4g-005.html 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/4g/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp
    
```

**Response Details:**

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Sun, 06 Jan 2019 11:55:46 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 300
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "12c-56c2a2de97397-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body>
<script src="..../js/jquery-3.2.1.min.js"></script>
<script>
function display_time(obj) {
$("#time").text("時刻は" + obj.time + "です");
}

$.ajax({
url: "http://api.example.net/4g/4g-003.php",
dataType: "json",
jsonpCallback: "display_time"
});
</script>
<p id="time"></p>
</body>
    
```

The bottom pane shows a table of request logs:

Id	リクエスト日時	メソッド	URL	ステータスコード	ステータススコ...	ラウンドトリップ...	レスポンス...	検出ア...	ノ...	タグ
78	19/01/06 20:55:44	GET	http://example.jp/4g/4g-005.html	200	OK	6 ms	300 bytes	Med...		Sc...
79	19/01/06 20:55:44	GET	http://example.jp/js/jquery-3.2.1.min.js	200	OK	21 ms	86,659 by...	Low		Fo...
81	19/01/06 20:55:44	GET	http://api.example.net/4g/4g-003.php?callback=display_time&_=1546775744919	200	OK	25 ms	31 bytes	Low		

At the bottom, there are status indicators for 'アラート' (Alerts) and '現在のスキャン' (Current Scan).

【ブラウザ→サーバ: リクエスト js/jquery-3.2.1.min.js → レスポンス】

無題セッション - 20190105-221150 - 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

リクエスト: GET http://example.jp/js/jquery-3.2.1.min.js HTTP/1.1  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0  
Accept: \*/\*  
Accept-Language: ja,en-US;q=0.7,en;q=0.3  
Accept-Encoding: gzip, deflate  
Referer: http://example.jp/4g/4g-005.html  
DNT: 1  
Connection: keep-alive  
Host: example.jp

レスポンス: HTTP/1.1 200 OK  
Server: nginx/1.10.3  
Date: Sun, 06 Jan 2019 11:55:46 GMT  
Content-Type: application/javascript  
Content-Length: 86659  
Connection: keep-alive  
Last-Modified: Mon, 14 May 2018 13:08:18 GMT  
ETag: "15283-56c2a2df00b1e-gzip"  
Accept-Ranges: bytes  
Vary: Accept-Encoding  
X-UA-Compatible: IE=edge

```

/*! jQuery v3.2.1 | (c) JS Foundation and other contributors | jquery.org/license */
(function(a,b){"use strict";object=="typeof module&&object=="typeof module.
exports?module.exports=a.document?b(a,!0):function(a){if(!a.document)throw new
Error("Query requires a window with a document");return b(a)}("undefined"!
=typeof window>window:this,function(a,b){"use strict";var c=[],d=a.document,e=
Object.getPrototypeOf,f=c.slice,g=c.concat,h=c.push,i=c.indexOf,j={},k=j.toString,l=j.

```

履歴 検索 アラート アウトプット +

Id	リクエスト日時	メソッド	URL	ステータスコード	ステータス...	ラウンドトリップ...	レスポンス...	検出ア...	ノ...	タグ
78	19/01/06 20:55:44	GET	http://example.jp/4g/4g-005.html	200	OK	6 ms	300 bytes	Med...	Sc...	
79	19/01/06 20:55:44	GET	http://example.jp/js/jquery-3.2.1.min.js	200	OK	21 ms	86,659 by...	Low	Fo...	
81	19/01/06 20:55:44	GET	http://api.example.net/4g/4g-003.php?callback=display_time&_=1546775744919	200	OK	25 ms	31 bytes	Low		

アラート 0 0 1 3 0 現在のスキャン 0 0 0 0 0 0 0 0 0 0

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

無題セッション - 20190105-221150 - 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

リクエスト: GET http://api.example.net/4g/4g-003.php?callback=display\_time&\_=1546775744919 HTTP/1.1  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0  
Accept: \*/\*  
Accept-Language: ja,en-US;q=0.7,en;q=0.3  
Accept-Encoding: gzip, deflate  
Referer: http://example.jp/4g/4g-005.html  
DNT: 1  
Connection: keep-alive  
Host: api.example.net

レスポンス: HTTP/1.1 200 OK  
Server: nginx/1.10.3  
Date: Sun, 06 Jan 2019 11:55:47 GMT  
Content-Type: application/javascript; charset=utf-8  
Content-Length: 31  
Connection: keep-alive  
X-Powered-By: PHP/5.3.3  
X-UA-Compatible: IE=edge

```

display_time({"time": "20:55"});

```

履歴 検索 アラート アウトプット +

Id	リクエスト日時	メソッド	URL	ステータスコード	ステータス...	ラウンドトリップ...	レスポンス...	検出ア...	ノ...	タグ
78	19/01/06 20:55:44	GET	http://example.jp/4g/4g-005.html	200	OK	6 ms	300 bytes	Med...	Sc...	
79	19/01/06 20:55:44	GET	http://example.jp/js/jquery-3.2.1.min.js	200	OK	21 ms	86,659 by...	Low	Fo...	
81	19/01/06 20:55:44	GET	http://api.example.net/4g/4g-003.php?callback=display_time&_=1546775744919	200	OK	25 ms	31 bytes	Low		

アラート 0 0 1 3 0 現在のスキャン 0 0 0 0 0 0 0 0 0 0

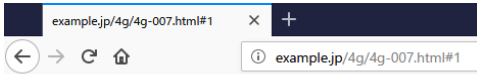
## JSONエスケープの問題 4g-002 :JSONによる時計

### 【ブラウザ】

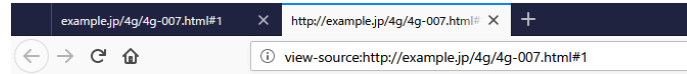
#### JSONエスケープの問題

- 4g-007 :JSONエスケープの不備(正常系)
- 4g-007 :JSONエスケープの不備(XSS攻撃)
- 4g-007a:JSONエスケープの対策版(XSS攻撃)

```
12 </li>JSONエスケープの問題</li>
13 <ol>
14 </li><a href="#4g-007">4g-007 :JSONエスケープの不備(正常系)</a></li>
15 </li><a href="#4g-007">4g-007 :JSONエスケープの不備(XSS攻撃)</a></li>
16 </li><a href="#4g-007a">4g-007a:JSONエスケープの対策版(XSS攻撃)</a></li>
17 </ol>
```



郵便番号が見つかりません:1



```
1 <body>
2 <script src="../js/jquery-3.2.1.min.js"></script>
3 <script>
4 $.ajax({
5   url: "http://api.example.net/4g/4g-006.php?zip=" + location.hash.slice(1),
6   dataType: "json",
7   jsonpCallback: "callback_zip",
8 }).done(function(data) {
9   $("#message").text(data.message);
10 });
11 </script>
12 <p id="message"></p>
13 </body>
```

### 【サーバ: 4g/4g-007.html 】

/var/www/html/4g/4g-007.html - wasbook@example.jp - エディタ - WinSCP

```
<body>
<script src="../js/jquery-3.2.1.min.js"></script>
<script>
$.ajax({
  url: "http://api.example.net/4g/4g-006.php?zip=" + location.hash.slice(1),
  dataType: "json",
  jsonpCallback: "callback_zip",
}).done(function(data) {
  $("#message").text(data.message);
});
</script>
<p id="message"></p>
</body>
```

### 【サーバ: 4g/4g-006.php 】

/var/www/html/4g/4g-006.php - wasbook@example.jp - エディタ - WinSCP

```
k?php
$zip = $_GET['zip'];
// 以下は郵便番号が見つからなかった場合の処理
$json = '{"message": "郵便番号が見つかりません: ' . $zip . '"}';
header('Content-Type: text/javascript; charset=utf-8');
echo "callback_zip($json);"
```

【ブラウザ→サーバ: リクエスト 4g/4g-007.html → レスポンス】

The screenshot displays the OWASP ZAP 2.7.0 interface. The main window is split into two panes: 'リクエスト' (Request) on the left and 'レスポンス' (Response) on the right. The request pane shows the following details:

```

GET http://example.jp/4g/4g-007.html 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/4g/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp
    
```

The response pane shows the following details:

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Sun, 06 Jan 2019 12:21:44 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 308
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "134-56c2a2dea9c78-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body>
<script src="..js/jquery-3.2.1.min.js"></script>
<script>
$.ajax({
  url: "http://api.example.net/4g/4g-006.php?zip=" + location.hash.slice(1),
  dataType: "jsonp",
  jsonpCallback: "callback_zip"
}).done(function(data) {
  s("#message").text(data.message);
});
</script>
<p id="message"></p>
</body>
    
```

At the bottom of the interface, there is a table showing a list of requests:

Id	リクエスト日時	メソ...	URL	ステータスコ...	ステータス...	ラウンドトリッ...	レスポンス...	検...	...	...
85	19/01/06 21:21...	GET	http://example.jp/4g/4g-007.html	200	OK	4 ms	308 bytes	...	...	S...
88	19/01/06 21:21...	GET	http://example.jp/js/jquery-3.2.1.min.js	200	OK	18 ms	86,659 b...	...	...	F...
90	19/01/06 21:21...	GET	http://api.example.net/4g/4g-006.php?zip=1&callback=callback_zip&_=1546777302741	200	OK	29 ms	67 bytes	...	...	L...

The status bar at the bottom indicates 'アラート' (Alerts) with counts: 0, 1, 2, 0 and '現在のスキャン' (Current Scan) with counts: 0, 0, 0, 0, 0, 0, 0.

【ブラウザ→サーバ: リクエスト js/jquery-3.2.1.min.js → レスポンス】

無題セッション - 20190105-221150 - 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

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/js/jquery-3.2.1.min.js HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: */*
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://example.jp/4g/4g-007.html
DNT: 1
Connection: keep-alive
Host: example.jp
```

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Sun, 06 Jan 2019 12:21:44 GMT
Content-Type: application/javascript
Content-Length: 86659
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "15283-56c2a2df00b1e-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

/*! jQuery v3.2.1 | (c) JS Foundation and other contributors | jquery.org/license
 */
!function(a,b){"use strict";"object"==typeof module&&"object"==typeof
module.exports?module.exports=a.document?b(a,0):function(a){if(!a.
document)throw new Error("jQuery requires a window with a document");
return b(a)}:b(a)}("undefined"!typeof window?window:this,function(a,b){
"use strict";var c=[],d=a.document,e=Object.getPrototypeOf,f=c.slice,g=c.
```

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...	URL	ステータスコ...	ステータス...	ラウンドトリッ...	レスポンス...	検...	...	...
85	19/01/06 21:21...	GET	http://example.jp/4g/4g-007.html	200	OK	4 ms	308 bytes	...	S...	
88	19/01/06 21:21...	GET	http://example.jp/js/jquery-3.2.1.min.js	200	OK	18 ms	86,659 b...	L...	F...	
90	19/01/06 21:21...	GET	http://api.example.net/4g/4g-006.php?zip=1&callback=callback_zip&_=1546777302741	200	OK	29 ms	67 bytes	L...		

アラート 0 1 2 0

現在のスキャン 0 0 0 0 0 0 0 0 0 0

【ブラウザ→APIサーバ: リクエスト 4g/4g-006.php → レスポンス】

The screenshot displays the OWASP ZAP 2.7.0 interface. The top menu bar includes 'ファイル', '編集', '表示', 'ポリシー', 'レポート', 'ツール', 'オンライン', and 'ヘルプ'. The main workspace is divided into two panes: 'リクエスト' (Request) and 'レスポンス' (Response). The request pane shows a GET request to 'http://api.example.net/4g/4g-006.php?zip=1&callback=callback\_zip&\_=1546777302741'. The response pane shows an HTTP 200 OK response with headers including 'Server: nginx/1.10.3', 'Date: Sun, 06 Jan 2019 12:21:44 GMT', and 'Content-Type: text/javascript; charset=utf-8'. The response body contains a JSON callback function: 'callback\_zip({"message": "郵便番号が見つかりません:1"})'. Below the workspace is a table of request history.

Id	リクエスト日時	メソ...	URL	ステータスコ...	ステータス...	ラウンドトリッ...	レスポンス...	検...	...	...
85	19/01/06 21:21...	GET	http://example.jp/4g/4g-007.html	200	OK	4 ms	308 bytes	...	S...	...
88	19/01/06 21:21...	GET	http://example.jp/js/jquery-3.2.1.min.js	200	OK	18 ms	86,659 b...	...	F...	...
90	19/01/06 21:21...	GET	http://api.example.net/4g/4g-006.php?zip=1&callback=callback_zip&_=1546777302741	200	OK	29 ms	67 bytes	...	L...	...

アラート: 0 1 2 0

現在のスキャン: 0 0 0 0 0 0 0 0



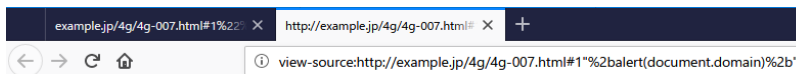
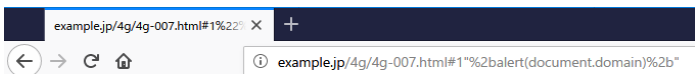
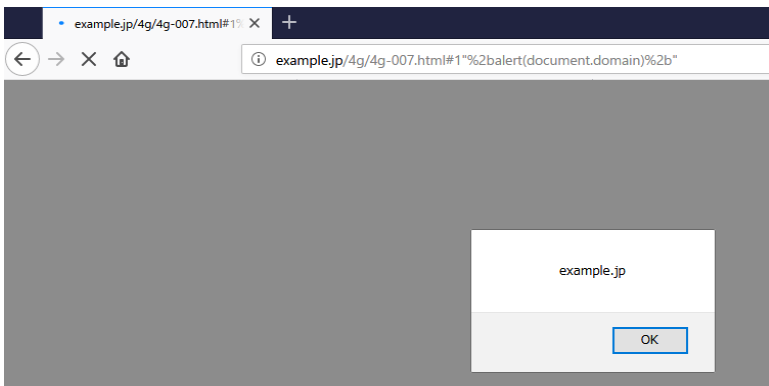
## 4g-002 :JSONによる時計

### 【ブラウザ】

- JSONエスケープの問題

- 4g-007 :JSONエスケープの不備(正常系)
- 4g-007 :JSONエスケープの不備(XSS攻撃) ←
- 4g-007a:JSONエスケープの対策版(XSS攻撃)

```
12 <li>JSONエスケープの問題</li>
13 </ol>
14 <li><a href="#4g-007.html#1">4g-007 :JSONエスケープの不備(正常系)</a></li>
15 <li><a href="#4g-007.html#1">4g-007 :JSONエスケープの不備(XSS攻撃)</a></li>
16 <li><a href="#4g-007a.html#1">4g-007a:JSONエスケープの対策版(XSS攻撃)</a></li>
17 </ol>
```



```
1 <body>
2 <script src="../js/jquery-3.2.1.min.js"></script>
3 <script>
4 $.ajax({
5   url: "http://api.example.net/4g/4g-006.php?zip=" + location.hash.slice(1),
6   dataType: "json",
7   jsonCallback: "callback_zip",
8 }).done(function(data) {
9   $('#message').text(data.message);
10 });
11 </script>
12 <p id="message"></p>
13 </body>
```

### 【サーバ: 4g/4g-007.html】

```
/var/www/html/4g/4g-007.html - wasbook@example.jp - エディタ - WinSCP
<body>
<script src="../../js/jquery-3.2.1.min.js"></script>
<script>
$.ajax({
  url: "http://api.example.net/4g/4g-006.php?zip=" + location.hash.slice(1),
  dataType: "json",
  jsonpCallback: "callback_zip"
}).done(function(data) {
  $('#message').text(data.message);
});
</script>
<p id="message"></p>
</body>
```

### 【サーバ: 4g/4g-006.php】

```
/var/www/html/4g/4g-006.php - wasbook@example.jp - エディタ - WinSCP
k?php
$zip = $_GET['zip'];
// 以下は郵便番号が見つからなかった場合の処理
$json = '{"message": "郵便番号が見つかりません: ' . $zip . '"}';
header('Content-Type: text/javascript; charset=utf-8');
echo "callback_zip($json);"
```

## 【ブラウザ→APIサーバ: リクエスト 4g/4g-006.php → レスポンス】

The screenshot displays the OWASP ZAP interface. The top toolbar includes buttons for 'サイト' (Site), 'クイックスタート' (Quick Start), 'リクエスト' (Request), and 'レスポンス' (Response). The main area is split into two panes: 'リクエスト' (Request) on the left and 'レスポンス' (Response) on the right. The request pane shows a GET request to `http://api.example.net/4g/4g-006.php?zip=1%22%2balert(document.domain)%2b%22&callback=callback_zip&_=154677997612`. The response pane shows an HTTP 200 OK response with headers including `Server: nginx/1.10.3`, `Date: Sun, 06 Jan 2019 12:33:19 GMT`, and `Content-Type: text/javascript; charset=utf-8`. The response body contains a JavaScript callback function: `callback_zip({"message": "郵便番号が見つかりません:1"+alert(document.domain)+""});`. At the bottom, a table lists the request details.

Id	リクエスト日時	メソッド	URL	ステータスコード	ステータス...	ラウンドトリッ...	レスポ...	検...	...	...
94	19/01/06 21:33:17	GET	http://api.example.net/4g/4g-006.php?zip=1%22%2balert(document.domain)%2b%22&callback=callback_zip&_=154677997612	200	OK	27 ms	93 bytes	L...		

現在のスキャン 0 0 0 0 0 0 0 0 0 0

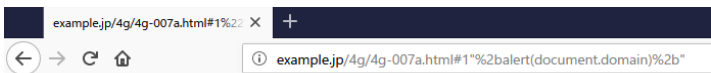
## 4g-007a:JSONエスケープの対策版(XSS攻撃)

### 【ブラウザ】

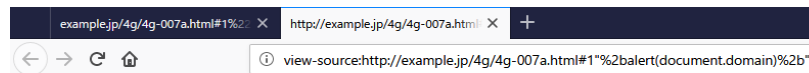
#### • JSONエスケープの問題

1. [4g-007 :JSONエスケープの不備\(正常系\)](#)
2. [4g-007 :JSONエスケープの不備\(XSS攻撃\)](#)
3. [4g-007a:JSONエスケープの対策版\(XSS攻撃\)](#)

```
12 </li>JSONエスケープの問題</li>
13 </ol>
14 <li><a href="4g-007.html#1">4g-007 :JSONエスケープの不備(正常系)</a></li>
15 <li><a href="4g-007.html#1"&quot;%2balert(document.domain)%2b">4g-007 :JSONエスケープの不備(XSS攻撃)</a></li>
16 <li><a href="4g-007a.html#1"&quot;%2balert(document.domain)%2b">4g-007a:JSONエスケープの対策版(XSS攻撃)</a></li>
17 </ol>
```



郵便番号が見つかりません:1'+alert(document.domain)+'



```
1 <body>
2 <script src= "../js/jquery-3.2.1.min.js"></script>
3 <script>
4 $.ajax({
5   url: "http://api.example.net/4g/4g-006a.php?zip=" + location.hash.slice(1),
6   dataType: "jsonp",
7   jsonpCallback: "callback_zip",
8 }).done(function(data) {
9   $('#message').text(data.message);
10});
11 </script>
12 <p id="message"></p>
13 </body>
```

### 【サーバ: 4g/4g-007a.html】

```
/var/www/html/4g/4g-007a.html - wasbook@example.jp - エディタ - WinSCP
<body>
<script src= "../js/jquery-3.2.1.min.js"></script>
<script>
$.ajax({
  url: "http://api.example.net/4g/4g-006a.php?zip=" + location.hash.slice(1),
  dataType: "jsonp",
  jsonpCallback: "callback_zip",
}).done(function(data) {
  $('#message').text(data.message);
});
</script>
<p id="message"></p>
</body>
```

### 【サーバ: 4g/4g-006a.php】

```
/var/www/html/4g/4g-006a.php - wasbook@example.jp - エディタ - WinSCP
<?php
$zip = $_GET['zip'];
// 以下は郵便番号が見つからなかった場合の処理
$json = json_encode(array("message" => "郵便番号が見つかりません:". $zip));
header('Content-Type: text/javascript; charset=utf-8');
echo "callback_zip($json)";
```

# 【ブラウザ→サーバ: リクエスト 4g/4g-007a.php → レスポンス】

The screenshot shows the network tab of a web browser's developer tools. The left pane shows the request details for 'GET http://example.jp/4g/4g-007a.html HTTP/1.1'. The right pane shows the response details for 'HTTP/1.1 200 OK'. The response body contains HTML code with an AJAX call to 'http://api.example.net/4g/4g-006a.php?zip=' followed by a JavaScript function to display the message.

```
GET http://example.jp/4g/4g-007a.html 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/4g/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Sun, 06 Jan 2019 12:41:18 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 309
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "135-56c2a2dea5df8-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body>
<script src="../../js/jquery-3.2.1.min.js"></script>
<script>
$.ajax({
  url: "http://api.example.net/4g/4g-006a.php?zip=" + location.hash.slice(1),
  dataType: "jsonp",
  jsonpCallback: "callback_zip"
}).done(function(data) {
  $('#message').text(data.message);
});
</script>
<p id="message"></p>
</body>
```

Id	リクエスト日時	メソッド	URL	ステータス...	ステータ...	ラウンドト...	レ...	...	...
95	19/01/06 21:41...	GET	http://example.jp/4g/4g-007a.html	200	OK	5 ms	309...	..	..
96	19/01/06 21:41...	GET	http://api.example.net/4g/4g-006a.php?zip=%22%2balert(document.domain)%2b%22&callback=callback...	200	OK	23 ms	131...	..	..

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 4g/4g-006a.php → レスポンス】

The screenshot displays the OWASP ZAP interface. The top toolbar includes buttons for 'クイックスタート' (Quick Start), 'リクエスト' (Request), and 'レスポンス' (Response). The main window is split into two panes: 'リクエスト' (Request) and 'レスポンス' (Response).

**Request:**

```
GET http://api.example.net/4g/4g-006a.php?zip=1%22%2balert(document.domain)%2b%22&callback=callback_zip&_=1546778476844 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: */*
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://example.jp/4g/4g-007a.html
DNT: 1
Connection: keep-alive
Host: api.example.net
```

**Response:**

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Sun, 06 Jan 2019 12:41:19 GMT
Content-Type: text/javascript; charset=utf-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
X-UA-Compatible: IE=edge

callback_zip({"message":"\u90f5\u4fbf\u756a\u53f7\u304c\u898b\u3064\u304b\u308a\u307e\u305b\u3093:1*"+alert(document.domain)+""});
```

At the bottom, a table shows the request and response details:

Id	リクエスト日時	メソッド	URL	ステ...	ス...	...	...	...	...
95	19/01/06 21:41:...	GET	http://example.jp/4g/4g-007a.html	200	OK	5...	...	...	...
96	19/01/06 21:41:...	GET	http://api.example.net/4g/4g-006a.php?zip=1%22%2balert(document.domain)%2b%22&callback=callback_zip&_=1546778476844	200	OK	2...	...	...	...

The status bar at the bottom shows 'アラート' (Alerts) with 0 counts and '現在のスキャン' (Current Scan) with 0 counts.

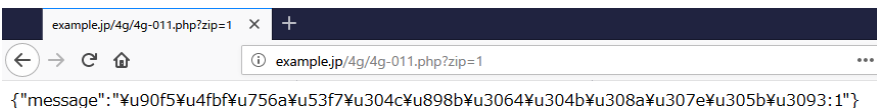
## JSON直接閲覧によるXSS 4g-011 :JSON直接閲覧

### 【ブラウザ】

#### • JSON直接閲覧によるXSS

1. [4g-011 :JSON直接閲覧](#)
2. [4g-011 :JSON直接閲覧\(XSS攻撃\)](#)
3. [4g-011a:JSON直接閲覧\(Content-Type設定版\)\(IE9以前ならXSS可能\)](#)
4. [4g-011b:JSON直接閲覧\(X-Content-Type-Options: nosniff設定版\)\(IE7以前ならXSS可能\)](#)
5. [4g-011c:JSON直接閲覧\(タグ等もエスケープ\)\(IE7でもJavaScriptは実行されない\)](#)
6. [4g-011d:JSON直接閲覧\(X-Requested-Withヘッダの確認: 403 Forbiddenとなる\)](#)
7. [4g-012 :4g-011d.phpの呼び出し\(iQuery利用\)](#)

```
18 </li>JSON直接閲覧によるXSS</li>
19 </ol>
20 <li><a href="4g-011.php?zip=1">4g-011 :JSON直接閲覧</a></li>
21 <li><a href="4g-011.php?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011 :JSON直接閲覧(XSS攻撃)</a></li>
22 <li><a href="4g-011a.php/a.html?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011a:JSON直接閲覧(Content-Type設定版)(IE9以前ならXSS可能)</a></li>
23 <li><a href="4g-011b.php/a.html?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011b:JSON直接閲覧(X-Content-Type-Options: nosniff設定版)(IE7以前ならXSS可能)</a></li>
24 <li><a href="4g-011c.php/a.html?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011c:JSON直接閲覧(タグ等もエスケープ)(IE7でもJavaScriptは実行されない)</a></li>
25 <li><a href="4g-011d.php/a.html?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011d:JSON直接閲覧(X-Requested-Withヘッダの確認: 403 Forbiddenとなる)</a></li>
26 <li><a href="4g-012.html#1">4g-012 :4g-011d.phpの呼び出し(iQuery利用)</a></li>
27 </ol>
```



### 【サーバ: 4g/4g-011.php】

```
/var/www/html/4g/4g-011.php - wasbook@example.jp - エディタ - WinSCP
k?php
$zip = $_GET['zip'];
// 以下は郵便番号が見つからなかった場合の処理
echo json_encode(array("message" => "郵便番号が見つかりません:" . $zip));
```

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

無題セッション - 20190107-015443 - 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

デフォルトビュー

コンテキスト

- 既定コンテ...
- サイト
  - http://e...

リクエスト

```
GET http://example.jp/4g/4g-011.php?zip=1 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/4g/
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, 07 Jan 2019 03:13:25 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 88
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

{"message": "\u90f5\u4fb\u756a\u53f7\u304c\u898b\u3064\u304b\u308a\u307e\u305b\u3093:1"}
```

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
7	19/01/07 3:13:25	GET	http://example.jp/4g/4g-011.php?zip=1	200	OK	101 ms	88 bytes	Medium		

アラート 0 1 2 0


現在のスキャン 0 0 0 0 0 0



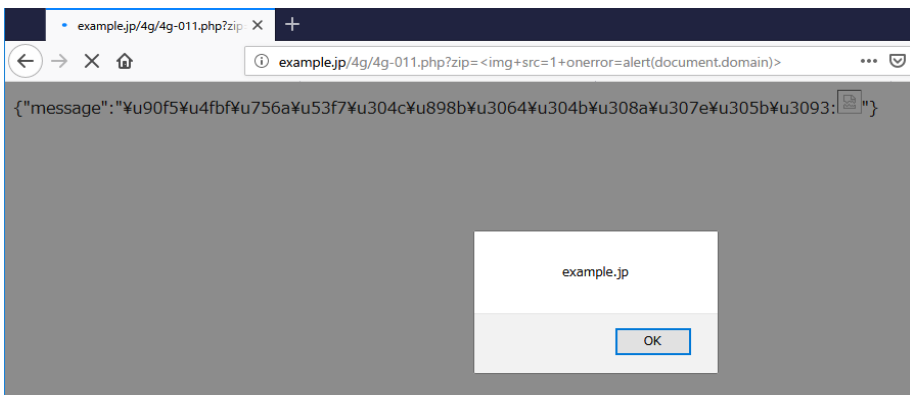
## 4g-011 :JSON直接閲覧(XSS攻撃)

### 【ブラウザ】

- JSON直接閲覧によるXSS

- 4g-011 :JSON直接閲覧
- 4g-011 :JSON直接閲覧(XSS攻撃) 
- 4g-011a:JSON直接閲覧(Content-Type設定版)(IE9以前ならXSS可能)
- 4g-011b:JSON直接閲覧(X-Content-Type-Options: nosniff設定版)(IE7以前ならXSS可能)
- 4g-011c:JSON直接閲覧(タグ等もエスケープ)(IE7でもJavaScriptは実行されない)
- 4g-011d:JSON直接閲覧(X-Requested-Withヘッダの確認; 403 Forbiddenとなる)
- 4g-012 :4g-011d.phpの呼び出し(jQuery利用)

```
18 <li>JSON直接閲覧によるXSS</li>
19 </ol>
20 <li><a href="4g-011.php?zip=">4g-011 :JSON直接閲覧</a></li>
21 <li><a href="4g-011.php?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011 :JSON直接閲覧(XSS攻撃)</a></li>
22 <li><a href="4g-011a.php/a.html?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011a:JSON直接閲覧(Content-Type設定版)(IE9以前ならXSS可能)</a></li>
23 <li><a href="4g-011b.php/a.html?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011b:JSON直接閲覧(X-Content-Type-Options: nosniff設定版)(IE7以前ならXSS可能)</a></li>
24 <li><a href="4g-011c.php/a.html?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011c:JSON直接閲覧(タグ等もエスケープ)(IE7でもJavaScriptは実行されない)</a></li>
25 <li><a href="4g-011d.php/a.html?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011d:JSON直接閲覧(X-Requested-Withヘッダの確認; 403 Forbiddenとなる)</a></li>
26 <li><a href="4g-012.html#1">4g-012 :4g-011d.phpの呼び出し(jQuery利用)</a></li>
27 </ol>
```



### 【サーバ: 4g/4g-011.php】

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

```
k?php
$zip = $_GET['zip'];
// 以下は郵便番号が見つからなかった場合の処理
echo [json_encode(array("message" => "郵便番号が見つかりません:" . $zip))];
```

JSONを返すので、MIMEタイプは `application/json` にすべきですが、指定していないので、デフォルトの `text/html` を返します  
そのため、JavaScriptが実行されてしまいます

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

The screenshot shows a successful HTTP request and response in Burp Suite. The request is a GET request to `http://example.jp/4g/4g-011.php?zip=%3Cimg+src=1+onerror=alert(document.domain)%3E`. The response is an HTTP/1.1 200 OK from a server running nginx/1.10.3. The response body contains a JavaScript alert message and an image tag with a payload.

```

GET http://example.jp/4g/4g-011.php?zip=%3Cimg+src=1+onerror=alert(document.domain)%3E 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/4g/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 03:21:19 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 129
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

{"message":"\u0f5\u4fbf\u756a\u53f7\u304c\u898b\u3064\u304b\u308a\u307e\u305b\u3093:"}
<img src=1 onerror=alert(document.domain)>
    
```

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...	ラウンドトリップ...	レスポンス...	検出...	...	...
9	19/01/07 3:21:18	GET	http://example.jp/4g/4g-011.php?zip=%3Cimg+src=1+onerror=alert(document.domain)%3E	200	OK	23 ms	129 bytes	M...		
10	19/01/07 3:21:18	GET	http://example.jp/4g/1	404	Not Found	18 ms	278 bytes	L...		

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

The screenshot shows a 404 Not Found response in Burp Suite. The request is a GET request to `http://example.jp/4g/1`. The response is an HTTP/1.1 404 Not Found from a server running nginx/1.10.3. The response body contains an HTML error message.

```

GET http://example.jp/4g/1 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
Accept: /*
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://example.jp/4g/4g-011.php?zip=%3Cimg+src=1+onerror=alert(document.domain)%3E
DNT: 1
Connection: keep-alive
Host: example.jp

HTTP/1.1 404 Not Found
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 03:21:19 GMT
Content-Type: text/html; charset=iso-8859-1
Connection: keep-alive

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /4g/1 was not found on this server.</p>
<hr>
<address>Apache/2.4.25 (Debian) Server at example.jp Port 88</address>
</body></html>
    
```

Id	リクエスト日時	メソ...	URL	ステータスコー...	ステータスコー...	ラウンドトリップ...	レスポ...	検...	...	...
9	19/01/07 3:21:...	GET	http://example.jp/4g/4g-011.php?zip=%3Cimg+src=1+onerror=alert(document.domain)%3E	200	OK	23 ms	129 bytes	M...		
10	19/01/07 3:21:...	GET	http://example.jp/4g/1	404	Not Found	18 ms	278 bytes	M...		

## 4g-011a:JSON直接閲覧(Content-Type設定版)(IE9以前ならXSS可能)

### 【ブラウザ】

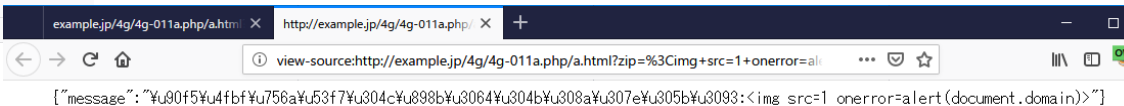
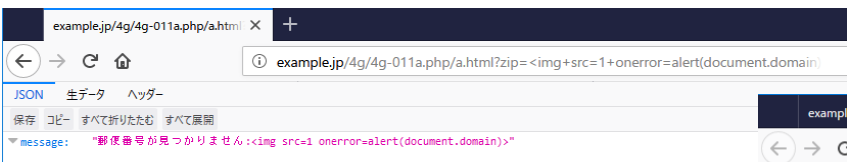
• JSON直接閲覧によるXSS

1. 4g-011 :JSON直接閲覧
2. 4g-011 :JSON直接閲覧(XSS攻撃)
3. 4g-011a:JSON直接閲覧(Content-Type設定版)(IE9以前ならXSS可能) ←
4. 4g-011b:JSON直接閲覧(X-Content-Type-Options: nosniff設定版)(IE7以前ならXSS可能)
5. 4g-011c:JSON直接閲覧(タグ等もエスケープ)(IE7でもJavaScriptは実行されない)
6. 4g-011d:JSON直接閲覧(X-Requested-Withヘッダの確認; 403 Forbiddenとなる)
7. 4g-012 :4g-011d.phpの呼び出し(jQuery利用)

```

18 </i>JSON直接閲覧によるXSS</i>
19 <ol>
20 </i><a href="4g-011.php?zip=1">4g-011 :JSON直接閲覧</a></i>
21 </i><a href="4g-011.php?zip=&lt;img src=1 onerror=alert(document.domain)&gt;">4g-011 :JSON直接閲覧(XSS攻撃)</a></i>
22 </i><a href="4g-011a.php/a.html?zip=&lt;img src=1 onerror=alert(document.domain)&gt;">4g-011a:JSON直接閲覧(Content-Type設定版)(IE9以前ならXSS可能)</a></i>
23 </i><a href="4g-011b.php/a.html?zip=&lt;img src=1 onerror=alert(document.domain)&gt;">4g-011b:JSON直接閲覧(X-Content-Type-Options: nosniff設定版)(IE7以前ならXSS可能)</a></i>
24 </i><a href="4g-011c.php/a.html?zip=&lt;img src=1 onerror=alert(document.domain)&gt;">4g-011c:JSON直接閲覧(タグ等もエスケープ)(IE7でもJavaScriptは実行されない)</a></i>
25 </i><a href="4g-011d.php/a.html?zip=&lt;img src=1 onerror=alert(document.domain)&gt;">4g-011d:JSON直接閲覧(X-Requested-Withヘッダの確認; 403 Forbiddenとなる)</a></i>
26 </i><a href="4g-012.html#1">4g-012 :4g-011d.phpの呼び出し(jQuery利用)</a></i>
27 </ol>

```



### 【サーバ: 4g/4g-011a.php】

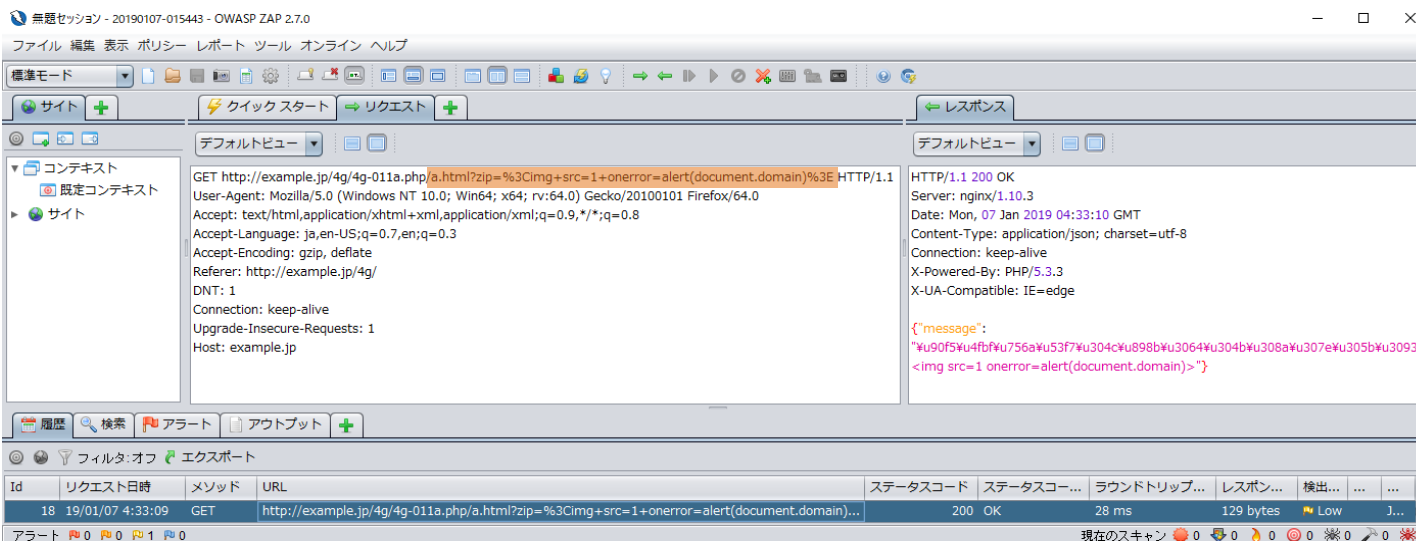
```

/var/www/html/4g/4g-011a.php - wasbook@example.jp - エディタ - WinSCP
k?php
$zip = $_GET["zip"];
// 以下は郵便番号が見つからなかった場合の処理
header('Content-Type: application/json; charset=utf-8');
echo json_encode(array("message" => "郵便番号が見つかりません:". $zip));

```

application/json を正しく指定しているが、/a.html(PATHINFO)を指定して、IEがContent-Typeを正しく判断しない脆弱性をつけています  
 IE9以前ではJavaScriptが実行されて、攻撃が成功します  
 レスポンスヘッダ X-Content-Type-Options:nosniff を出力するとJavaScript実行を回避できます  
 この設定でも、IE7以前ではJavaScriptは実行されてしまいます

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



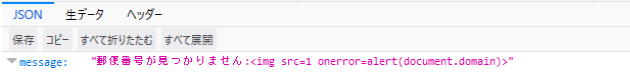
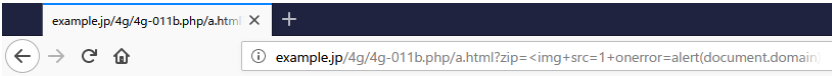
## 4g-011b:JSON直接閲覧(X-Content-Type-Options: nosniff設定版)(IE7以前ならXSS可能)

### 【ブラウザ】

#### • JSON直接閲覧によるXSS

1. [4g-011 :JSON直接閲覧](#)
2. [4g-011 :JSON直接閲覧\(XSS攻撃\)](#)
3. [4g-011a:JSON直接閲覧\(Content-Type設定版\)\(IE9以前ならXSS可能\)](#)
4. [4g-011b:JSON直接閲覧\(X-Content-Type-Options: nosniff設定版\)\(IE7以前ならXSS可能\)](#)
5. [4g-011c:JSON直接閲覧\(タグ等もエスケープ\)\(IE7でもJavaScriptは実行されない\)](#)
6. [4g-011d:JSON直接閲覧\(X-Requested-Withヘッダの確認、403 Forbiddenとなる\)](#)
7. [4g-012 :4g-011d.phpの呼び出し\(iQuery利用\)](#)

```
18 </li>JSON直接閲覧によるXSS</li>
19 </ol>
20 <li><a href="4g-011.php?zip=1">4g-011 :JSON直接閲覧</a></li>
21 <li><a href="4g-011.php?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011 :JSON直接閲覧(XSS攻撃)</a></li>
22 <li><a href="4g-011a.php/a.html?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011a:JSON直接閲覧(Content-Type設定版)(IE9以前ならXSS可能)</a></li>
23 <li><a href="4g-011b.php/a.html?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011b:JSON直接閲覧(X-Content-Type-Options: nosniff設定版)(IE7以前ならXSS可能)</a></li>
24 <li><a href="4g-011c.php/a.html?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011c:JSON直接閲覧(タグ等もエスケープ)(IE7でもJavaScriptは実行されない)</a></li>
25 <li><a href="4g-011d.php/a.html?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011d:JSON直接閲覧(X-Requested-Withヘッダの確認、403 Forbiddenとなる)</a></li>
26 <li><a href="4g-012.html#1">4g-012 :4g-011d.phpの呼び出し(iQuery利用)</a></li>
27 </ol>
```



### 【サーバ:4g/4g-011b.php】

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

```
k?php
$zip = $_GET['zip'];
// 以下は郵便番号が見つからなかった場合の処理
header('Content-Type: application/json; charset=utf-8');
header('X-Content-Type-Options: nosniff');
echo json_encode(array("message" => "郵便番号が見つかりません:" . $zip));
```

application/json を正しく指定しているが、/a.html(PATHINFO)を指定して、IEがContent-Typeを正しく判断しない脆弱性があります。IE9以前ではJavaScriptが実行されて、攻撃が成功します。

レスポンスヘッダ X-Content-Type-Options:nosniff を出力するとJavaScript実行を回避できます。

この設定でも、IE7以前ではJavaScriptは実行されてしまいます。

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

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...	ラウンドトリップ...	レスポン...	検出...	...
22	19/01/07 4:36:59	GET	http://example.jp/4g/4g-011b.php/a.html?zip=%3Cimg+src=1+onerror=alert(document.domain)...	200	OK	26 ms	129 bytes	J...	

## 4g-011c:JSON直接閲覧(タグ等もエスケープ)(IE7でもJavaScriptは実行されない)

### 【ブラウザ】

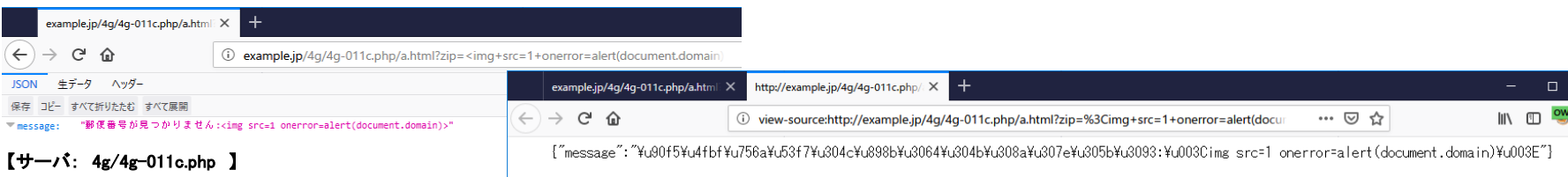
- JSON直接閲覧によるXSS

- 4g-011 :JSON直接閲覧
- 4g-011 :JSON直接閲覧(XSS攻撃)
- 4g-011a:JSON直接閲覧(Content-Type設定版)(IE9以前ならXSS可能)
- 4g-011b:JSON直接閲覧(X-Content-Type-Options: nosniff設定版)(IE7以前ならXSS可能)
- 4g-011c:JSON直接閲覧(タグ等もエスケープ)(IE7でもJavaScriptは実行されない)
- 4g-011d:JSON直接閲覧(X-Requested-Withヘッダの確認; 403 Forbiddenとなる)
- 4g-012 :4g-011d.phpの呼び出し(jQuery利用)

```

18 </i>JSON直接閲覧によるXSS</i>
19 </ol>
20 </i><a href="4g-011.php?zip=1">4g-011 :JSON直接閲覧</a></i>
21 </i><a href="4g-011.php?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011 :JSON直接閲覧(XSS攻撃)</a></i>
22 </i><a href="4g-011a.php/a.html?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011a:JSON直接閲覧(Content-Type設定版)(IE9以前ならXSS可能)</a></i>
23 </i><a href="4g-011b.php/a.html?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011b:JSON直接閲覧(X-Content-Type-Options: nosniff設定版)(IE7以前ならXSS可能)</a></i>
24 </i><a href="4g-011c.php/a.html?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011c:JSON直接閲覧(タグ等もエスケープ)(IE7でもJavaScriptは実行されない)</a></i>
25 </i><a href="4g-011d.php/a.html?zip=&lt;img src=1+onerror=alert(document.domain)&gt;">4g-011d:JSON直接閲覧(X-Requested-Withヘッダの確認; 403 Forbiddenとなる)</a></i>
26 </i><a href="4g-012.html#1">4g-012 :4g-011d.phpの呼び出し(jQuery利用)</a></i>
27 </ol>

```



### 【サーバ: 4g/4g-011c.php】

```

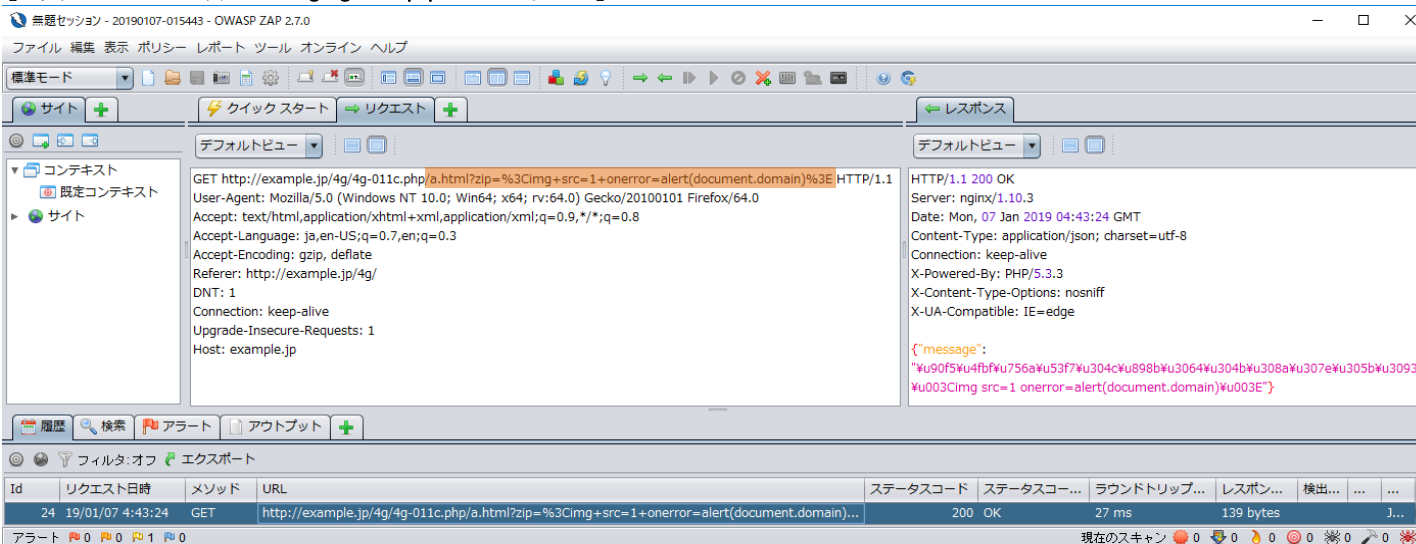
/var/www/html/4g/4g-011c.php - wasbook@example.jp - エディタ - WinSCP
k?php
$zip = $_GET["zip"];
// 以下は郵便番号が見つからなかった場合の処理
header('Content-Type: application/json; charset=utf-8');
header('X-Content-Type-Options: nosniff');
echo json_encode(array("message" => "郵便番号が見つかりません:" . $zip), JSON_HEX_TAG | JSON_HEX_AMP | JSON_HEX_APOS | JSON_HEX_QUOT);

```

application/json を正しく指定しているが、/a.html(PATHINFO)を指定して、IEがContent-Typeを正しく判断しない脆弱性についています。IE9以前ではJavaScriptが実行されて、攻撃が成功します。レスポンスヘッダ X-Content-Type-Options:nosniff を出力するとJavaScript実行を回避できます。この設定でも、IE7以前ではJavaScriptは実行されてしまいます。

PHPの場合、json\_encodeのオプションパラメータで、[ < > ] をUnicodeエスケープできるので、IE7以前でもJavaScriptの実行を回避できます。


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



## 4g-011d:JSON直接閲覧(X-Requested-Withヘッダの確認; 403 Forbiddenとなる)

### 【ブラウザ】

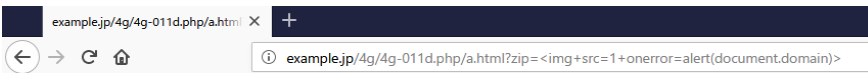
#### • JSON直接閲覧によるXSS

1. [4g-011 :JSON直接閲覧](#)
2. [4g-011 :JSON直接閲覧\(XSS攻撃\)](#)
3. [4g-011a:JSON直接閲覧\(Content-Type設定版\)\(IE9以前ならXSS可能\)](#)
4. [4g-011b:JSON直接閲覧\(X-Content-Type-Options: nosniff設定版\)\(IE7以前ならXSS可能\)](#)
5. [4g-011c:JSON直接閲覧\(タグ等もエスケープ\)\(IE7でもJavaScriptは実行されない\)](#)
6. [4g-011d:JSON直接閲覧\(X-Requested-Withヘッダの確認; 403 Forbiddenとなる\)](#) 
7. [4g-012 :4g-011d.phpの呼び出し\(jQuery利用\)](#)

```

18 </li>JSON直接閲覧によるXSS</li>
19 </ol>
20 <li><a href="4g-011.php?zip=1">4g-011 :JSON直接閲覧</a></li>
21 <li><a href="4g-011.php?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011 :JSON直接閲覧(XSS攻撃)</a></li>
22 <li><a href="4g-011a.php/a.html?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011a:JSON直接閲覧(Content-Type設定版)(IE9以前ならXSS可能)</a></li>
23 <li><a href="4g-011b.php/a.html?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011b:JSON直接閲覧(X-Content-Type-Options: nosniff設定版)(IE7以前ならXSS可能)</a></li>
24 <li><a href="4g-011c.php/a.html?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011c:JSON直接閲覧(タグ等もエスケープ)(IE7でもJavaScriptは実行されない)</a></li>
25 <li><a href="4g-011d.php/a.html?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011d:JSON直接閲覧(X-Requested-Withヘッダの確認; 403 Forbiddenとなる)</a></li>
26 <li><a href="4g-012.html#1">4g-012 :4g-011d.phpの呼び出し(jQuery利用)</a></li>
27 </ol>

```



不正な呼び出しです

### 【サーバ: 4g/4g-011d.php 】

```

/var/www/html/4g/4g-011d.php - wasbook@example.jp - エディタ - WinSCP
k?php
if (empty($_SERVER['HTTP_X_REQUESTED_WITH'])) {
    header('HTTP/1.1 403 Forbidden');
    die("不正な呼び出しです");
}
$zip = $_GET['zip'];
// 以下は郵便番号が見つからなかった場合の処理
header('Content-Type: application/json; charset=utf-8');
header('X-Content-Type-Options: nosniff');
echo json_encode(array("message" => "郵便番号が見つかりません:" . $zip), JSON_HEX_TAG | JSON_HEX_AMP | JSON_HEX_APOS | JSON_HEX_QUOT);

```

XMLHttpRequestからのリクエストには、「X-Requested-With: XMLHttpRequest」

application/json を正しく指定しているが、/a.html(PATHINFO)を指定して、IEがContent-Typeを正しく判断しない脆弱性をつけています  
IE9以前ではJavaScriptが実行されて、攻撃が成功します  
レスポンスヘッダ X-Content-Type-Options:nosniff を出力するとJavaScript実行を回避できます  
この設定でも、IE7以前ではJavaScriptは実行されてしまいます  
PHPの場合、json\_encodeのオプションパラメータで、[ < > ] をUnicodeエスケープできるので、IE7以前でも JavaScript の実行を回避できます

さらに左吹き出しの対処を追加しています

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

無題セッション - 20190107-015443 - 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

GET http://example.jp/4g/4g-011d.php/a.html?zip=%3Cimg+src=1+onerror=alert(document.domain)%3E 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/4g/  
 DNT: 1  
 Connection: keep-alive  
 Upgrade-Insecure-Requests: 1  
 Host: example.jp

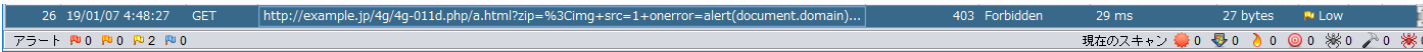
HTTP/1.1 403 Forbidden  
 Server: nginx/1.10.3  
 Date: Mon, 07 Jan 2019 04:48:28 GMT  
 Content-Type: text/html; charset=UTF-8  
 Connection: keep-alive  
 X-Powered-By: PHP/5.3.3  
 X-UA-Compatible: IE=edge

不正な呼び出しです

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...	ラウンドトリップ...	レスポン...	検出...	...
----	---------	------	-----	----------	------------	-------------	---------	-------	-----

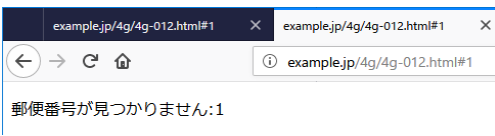


## 4g-012 :4g-011d.phpの呼び出し(jQuery利用)

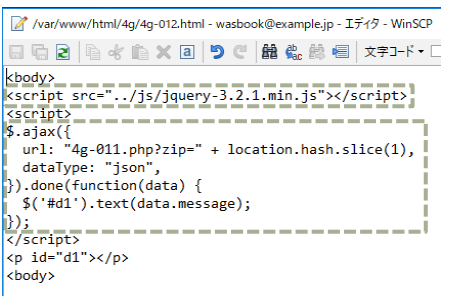
### 【ブラウザ】

- JSON直接閲覧によるXSS
  1. 4g-011 :JSON直接閲覧
  2. 4g-011 :JSON直接閲覧(XSS攻撃)
  3. 4g-011a:JSON直接閲覧(Content-Type設定版)(IE9以前ならXSS可能)
  4. 4g-011b:JSON直接閲覧(X-Content-Type-Options: nosniff設定版)(IE7以前ならXSS可能)
  5. 4g-011c:JSON直接閲覧(タグ等もエスケープ)(IE7でもJavaScriptは実行されない)
  6. 4g-011d:JSON直接閲覧(X-Requested-Withヘッダの確認; 403 Forbiddenとなる)
  7. 4g-012 :4g-011d.phpの呼び出し(jQuery利用) 

```
18 </li>JSON直接閲覧によるXSS</li>
19 </ol>
20 <li><a href="4g-011.php?zip=1">4g-011 :JSON直接閲覧</a></li>
21 <li><a href="4g-011.php?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011 :JSON直接閲覧(XSS攻撃)</a></li>
22 <li><a href="4g-011a.php/a.html?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011a:JSON直接閲覧(Content-Type設定版)(IE9以前ならXSS可能)</a></li>
23 <li><a href="4g-011b.php/a.html?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011b:JSON直接閲覧(X-Content-Type-Options: nosniff設定版)(IE7以前ならXSS可能)</a></li>
24 <li><a href="4g-011c.php/a.html?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011c:JSON直接閲覧(タグ等もエスケープ)(IE7でもJavaScriptは実行されない)</a></li>
25 <li><a href="4g-011d.php/a.html?zip=&lt;img+src=1+onerror=alert(document.domain)&gt;">4g-011d:JSON直接閲覧(X-Requested-Withヘッダの確認; 403 Forbiddenとなる)</a></li>
26 <li><a href="4g-012.html#1">4g-012 :4g-011d.phpの呼び出し(jQuery利用)</a></li>
27 </ol>
```



### 【サーバ: 4g/4g-012.html】



【ブラウザ→サーバ: リクエスト 4g/4g-012.html → レスポンス】

The screenshot displays the OWASP ZAP 2.7.0 interface. The main window is titled "無題セッション - 20190107-015443 - OWASP ZAP 2.7.0". The interface is divided into several sections:

- Left Panel:** Contains a tree view with "コンテキスト" (Contexts) and "サイト" (Sites) folders. Under "サイト", there is a sub-entry for "既定コンテキスト" (Default Context).
- Request Panel (デフォルトビュー):** Shows the details of the request:
 

```
GET http://example.jp/4g/4g-012.html 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/4g/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp
```
- Response Panel (デフォルトビュー):** Shows the details of the response:
 

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 05:02:33 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 239
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "ef-56c2a2dea0037-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body>
<script src="..js/jquery-3.2.1.min.js"></script>
<script>
$.ajax({
  url: "4g-011.php?zip=" + location.hash.slice(1),
  dataType: "json",
}).done(function(data) {
  $("#d1").text(data.message);
});
</script>
<p id="d1"></p>
</body>
```
- Bottom Panel:** Contains a table of request history and a status bar.
 

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...	ラウンドトリップ...	レスポンス...	検出...	...	...
35	19/01/07 5:02:32	GET	http://example.jp/4g/4g-012.html	200	OK	3 ms	239 bytes	M...	S...	
37	19/01/07 5:02:32	GET	http://example.jp/js/jquery-3.2.1.min.js	200	OK	16 ms	86,659 by...	Low	F...	
39	19/01/07 5:02:32	GET	http://example.jp/4g/4g-011.php?zip=1	200	OK	19 ms	88 bytes	M...		

Alerts: 0 (red), 1 (yellow), 2 (green), 0 (blue). Current scan status: 0 (red), 0 (yellow), 0 (green), 0 (blue), 0 (purple), 0 (pink), 0 (orange).



【ブラウザ→サーバ: リクエスト js/jquery-3.2.1.min.js → レスポンス】

The screenshot shows the OWASP ZAP interface with a request and response view. The request is a GET for http://example.jp/js/jquery-3.2.1.min.js. The response is an HTTP/1.1 200 OK from nginx/1.10.3, containing the jQuery v3.2.1 library code.

**Request:**

```
GET http://example.jp/js/jquery-3.2.1.min.js HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
Accept: */*
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://example.jp/4g/4g-012.html
DNT: 1
Connection: keep-alive
Host: example.jp
```

**Response:**

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 05:02:33 GMT
Content-Type: application/javascript
Content-Length: 86659
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "15283-56c2a2df001e-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

/*! jQuery v3.2.1 | (c) JS Foundation and other contributors | jquery.org/licen
se */
!function(a,b){"use strict";"object"==typeof module&&"object"==typeof
module.exports?module.exports=a.document?b(a,!0):function(a){if(!a.
```

**Request 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...	ラウンドトリップ...	レスポンス...	検出...	...	...
35	19/01/07 5:02:32	GET	http://example.jp/4g/4g-012.html	200	OK	3 ms	239 bytes	M...	S...	
37	19/01/07 5:02:32	GET	http://example.jp/js/jquery-3.2.1.min.js	200	OK	16 ms	86,659 by...	Low	F...	
39	19/01/07 5:02:32	GET	http://example.jp/4g/4g-011.php?zip=1	200	OK	19 ms	88 bytes	M...		

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

jQueryなどの著名なJavaScriptライブラリは、XMLHttpRequestによるHTTPリクエストに、X-Requested-Withヘッダを付与する

The screenshot displays the OWASP ZAP interface. The top toolbar includes buttons for 'サイト' (Site), 'クイックスタート' (Quick Start), 'リクエスト' (Request), and 'レスポンス' (Response). The main window is split into two panes: 'リクエスト' (Request) on the left and 'レスポンス' (Response) on the right. The request pane shows the following details:

```
GET http://example.jp/4g/4g-011.php?zip=1 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://example.jp/4g/4g-012.html
X-Requested-With: XMLHttpRequest
DNT: 1
Connection: keep-alive
Host: example.jp
```

The response pane shows the following details:

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 05:02:33 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 88
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

{"message":"\u090f5\u4fbf\u756a\u53f7\u304c\u898b\u3064\u304b\u308a\u307e\u305b\u3093:1"}
```

At the bottom, the '履歴' (History) table lists the requests:

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...	ラウンドトリップ...	レスポンス...	検出...	...	...
35	19/01/07 5:02:32	GET	http://example.jp/4g/4g-012.html	200	OK	3 ms	239 bytes	M...	S...	
37	19/01/07 5:02:32	GET	http://example.jp/js/jquery-3.2.1.min.js	200	OK	16 ms	86,659 by...	Low	F...	
39	19/01/07 5:02:32	GET	http://example.jp/4g/4g-011.php?zip=1	200	OK	19 ms	88 bytes	M...		

The bottom status bar shows 'アラート' (Alerts) with counts for various categories and '現在のスキャン' (Current Scan) with counts for various metrics.

## JSONPコールバック関数名によるXSS 4g-016 : JSONPコールバック関数名によるXSS(HTMLからの正常系利用)

### 【ブラウザ】

#### JSONPコールバック関数名によるXSS

- 4g-016 :JSONPコールバック関数名によるXSS(HTMLからの正常系利用)
- 4g-015 :JSONPコールバック関数名によるXSS(攻撃でない直接閲覧)
- 4g-015 :JSONPコールバック関数名によるXSS(XSS攻撃)
- 4g-015a:JSONPコールバック関数名によるXSS(対策版)

```
28 </li>JSONPコールバック関数名によるXSS</li>
```

```
29 </ol>
```

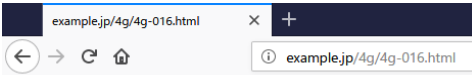
```
30 <li><a href="4g-016.html">4g-016 :JSONPコールバック関数名によるXSS(HTMLからの正常系利用)</a></li>
```

```
31 <li><a href="http://api.example.net/4g/4g-015.php?callback=display_time">4g-015 :JSONPコールバック関数名によるXSS(攻撃でない直接閲覧)</a></li>
```

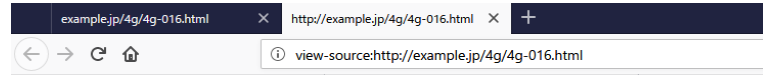
```
32 <li><a href="http://api.example.net/4g/4g-015.php?callback=%3Cscript%3Ealert(document.domain)%3C/script%3E">4g-015 :JSONPコールバック関数名によるXSS(XSS攻撃)</a></li>
```

```
33 <li><a href="http://api.example.net/4g/4g-015a.php?callback=%3Cscript%3Ealert(document.domain)%3C/script%3E">4g-015a:JSONPコールバック関数名によるXSS(対策版)</a></li>
```

```
34 </ol>
```



時刻は14:13です



```
1 <body>
2 <script>
3 function display_time(obj) {
4   var txt = document.createTextNode("時刻は" + obj.time + "です");
5   var p = document.getElementById("time");
6   p.appendChild(txt);
7 }
8 </script>
9 <p id="time"></p>
10 <script src="http://api.example.net/4g/4g-015.php?callback=display_time"></script>
11 </body>
```

### 【サーバ: 4g/4g-016.html】

/var/www/html/4g/4g-016.html - wasbook@example.jp - エディタ - WinSCP

```
<body>
<script>
function display_time(obj) {
  var txt = document.createTextNode("時刻は" + obj.time + "です");
  var p = document.getElementById("time");
  p.appendChild(txt);
}
</script>
<p id="time"></p>
<script src="http://api.example.net/4g/4g-015.php?callback=display_time"></script>
</body>
```

### 【サーバ: 4g/4g-015.php】

/var/www/html/4g/4g-015.php - wasbook@example.jp - エディタ - WinSCP

```
k?php
$callback = $_GET['callback'];
$json = json_encode(array('time' => date('G:i')));
echo "$callback($json);";
```

【ブラウザ→サーバ: リクエスト 4g/4g-016.html → レスポンス】

無題セッション - 20190107-015443 - 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

デフォルトビュー

```
GET http://example.jp/4g/4g-016.html 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/4g/
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Host: example.jp
```

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 05:13:47 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 302
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "12e-56c2a2dea2f18-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge
```

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...	ラウンドトリップ...	レスポンス...	検出...	...	...
41	19/01/07 5:13:46	GET	http://example.jp/4g/4g-016.html	200	OK	8 ms	302 bytes	M...	S...	
42	19/01/07 5:13:46	GET	http://api.example.net/4g/4g-015.php?callback=display_time	200	OK	25 ms	31 bytes	M...		

アラート 0 0 1 3 0 現在のスキャン 0 0 0 0 0 0 0 0 0 0

【ブラウザ→APIサーバ: リクエスト 4g/4g-015.php → レスポンス】

無題セッション - 20190107-015443 - 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

デフォルトビュー

```
GET http://api.example.net/4g/4g-015.php?callback=display_time HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
Accept: */*
Accept-Language: ja,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://example.jp/4g/4g-016.html
DNT: 1
Connection: keep-alive
Host: api.example.net
```

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 05:13:47 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
X-UA-Compatible: IE=edge
display_time({"time": "14:13"})
```

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...	ラウンドトリップ...	レスポンス...	検出...	...	...
41	19/01/07 5:13:46	GET	http://example.jp/4g/4g-016.html	200	OK	8 ms	302 bytes	M...	S...	
42	19/01/07 5:13:46	GET	http://api.example.net/4g/4g-015.php?callback=display_time	200	OK	25 ms	31 bytes	M...		

アラート 0 0 1 3 0 現在のスキャン 0 0 0 0 0 0 0 0 0 0

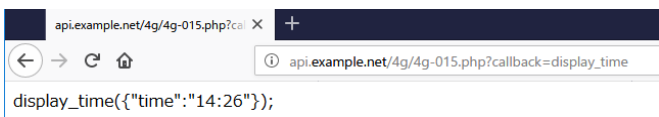
## 4g-015 : JSONPコールバック関数名によるXSS(攻撃でない直接閲覧)

### 【ブラウザ】

- JSONPコールバック関数名によるXSS

- [4g-016 :JSONPコールバック関数名によるXSS\(HTMLからの正常系利用\)](#)
- [4g-015 :JSONPコールバック関数名によるXSS\(攻撃でない直接閲覧\)](#)
- [4g-015 :JSONPコールバック関数名によるXSS\(XSS攻撃\)](#)
- [4g-015a:JSONPコールバック関数名によるXSS\(対策版\)](#)

```
28 </i>JSONPコールバック関数名によるXSS</i>
29 </ol>
30 <li><a href="/4g-016.html">4g-016 :JSONPコールバック関数名によるXSS(HTMLからの正常系利用)</a></li>
31 <li><a href="http://api.example.net/4g/4g-015.php?callback=display_time">4g-015 :JSONPコールバック関数名によるXSS(攻撃でない直接閲覧)</a></li>
32 <li><a href="http://api.example.net/4g/4g-015.php?callback=%3Cscript%3Ealert(document.domain)%3C/script%3E">4g-015 :JSONPコールバック関数名によるXSS(XSS攻撃)</a></li>
33 <li><a href="http://api.example.net/4g/4g-015a.php?callback=%3Cscript%3Ealert(document.domain)%3C/script%3E">4g-015a:JSONPコールバック関数名によるXSS(対策版)</a></li>
34 </ol>
```



### 【サーバ: 4g/4g-015.php】

```
/var/www/html/4g/4g-015.php - wasbook@example.jp - エディタ - WinSCP
k?php
$callback = $_GET['callback'];
$json = json_encode(array('time' => date('G:i')));
echo "$callback($json)";
```

### 【ブラウザ→APIサーバ: リクエスト 4g/4g-015.php → レスポンス】

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...	ラウンドトリップ...	レスポンス...	検出...	...
45	19/01/07 5:26:45	GET	http://api.example.net/4g/4g-015.php?callback=display_time	200	OK	25 ms	31 bytes	M...	

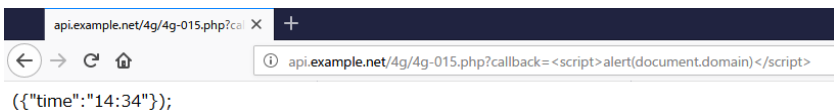
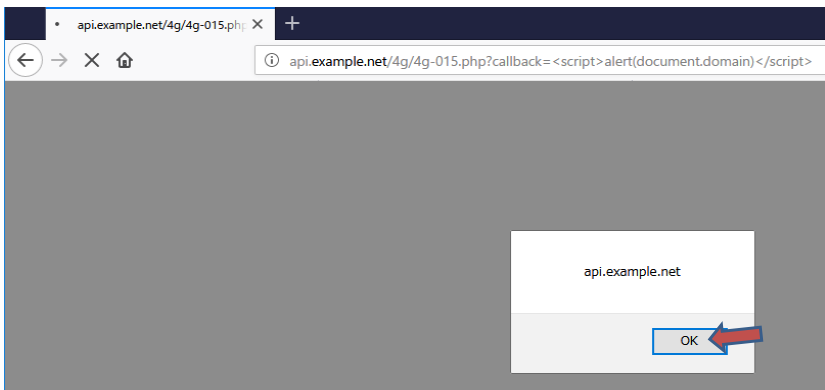
## 4g-015 : JSONPコールバック関数名によるXSS(XSS攻撃)

### 【ブラウザ】

#### • JSONPコールバック関数名によるXSS

1. [4g-016 :JSONPコールバック関数名によるXSS\(HTMLからの正常系利用\)](#)
2. [4g-015 :JSONPコールバック関数名によるXSS\(攻撃でない直接閲覧\)](#)
3. [4g-015 :JSONPコールバック関数名によるXSS\(XSS攻撃\)](#)
4. [4g-015a:JSONPコールバック関数名によるXSS\(対策版\)](#)

```
28 <li>JSONPコールバック関数名によるXSS</li>
29 </ol>
30 <li><a href="4g-016.html">4g-016 :JSONPコールバック関数名によるXSS(HTMLからの正常系利用)</a></li>
31 <li><a href="http://api.example.net/4g/4g-015.php?callback=display_time">4g-015 :JSONPコールバック関数名によるXSS(攻撃でない直接閲覧)</a></li>
32 <li><a href="http://api.example.net/4g/4g-015.php?callback=%3Cscript%3Ealert(document.domain)%3C/script%3E">4g-015 :JSONPコールバック関数名によるXSS(XSS攻撃)</a></li>
33 <li><a href="http://api.example.net/4g/4g-015a.php?callback=%3Cscript%3Ealert(document.domain)%3C/script%3E">4g-015a:JSONPコールバック関数名によるXSS(対策版)</a></li>
34 </ol>
```



### 【サーバ: 4g/4g-015.php】

```
/var/www/html/4g/4g-015.php - wasbook@example.jp - エディタ - WinSCP
k?php
$callback = $_GET['callback'];
$json = json_encode(array('time' => date('G:i')));
echo "$callback($json)";
```

JSONPはJSONではなく、JavaScriptそのままなので、MIMEタイプは text/javascript にすべきですが、指定していないので、デフォルトの text/html を返しますそのため、JavaScriptが実行されてしまいます

【ブラウザ→APIサーバ: リクエスト 4g/4g-015.php → レスポンス】

The screenshot displays the OWASP ZAP 2.7.0 interface. The top toolbar includes buttons for 'サイト' (Site), 'クイックスタート' (Quick Start), 'リクエスト' (Request), and 'レスポンス' (Response). The main area is split into two panes: 'リクエスト' (Request) on the left and 'レスポンス' (Response) on the right. The request pane shows a GET request to `http://api.example.net/4g/4g-015.php?callback=%3Cscript%3Ealert(document.domain)%3C/script%3E`. The response pane shows an HTTP/1.1 200 OK response with headers including `Server: nginx/1.10.3` and `Date: Mon, 07 Jan 2019 05:30:49 GMT`. The response body contains the JavaScript code `<script>alert(document.domain)</script>({"time":"14:30"});`. At the bottom, a table lists the request details.

Id	リクエスト日時	メソッド	URL	ステータスコード	ステータスコー...	ラウンドトリップ...	レスポンス...	検出...	...	...
46	19/01/07 5:30:48	GET	http://api.example.net/4g/4g-015.php?callback=%3Cscript%3Ealert(document.domain)%3C/scrip...	200	OK	35 ms	58 bytes	M...	S...	

アラート 0 0 1 3 0

現在のスキャン 0 0 0 0 0 0 0 0 0 0

## 4g-015 : JSONPコールバック関数名によるXSS(XSS攻撃)

### 【ブラウザ】

- JSONPコールバック関数名によるXSS

- 4g-016 :JSONPコールバック関数名によるXSS(HTMLからの正常系利用)
- 4g-015 :JSONPコールバック関数名によるXSS(攻撃でない直接閲覧)
- 4g-015 :JSONPコールバック関数名によるXSS(XSS攻撃)
- 4g-015a:JSONPコールバック関数名によるXSS(対策版)

```
28 <li>JSONPコールバック関数名によるXSS</li>
29 </ol>
30 <li><a href="4g-016.html">4g-016 :JSONPコールバック関数名によるXSS(HTMLからの正常系利用)</a></li>
31 <li><a href="http://api.example.net/4g/4g-015.php?callback=display_time">4g-015 :JSONPコールバック関数名によるXSS(攻撃でない直接閲覧)</a></li>
32 <li><a href="http://api.example.net/4g/4g-015.php?callback=%3Cscript%3Ealert(document.domain)%3C/script%3E">4g-015 :JSONPコールバック関数名によるXSS(XSS攻撃)</a></li>
33 <li><a href="http://api.example.net/4g/4g-015a.php?callback=%3Cscript%3Ealert(document.domain)%3C/script%3E">4g-015a:JSONPコールバック関数名によるXSS(対策版)</a></li>
34 </ol>
```



### 【サーバ: 4g/4g-015a.php】

```
#!/var/www/html/4g/4g-015a.php - wasbook@example.jp - エディタ - WinSCP
k?php
$callback = $_GET['callback'];
if (preg_match('/\A[\_a-z][\_a-z0-9]*\z/i', $callback) !== 1) {
    header('HTTP/1.1 403 Forbidden');
    die('コールバック関数名が不正です');
}
header('Content-Type: text/javascript; charset=UTF-8');
$json = json_encode(array('time' => date('G:i')));
echo "$callback($json)";
```

コールバック関数名の文字種と文字数を制限しています  
※ 英数字とアンダースコア「\_」のみに制限しても問題ないと思われる。<>を使えなくする。

MIMEタイプを正しく設定する  
text/javascript とすると、レスポンスをHTMLとして解釈しないので、script要素は解釈されず、JavaScriptとしても実行されません

### 【ブラウザ→APIサーバ: リクエスト 4g/4g-015a.php → レスポンス】

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...	ラウンドトリップ...	レスポンス...	検出...	...
51	19/01/07 5:40:11	GET	http://api.example.net/4g/4g-015a.php?callback=%3Cscript%3Ealert(document.domain)%3C/scri...	403	Forbidden	40 ms	42 bytes	Low	