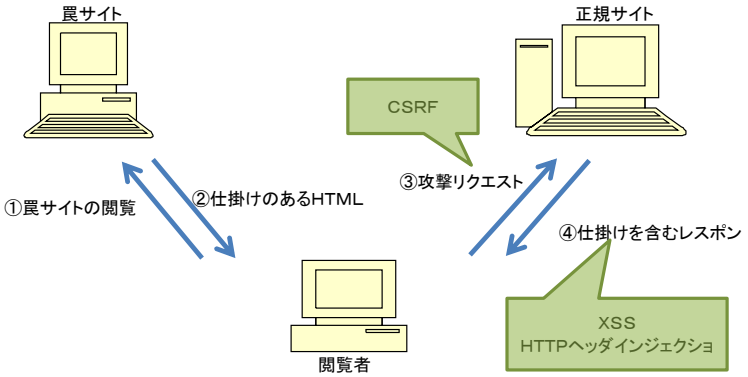
3.2 受動的攻撃と同一生成元ポリシー

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

①URL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 Internat Explore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】

【サーバ： 32/32-001.html 】

/var/www/html/32/32-001.html - wasbook@192.168.56.101 - エディタ - WinSCP

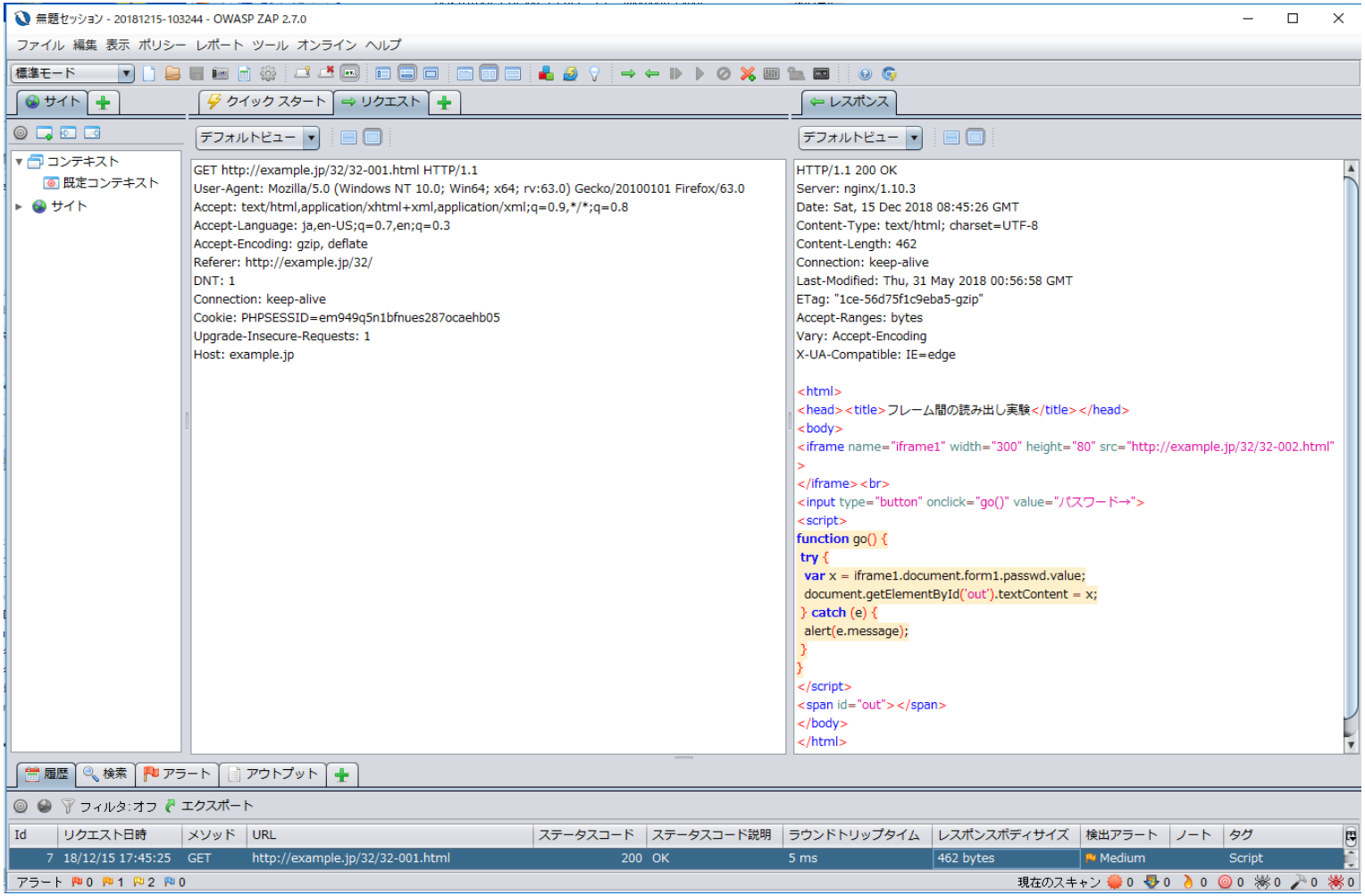
```
<html>
<head><title>フレーム間の読み出し実験</title></head>
<body>
<iframe name="iframe1" width="300" height="80" src="http://example.jp/32/32-002.html">
</iframe><br>
<input type="button" onclick="go()" value="パスワード→">
<script>
function go() {
 try {
  var x = iframe1.document.form1.passwd.value;
  document.getElementById('out').textContent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

【サーバ： 32/32-002.html 】

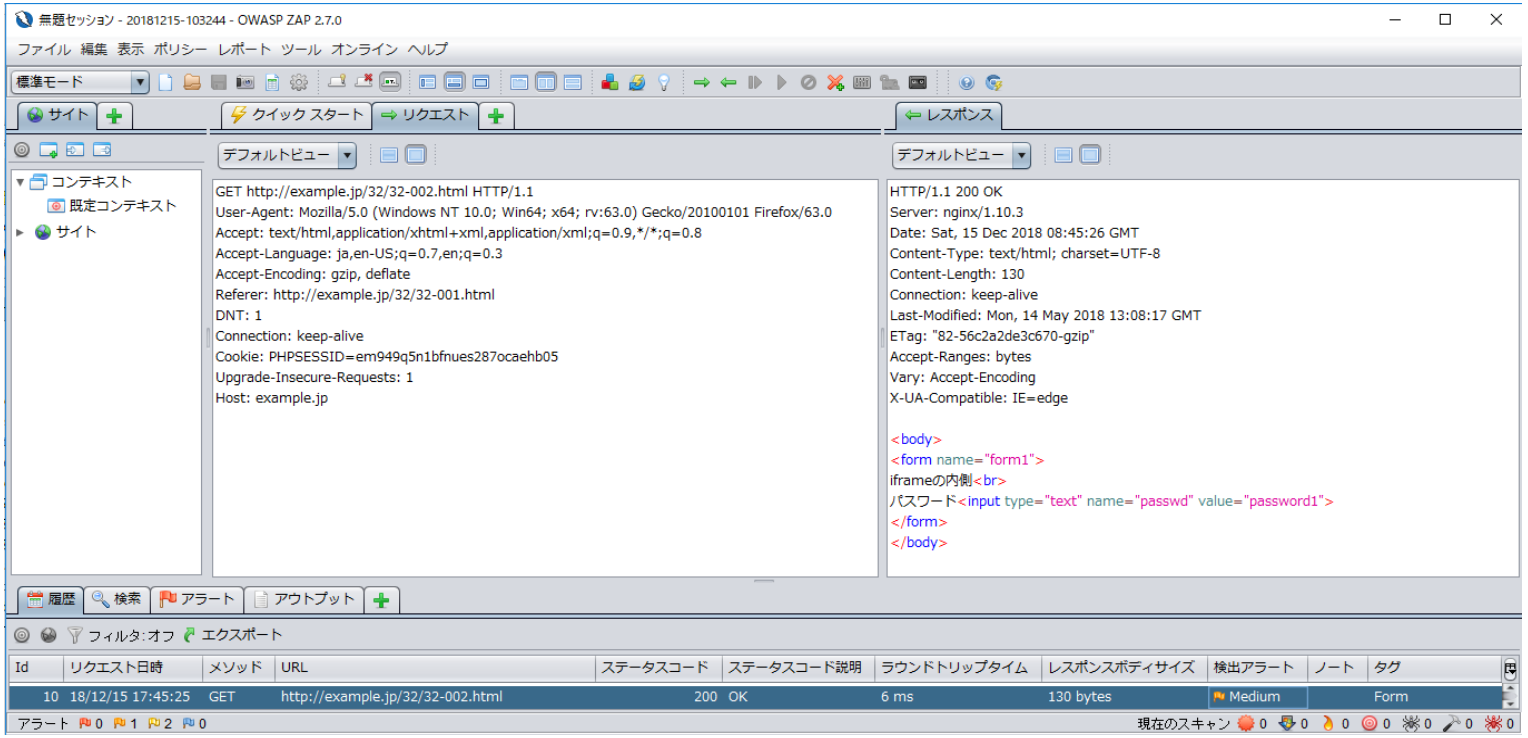
/var/www/html/32/32-002.html - wasbook@192.168.56.101 - エディタ - WinSCP

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

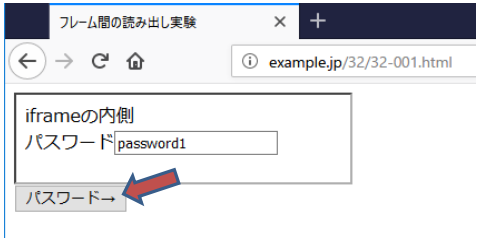
【ブラウザ→サーバ：リクエスト 32/32-001.html → レスポンス 】



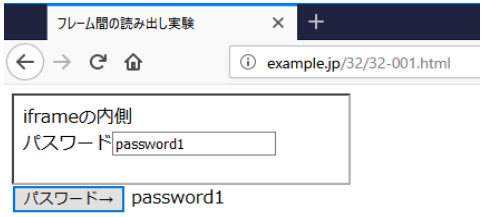
【ブラウザ→サーバ：リクエスト 32/32-002.html → レスポンス 】



【ブラウザ】


フレーム間の読み出し実験
example.jp/32/32-001.html
iframeの内側
パスワード password1
パスワード→


フレーム間の読み出し実験
http://example.jp/32/32-001.html
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>
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
18 </body>
19 </html>
20


フレーム間の読み出し実験
example.jp/32/32-001.html
iframeの内側
パスワード password1
パスワード→ password1

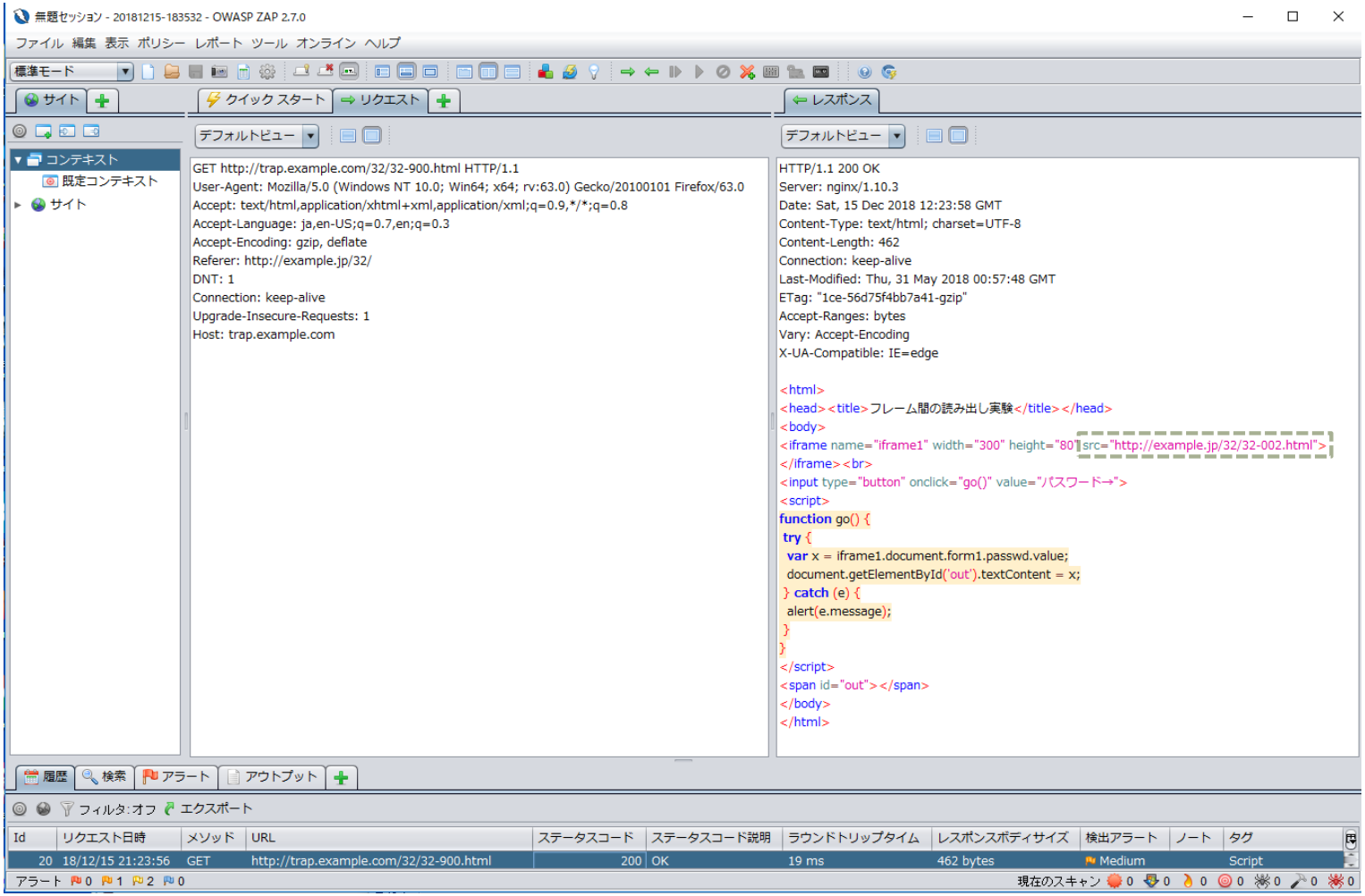

フレーム間の読み出し実験
http://example.jp/32/32-001.html
view-source:http://example.jp/32/32-001.html
1 <html>
2 <head><title>フレーム間の読み出し実験</title></head>
3 <body>
4 <iframe name="iframe1" width="300" height="80" src="http://example.jp/32/32-002.html">
5 </iframe>
6 <input type="button" onclick="go()" value="パスワード→">
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
18 </body>
19 </html>
20

32-900:iframe要素を用いた罠

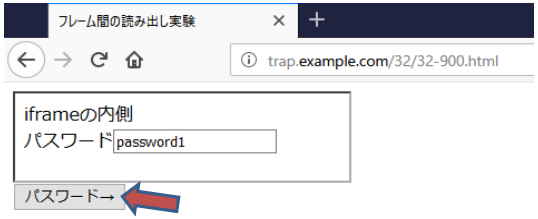
【ブラウザ】

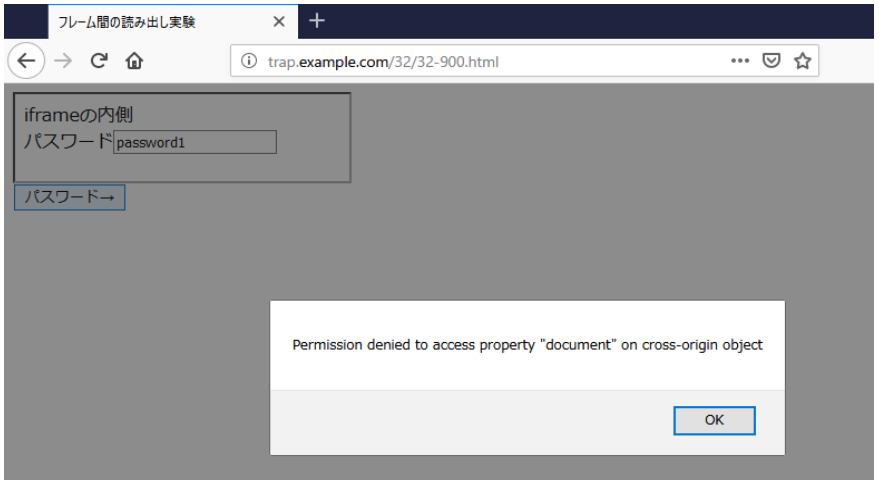
【サーバ： 32/32-900.html 】

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



【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。