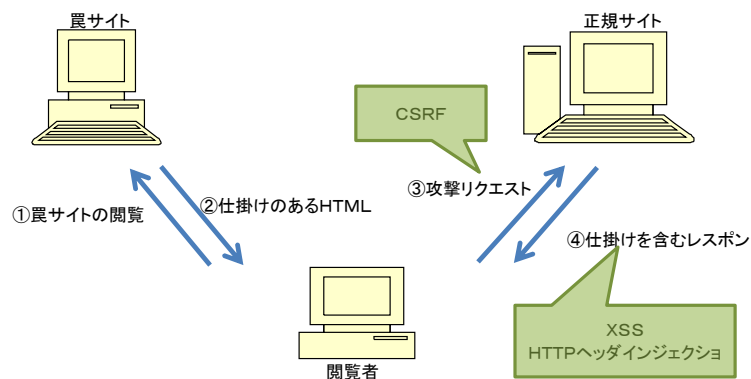


3.2 受動的攻撃と同一生成元ポリシー

サイトをまたがった攻撃



同一オリジンポリシー

上記のようなサイトをまたがった攻撃に対して、JavaScriptなどのクライアントスクリプトからサイトをまたがったアクセスを禁止するセキュリティ上の制限があります。

同一オリジンの条件

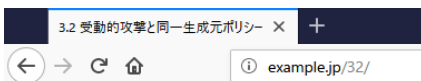
- ① URLのホスト (FQDN) が一致している
- ② スキーム (プロトコル) が一致している
- ③ ポート番号が一致している

JavaScript以外のクロスドメインアクセス

- frame要素とiframe要素 クロスドメインアクセスできるが、JavaScriptによって、クロスドメインのドキュメントにアクセスするのは禁止されている
- img要素のsrc属性 アクセス可能。HTML5のcanvas要素は、同一オリジンポリシー制約、CORS制約を受ける。
- script要素 他サイトからスクリプトを読み込める。読み込んだサイトに対するクッキーが送信され、そのサイトのログイン状態によって、読み込むスクリプトが変化して、元のサイトが影響を受ける場合がある。このケースはJSONPが該当し、JSONPでは公開情報のみを提供するようにすべき。
- CSS アクセス可能。HTMLのlink要素、CSS内からの@import、JavaScriptからのAddImportメソッド Internet ExplorerにはCSSXSSという脆弱性あり。
- form要素のaction属性 アクセス可能。CSRFを引き起こす。

32-001:iframe内データの読み出し

【ブラウザ】

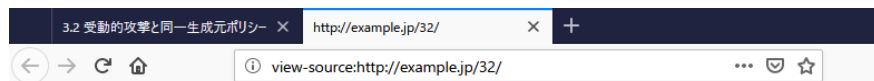


3.2 受動的攻撃と同一生成元ポリシー

1. 32-001:iframe内データの読み出し
2. 32-900:iframe要素を用いた罠

[phpinfo](#)

[ホームに戻る](#)



```
1 <html>
2 <head><title>3.2 受動的攻撃と同一生成元ポリシー</title></head>
3 <body>
4 3.2 受動的攻撃と同一生成元ポリシー
5 <ol>
6 <li><a href="32-001.html">32-001:iframe内データの読み出し</a></li>
7 <li><a href="http://trap.example.com/32/32-900.html">32-900:iframe要素を用いた罠</a></li>
8 </ol>
9 <a href="phpinfo.php">phpinfo</a><br>
10 <a href="/">ホームに戻る</a>
11 </body>
12 </html>
13
```

【サーバ: 32/32-001.html】

```
/var/www/html/32/32-001.html - wasbook@192.168.56.101 - エディタ - WinSCP
<html>
<head><title>フレーム間の読み出し実験</title></head>
<body>
<iframe name="iframe1" width="300" height="80" src="http://example.jp/32/32-002.html">
</iframe><br>
<input type="button" onclick="go()" value="パスワード→">
<script>
function go() {
try {
var x = iframe1.document.form1.passwd.value;
document.getElementById('out').textContent = x;
} catch (e) {
alert(e.message);
}
}
</script>
<span id="out"></span>
</body>
</html>
```

【サーバ: 32/32-002.html】

```
/var/www/html/32/32-002.html - wasbook@192.168.56.101 - エディタ - WinSCP
<body>
<form name="form1">
iframeの内側<br>
パスワード<input type="text" name="passwd" value="password1">
</form>
</body>
```

【ブラウザ→サーバ: リクエスト 32/32-001.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 'リクエスト' pane shows the following details:

```
GET http://example.jp/32/32-001.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/32/
DNT: 1
Connection: keep-alive
Cookie: PHPSESSID=em949q5n1bfhues287ocaehb05
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: Sat, 15 Dec 2018 08:45:26 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 462
Connection: keep-alive
Last-Modified: Thu, 31 May 2018 00:56:58 GMT
ETag: "1ce-56d75f1c9eba5-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<html>
<head><title>フレーム間の読み出し実験</title></head>
<body>
<iframe name="iframe1" width="300" height="80" src="http://example.jp/32/32-002.html"
>
</iframe><br>
<input type="button" onclick="go()" value="パスワード→">
<script>
function go() {
  try {
    var x = iframe1.document.form1.passwd.value;
    document.getElementById("out").textContent = x;
  } catch (e) {
    alert(e.message);
  }
}
</script>
<span id="out"></span>
</body>
</html>
```

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	18/12/15 17:45:25	GET	http://example.jp/32/32-001.html	200	OK	5 ms	462 bytes	Medium		Script

The status bar at the bottom indicates '現在のスキャン' (Current Scan) with various icons and counts.

【ブラウザ→サーバ: リクエスト 32/32-002.html → レスポンス】

無題セッション - 20181215-103244 - 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/32/32-002.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/32/32-001.html
DNT: 1
Connection: keep-alive
Cookie: PHPSESSID=em949q5n1bfnues287ocaehb05
Upgrade-Insecure-Requests: 1
Host: example.jp

HTTP/1.1 200 OK
Server: nginx/1.10.3
Date: Sat, 15 Dec 2018 08:45:26 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 130
Connection: keep-alive
Last-Modified: Mon, 14 May 2018 13:08:17 GMT
ETag: "82-56c2a2de3c670-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

```
<body>  
<form name="form1">  
iframeの内側<br>  
パスワード<input type="text" name="passwd" value="password1">  
</form>  
</body>
```

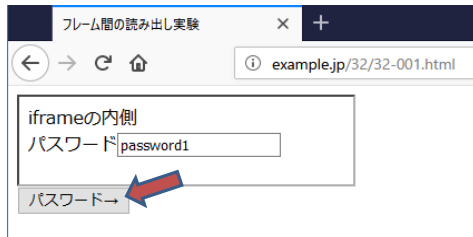
履歴 検索 アラート アウトプット +

フィルタ: オフ エクスポート

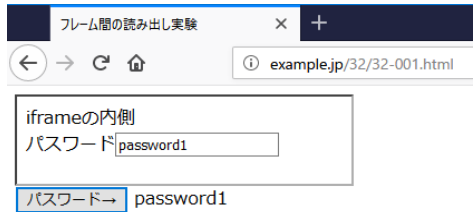
Id	リクエスト日時	メソッド	URL	ステータスコード	ステータスコード説明	ラウンドトリップタイム	レスポンスボディサイズ	検出アラート	ノート	タグ
10	18/12/15 17:45:25	GET	http://example.jp/32/32-002.html	200	OK	6 ms	130 bytes	Medium		Form

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】



```
view-source:http://example.jp/32/32-001.html
1 <html>
2 <head><title>フレーム間の読み出し実験</title></head>
3 <body>
4 <iframe name="iframe1" width="300" height="80" src="http://example.jp/32/32-002.html">
5 </iframe><br>
6 <input type="button" onclick="go()" value="パスワード→">
7 <script>
8 function go() {
9   try {
10    var x = iframe1.document.form1.passwd.value;
11    document.getElementById('out').textContent = x;
12  } catch (e) {
13    alert(e.message);
14  }
15 }
16 </script>
17 <span id="out"></span>
18 </body>
19 </html>
20
```



```
view-source:http://example.jp/32/32-001.html
1 <html>
2 <head><title>フレーム間の読み出し実験</title></head>
3 <body>
4 <iframe name="iframe1" width="300" height="80" src="http://example.jp/32/32-002.html">
5 </iframe><br>
6 <input type="button" onclick="go()" value="パスワード→">
7 <script>
8 function go() {
9   try {
10    var x = iframe1.document.form1.passwd.value;
11    document.getElementById('out').textContent = x;
12  } catch (e) {
13    alert(e.message);
14  }
15 }
16 </script>
17 <span id="out"></span>
18 </body>
19 </html>
20
```

32-900:iframe要素を用いた罠

【ブラウザ】

3.2 受動的攻撃と同一生成元ポリシー

- 32-001:iframe内データの読み出し
- 32-900:iframe要素を用いた罠

[phpinfo](#)
[ホームに戻る](#)

```
1 <html>
2 <head><title>3.2 受動的攻撃と同一生成元ポリシー</title></head>
3 <body>
4 3.2 受動的攻撃と同一生成元ポリシー
5 <ol>
6 <li><a href="/32-001.html">32-001:iframe内データの読み出し</a></li>
7 <li><a href="/http://trap.example.com/32/32-900.html">32-900:iframe要素を用いた罠</a></li>
8 </ol>
9 <a href="/phpinfo.php">phpinfo</a><br>
10 <a href="/">ホームに戻る</a>
11 </body>
12 </html>
13
```

【サーバ: 32/32-900.html】

```
<html>
<head><title>フレーム間の読み出し実験</title></head>
<body>
<iframe name="iframe1" width="300" height="80" src="http://example.jp/32/32-002.html">
</iframe><br>
<input type="button" onclick="go()" value="パスワード→">
<script>
function go() {
  try {
    var x = iframe1.document.form1.passwd.value;
    document.getElementById('out').textContent = x;
  } catch (e) {
    alert(e.message);
  }
}
</script>
<span id="out"></span>
</body>
</html>
```

【ブラウザ→罾サーバ: リクエスト trap.example.com/32/32-900.php → レスポンス】

The screenshot shows 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 'コンテキスト' (Context) pane on the far left shows the site 'example.jp/32/'.

Request:

```
GET http://trap.example.com/32/32-900.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/32/
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: Sat, 15 Dec 2018 12:23:58 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 462
Connection: keep-alive
Last-Modified: Thu, 31 May 2018 00:57:48 GMT
ETag: "1ce-56d75f4bb7a41-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
X-UA-Compatible: IE=edge

<html>
<head><title>フレーム間の読み出し実験</title></head>
<body>
<iframe name="iframe1" width="300" height="80" src="http://example.jp/32/32-002.html">
</iframe><br>
<input type="button" onclick="go()" value="/パスワード→">
<script>
function go(){
try {
var x = iframe1.document.form1.passwd.value;
document.getElementById("out").textContent = x;
} catch (e) {
alert(e.message);
}
}
</script>
<span id="out"></span>
</body>
</html>
```

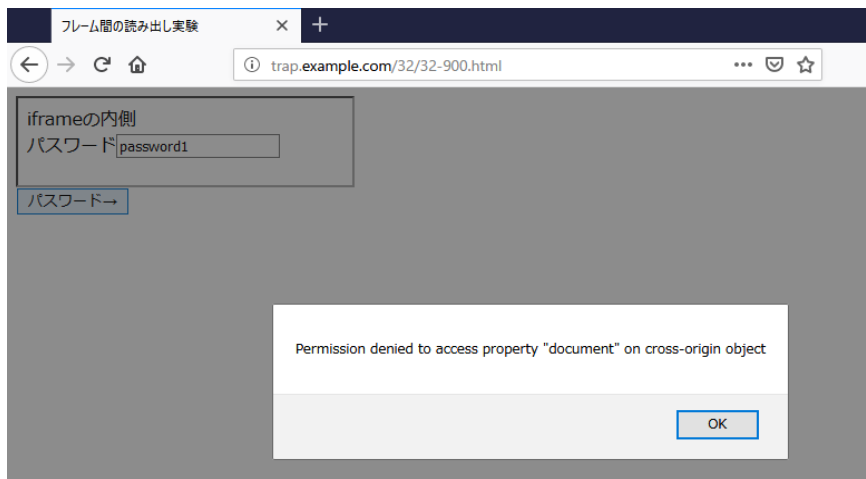
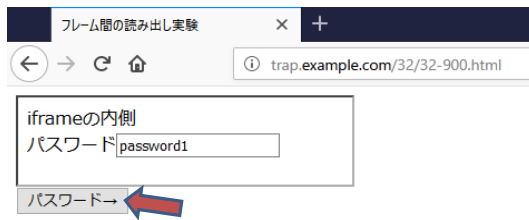
The response body contains an iframesrc and a JavaScript function named go(). The iframesrc is highlighted with a dashed red box. The JavaScript function go() is also highlighted with a dashed red box. The function attempts to access the password field of a form in an iframe and display its value in a span with id="out".

At the bottom of the interface, there is a table showing the request and response 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
20	18/12/15 21:23:56	GET	http://trap.example.com/32/32-900.html	200	OK	19 ms	462 bytes	Medium		Script

Below the table, there are several status indicators: アラート 0, 1, 2, 0. At the bottom right, there is a status bar showing '現在のスキャン' (Current Scan) with various icons and counts: 0, 0, 0, 0, 0, 0, 0.

【ブラウザ】



偽サイトのiframeから、クロスサイトの正規サイトの内容は参照を拒否される。