

脆弱性体験学習ツール

AppGoat

カスタマイズマニュアル

2016 年 7 月

改定履歴

更新日	更新内容
2016年7月29日	第1版 発行

内容

1.	はじめに	5
1.1.	本マニュアルについて	5
1.2.	カスタマイズ可能な項目	5
1.3.	カスタマイズ上の注意	8
1.4.	用語	8
2.	シナリオフォルダ構成	9
3.	シナリオ編集手順	10
3.1.	テーマタイトル	10
3.2.	ステージタイトル	11
3.3.	コンテンツ文章の記述箇所	12
3.4.	コンテンツ文章の編集ルール	13
4.	ヒント編集手順	20
4.1.	ヒントボタン	20
4.2.	ヒント内容	21
4.3.	ヒント内容の記述箇所	21
4.4.	ヒント文章の編集ルール	22
5.	称号名の設定手順	24
5.1.	称号名の設定	24
5.1.1.	①称号名の設定	26
5.1.2.	②ツールチップの変更	26
5.1.3.	③学習の進め方での称号の説明	27
6.	カスタマイズ実施例	28
6.1.	画像の変更(追加)	28
6.2.	ヒントの追加	33
6.3.	説明用コンテンツの記載内容修正(学習環境に合わせたファイルパスの追記)	38

7. 終わりに	42
補足資料	43

1. はじめに

1.1. 本マニュアルについて

本マニュアルは管理者向けに、「脆弱性体験学習ツール AppGoat」（以降、実習ツール）の画面に表示されている文言をカスタマイズ（修正）する方法を説明したものです。

1.2. カスタマイズ可能な項目

実習ツールのカスタマイズ可能な項目は以下になります。

表 1 実習ツールで可能なカスタマイズ項目と本マニュアルでの記載箇所

カスタマイズ項目	本マニュアルでの記載箇所
画像の変更・追加	2 シナリオフォルダ構成 6.1 画像の変更（追加）
シナリオの削除	2 シナリオフォルダ構成
テーマタイトルの変更	2 シナリオフォルダ構成 3.1 テーマタイトル
ステージタイトルの変更	2 シナリオフォルダ構成 3.2 ステージタイトル
コンテンツ文章の変更	2 シナリオフォルダ構成 3.3 コンテンツ文章の記述箇所 3.4 コンテンツ文章の編集ルール 6.3 説明用コンテンツの記載内容修正（学習環境に合わせたファイルパスの追記）

カスタマイズ項目	本マニュアルでの記載箇所
ヒントの追加・削除・変更	2 シナリオフォルダ構成 4 ヒント編集手順 6.2 ヒントの追加
称号名の設定	2 シナリオフォルダ構成 5.1 称号名の設定

IPA 脆弱性体験学習ツール AppGoat

総合メニュー 学習を進める前に 学習環境へ 学習者の管理 演習セットの設定 FAQ 利用者マニュアル AppGoatの終了方法 ログアウト ログインユーザ :

テーマ一覧

表示中のページ

- 基礎
 - クロスサイト・スクリプティング
 - Level1
 - 脆弱性の概要および発見演習
 - Level2
 - 脆弱性の概要および発見演習

脆弱性の概要および発見演習

脆弱性の発見手法 (1/2)

ステージタイトル

基本的な検査方法

HTMLで出力する時に「<」を「<」に置換するなど、特別な意味を持つ文字を、特別な意味を持たない表記文字に置換することをエスケープ処理と言います。もし、受け取った入力データを、エスケープ処理を行わずに画面に出力している箇所があれば、クロスサイト・スクリプティングの脆弱性になります。検査方法の一例を以下に示します。

テーマタイトル (URL中にあるクエリストリング)に「><hr>」を入れて、リクエスト送信

- 出力画面で右クリックして、メニューから「ソースコードの表示」
- エスケープされずに出力されていた場合、脆弱性あり

画像の変更(追加)

```
http://example.com/test.php?name='><hr>
```

```
<body>
<h4 class="name">><hr></h4>
</body>
```

ウェブサーバ

図 1: 脆弱性がある場合

図 1 カスタマイズ項目とリンク (タイトル・画像編)

コンテンツ文章の記述箇所

次の手順にしたがって、攻撃を行ってみましょう。

- 投稿後の挙動やHTMLソースの確認などから、アンケートページに存在する脆弱性のある箇所を探します。
- アンケートページの内容を書き換えるスクリプトを作成します。値を書き換えるスクリプトが分からない場合はヒント1を参照しましょう。
- ②で作成したスクリプトを含む頁のリンクを作成し、【掲示板】に貼り付けます。
- 頁のリンクをクリックすることによって、内容が書き換えられたアンケートページが表示されることを確認します。

疑似攻撃が難しい場合は、ヒントを参照してください。

URL、手順、ヒントに記載するURLの表記法

次のページでは脆弱性の影響を見ていきます。

ヒント編集手順

ヒント1

- 値を書き換えるスクリプトは、「document.getElementById("ID名").innerHTML=変更する値」です。

次のヒント

閉じる

図 2 カスタマイズ項目とリンク（コンテンツ文章偏）

IPA 脆弱性体験学習ツール AppGoat

総合メニュー 学習を進める前に 学習環境へ 学習者の管理 演習セットの設定 FAQ 利用者マニュアル AppGoatの終了方法 ログアウト ログインユーザ :

学習状況表示

1つ前のページに戻る

ログインID yamada
 氏名 山田
 所属 開発部
 演習セット 全演習
 ログイン状況 未ログイン
 称号 **入門者** (現在 0 問正答しています。あと 14 問)

称号名の設定手順

称号とは
 ・習熟度テストの全体の正答率を基に以下の名称を付与しています。
 - 卒業 (正答率: 100%)
 - 上級者 (正答率: 80%以上)
 - 中級者 (正答率: 60%以上)
 - 初級者 (正答率: 40%以上)
 - 初心者 (正答率: 20%以上)
 - 入門者 (正答率: 20%未満)

管理者向け...ウェブサイトの管理者
 開発者向け...ウェブアプリケーション開発者

種別	脆弱性	学習対象		正答率(%)
		管理者向け	開発者向け	
				-
				-
				-
	入力情報の漏えい(反射型)			-

図 3 カスタマイズ項目とリンク（称号偏）

1.3. カスタマイズ上の注意

コンテンツ内では HTML の文法や AppGoat 上で作成されたクラスが多用されています。表 1 に含まれないカスタマイズなど、修正内容によっては画面レイアウトが崩れる、演習が動かなくなる、想定外の脆弱性が生まれる、といったことが懸念されます。これらは全て修正者の自己責任で行ってください。

カスタマイズ項目の中にはアカウント作成時にのみ反映されるものが存在します。その場合、学習者のフォルダに直接反映や、アカウントの作り直しが必要になります。コンテンツのカスタマイズはアカウントの作成前に行うことを推奨します。

1.4. 用語

表 2 用語一覧

用語	意味
シナリオ	実習ツールの学習単位。複数のステージで構成される。
シナリオ番号	種別、カテゴリ、レベル、演習によって各シナリオに割り振った番号。
シナリオ ID	各シナリオに割り振った識別子 (ID)。Scenario + シナリオ番号の形式で記述され、フォルダ名や呼び出し時に用いる。
テーマ	シナリオに任意の名前をつけたもの。
ステージ	テーマを分割したもの。複数のページで構成される。
ページ	テーマを構成する最小単位。
コンテンツ	ページ内に表示する文章や画像。
ヒント	演習問題において、学習者を解答に導く文言。表示非表示の切り替えが可能。
言語ファイル	実習ツールのコンテンツ (文字列) を各言語で書いたファイル。これを複数作成することで多言語化を行う。

2. シナリオフォルダ構成

ここでは、シナリオの編集や削除手順について説明します。

シナリオフォルダは「AppGoat01¥IPATool¥Scenarios¥Web」および

「AppGoat01¥IPATool¥Users¥アカウント名¥Web」に格納されている、1つ1つのシナリオのファイルを格納しているフォルダで、それぞれ図 4 のような構成になっています。

「AppGoat01¥IPATool¥Scenarios¥Web」のシナリオフォルダは学習者アカウントを作成するときに「AppGoat01¥IPATool¥Users¥アカウント名¥Web」にコピーするため、全体に影響するものとなります。「AppGoat01¥IPATool¥Users¥アカウント名¥Web」のシナリオフォルダはログイン時に参照するフォルダでその学習者に影響するものとなります。

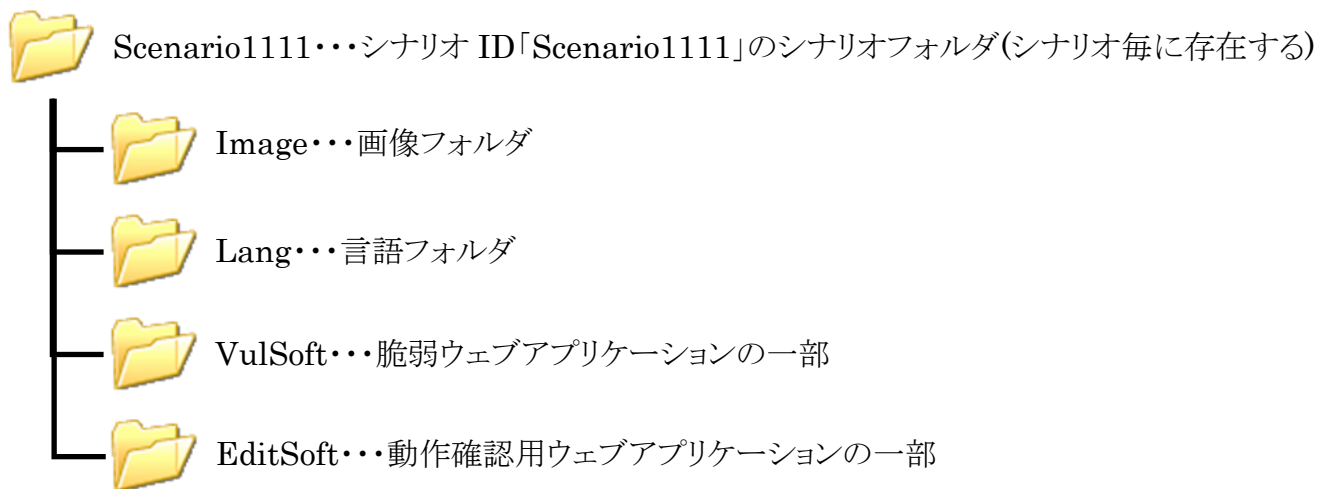


図 4 シナリオフォルダ構成

このうち、シナリオの編集をする場合は主に、言語フォルダにある言語ファイル(ja.txt)を編集します。画像を変更したい場合には画像フォルダのファイルを変更します。

シナリオの削除を行う場合には、対象のシナリオフォルダを削除します。削除したフォルダの中身は復元できませんので、削除する場合にはバックアップを取るよう to してください。削除したシナリオをもとに戻すには、元のシナリオフォルダと同じ名前を、削除したフォルダをもとに戻します。

3. シナリオ編集手順

ここでは、シナリオのコンテンツ文章（文言）の編集方法について説明します。コンテンツ文章は全て言語フォルダ内の言語ファイルに保存されています。言語ファイルは JSON という形式を使用しているため、記述の仕方を誤ると、テーマのコンテンツが表示されなくなってしまうことに注意が必要です。編集内容は変更後に作成したアカウントに反映が行われます。変更前から存在するアカウントへ反映するには、学習管理画面で対象アカウントへの「学習状況の全初期化」の実行が必要です。詳細は「6.2 ヒントの追加」の手順⑤をご覧ください。

※特定の文字の前の¥について

「”」～「”」を編集するため、「”」が文字なのか、終わりを表すのか区別するために文字の場合には「”」の前に¥の文字を記載する必要があります。このように、特定の文字の前には¥の文字の記載が必要です。

3.1. テーマタイトル

テーマタイトルは図 5 の赤い丸で囲んだ部分に相当します。

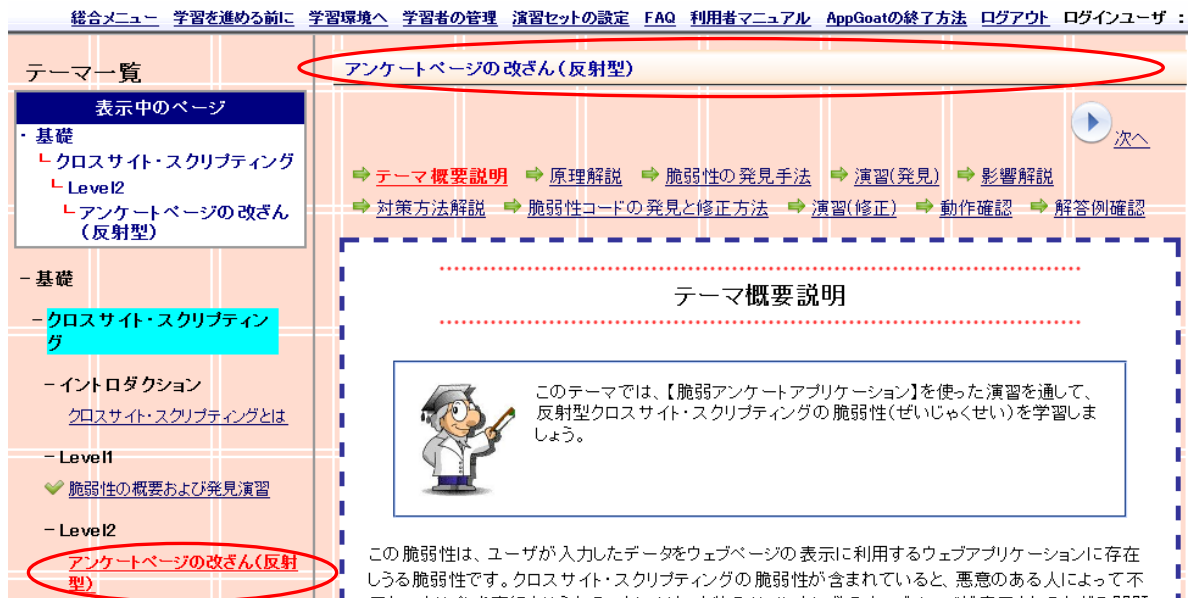


図 5 テーマタイトルの位置

テーマタイトルを変更したい場合は、シナリオ言語ファイルの「scenario_title」を編集します。

```
{
  "scenario_title" : "アンケートページの改ざん(反射型)",
  ...
}
```

図 6 テーマタイトル

3.2. ステージタイトル

ステージタイトルは図 7 の赤い丸で囲んだ部分に相当します。



図 7 ステージタイトルの位置

ステージタイトルを変更したい場合は、シナリオ言語ファイルの「stage_title」を編集します。

```

...
"stage4" : {
  "stage_title" : "演習(発見)",
...

```

図 8 ステージタイトルの例

3.3. コンテンツ文章の記述箇所

テーマは1つ以上のステージ(stage1~stage[n])で構成され、ステージは1つ以上のページ(page1~page[n])で構成されています。HTMLコンテンツは、ページ単位に用意され、HTMLコンテンツを編集する場合は、ページの「contents」という項目を編集します。

```
...
},
"stage3" : {
  "stage_title" : "脆弱性の発見手法",
  "page1" : {
    "contents" : "<h3>基本的な検査方法</h3>\n
                  <p class=\"%normal%\">HTMLで出力する時に「&lt;」を「&amp;lt;」に置換するな
                  ど、特別な意味を持つ文字を、特別な意味を持たない表記文字に置換することをエスケープ処理と言
                  います。<br />\n
                  もし、…
                  ~中略~
                  <p class=\"%normal%\">入力パラメータ全てに対して、このような検査を行い、水
                  平な線が引かれるパラメータがあれば、そこにクロスサイト・スクリプティングの脆弱性があります。
                  </p>\n
                  <p class=\"%picture%\"><img alt=\"%図&nbsp;3 : &lt;hr&gt;により引かれた水平
                  な線%\" width=\"%500%\" height=\"%100%\" src=\"%../Scenario1111/Image/image004.png%\" /></p>\n
                  <p class=\"%caption%\">図&nbsp;3 : &lt;hr&gt;により引かれた水平な線</p>\n
                  "
                },
  "page2" : {
    "contents" : "<h3>応用的な検査方法</h3>\n
                  <p class=\"%normal%\">「基本的な検査方法」では…
                  ~中略~
                  <p class=\"%normal%\">では次に、ここまで学習したことをふまえて、クロスサイ
                  ト・スクリプティングの脆弱性検査を行ってみましょう。</p>\n
                  "
                }
  }
},
...
```

ステージ番号が記載されている。

ページ番号が記載されている。

ページの contents のダブルクォート「」~「」に囲まれた部分を編集する。

図 9 HTMLコンテンツの記述箇所

3.4. コンテンツ文章の編集ルール

コンテンツ文章の編集にはいくつかのルールがあります。以下に示す記述ルールにしたがい編集します。

表 3 コンテンツ文章編集ルール一覧

分類	言語ファイルの修正箇所	目的	記載コード
改行	<code>~</code>外	HTML ソース	¥n
	<code>~</code>外	HTML コンテンツ	
	<code>~</code>内	HTML ソース HTML コンテンツ	¥n
ダブルクォート	<code>~</code>外	HTML ソース	¥"
	<code>~</code>外	HTML コンテンツ	"
	<code>~</code>内	HTML ソース HTML コンテンツ	¥" または "
シングルクォート	<code>~</code>外	HTML ソース	'
	<code>~</code>外	HTML コンテンツ	'
	<code>~</code>内	HTML コンテンツ	' または '
円マーク	<code>~</code>外 <code>~</code>内	HTML コンテンツ	¥¥
「<」 および 「>」	<code>~</code>外 <code>~</code>内	HTML ソース	< および >
	<code>~</code>外 <code>~</code>内	HTML コンテンツ	< および >

改行

JSON 形式データでは改行は特に意識しませんが、HTML ソース上で改行を表すために、明示的に「`\n`」を用います。文章が長くなったときは改行を行い、ブラウザからソースを表示した際の可読性を向上させましょう。HTML コンテンツとして改行を行うには`<code>~</code>`外は明示的に「`
`」を用います。`<code>~</code>`内では明示的に「`\n`」を用います。また、言語ファイルの`<p>~</p>`内の文章は、次の`<p>~</p>`内の文章との間に行間が入る設定にしています。

```
"contents" : "<h3>基本的な検査方法</h3>\n\n    <p class=\"%normal%\">HTML で出力する時に「&lt;」を「&amp;lt;」に置換するな  
    ど、特別な意味を持つ文 1 行で改行される。ない表記文字に置換することをエスケープ処理と言  
    います。<br />\n\n    もし、受け取った入力データを、エスケープ処理を行わずに画面に出力している  
    箇所があれば、クロスサイト・スクリプティングの脆弱性になります。</p>\n\n    <p class=\"%normal%\">検査方法の一例を以下に示します。</p>\n\n    ...  
    }
```

この場合、言語ファイルの見た目上は複数行でも、HTML ソースとして出力されると 1 行として扱われ、ここで改行される。

図 10 改行の例

⇒ テーマ概要説明 ⇒ 原理解説 ⇒ **脆弱性の発見手法** ⇒ 演習(発見) ⇒ 影響解説 ⇒ 動作確認

脆弱性の発見手法 (1/2)

基本的な検査方法

HTMLで出力する時に「<」を「<」に置換するなど、特別な意味を持つ記文字に置換することをエスケープ処理と言います。
により改行される。

もし、受け取った入力データを、エスケープ処理を行わずに画面に出力している箇所があれば、クロスサイト・スクリプティングの脆弱性になります。<p>~</p>により行間が入る。

検査方法の一例を以下に示します。

- ① 画面の入力ボックス(またはURL中にあるクエリストリング)に「<」
を入れて、リクエスト送信
- ② 出力画面で右クリックして、メニューから「ソースコードの表示」を選択
- ③ エスケープされずに出力されていた場合、脆弱性あり

図 11 改行の例

ダブルクォート

ダブルクォートは JSON 形式の引用符を表すため、ダブルクォートの前に¥の文字を記載します。ただし、HTML コンテンツとしてダブルクォートを表示する場合は HTML 特殊文字「"」を使用します。また、ソースコードを表す<code>~</code>内にダブルクォートを使用する場合にもダブルクォートの前に¥の文字を記載するか、HTML 特殊文字「"」を使用します。

```
...
    <p class="normal">外部に出力している箇所において入力パラメータを受け取
    った後、出力文字列にエスケープ処理をせず、そのまま出力している箇所があれば、クロスサイト・スクリプティ
    ングの脆弱性になります。<br />¥n タグの属性に使用するダブルクォートは、ダブルクォート
    PHP では外部に出力している箇所の前に¥の文字を記載する。
    $_GET["name"]などが使われている箇所を確認しましょう。</p>¥n
...
HTML コンテンツとして表示する場合は
「&quot;」を使用する。
```

図 12 ダブルクォートの例 1

```
...
    <pre><code>¥n
public function login()¥n
...
$stmt = $db->prepare("SELECT * FROM user WHERE id = ? AND password = ?");
...
</code></pre>¥n
...
```

ソースコード中でダブルクォートを表す場合は「¥"」か「"」のどちらでもよい。

図 13 ダブルクォートの例 2

シングルクオート

シングルクオートは JSON 形式の引用符ではないため、シングルクオートの前に¥の文字の記載は必要ありませんが、HTML コンテンツとして表示する場合には「'」を使用します。ただしソースコードを表す<code>~</code>内にはそのまま「'」を使用することもできます。

```
...
    <p>たとえば、&lt;script&gt;alert(&#39;Hello&#39;);&lt;/script&gt;という文字列を出力
    する場合には、エスケープ処理を施して
    &amp;lt;script&amp;gt;alert(&amp;#39;Hello&amp;#39;);&lt;/script&gt;という文字列を使用す。</p>
...

```

HTML コンテンツとして表示する場合は「'」を使用する。

図 14 シングルクオートの例1

```
...
<pre><code>¥n
public function proc_send() ¥n
...
if(!preg_match('/¥n|¥r/' . $param[self::FROM])) {¥n
...
</code></pre>¥n
...

```

ソースコードの箇所(<code>~</code>内)には「'」を使用してもよい。

図 15 シングルクオートの例 2

また、タグの属性値の引用符としてシングルクオートは使用せず、ダブルクオートを使用してください。

間違った例：

正しい例：

円マーク

円 (¥) マークを表示するには、円マークの前に¥の文字の記載が必要なため、円マーク 1 つを表すのに「¥¥」とします。ソースコード中の円マークも同じです。ただし、Faq の HTML ファイルはフレームワークで処理されないためそのまま「¥」を用います。

```
...
        <ol>¥n
            <li class="order1¥">商品管理ページの変更後ファイル名入力欄に、以下のフ
            ファイル名を入力してみましょう。<br>「example.txt &amp; dir /b c:¥¥」 </li>¥n
        </ol>¥n
...

```

円マークの前に¥の文字を記載して「¥¥」にする。

図 16 円マークの例1

```
...
<pre><code>¥n
public function proc_send() ¥n
...
if(!preg_match('/¥¥n|¥¥r/ ', $param[self::FROM])) {¥n
...
</code></pre>¥n
...

```

ソースコードで「¥n」を文字として表す
場合、「¥¥n」にする

図 17 円マークの例2

「<」および「>」

半角記号「<」と「>」は HTML におけるタグ文字にあたるため、そのまま記述すると、デザインが崩れることがあります。「<」と「>」はそれぞれ『<』と『>』にします。これは<code>〜</code>内のソースコードの部分でも同じです。

```
...
      <dd>たとえば、&lt;script&gt;alert(&#39;Hello&#39;)&lt;/script&gt;という文
      字列を出力する場合には、エスケープ処理を施して
&amp;lt;script&gt;alert(&#39;Hello&#39;)&lt;/script&gt;とします。</dd>¥n
...
HTML コンテンツとして表示する場合は「&lt;」を使用する。
```

図 18 「<」および「>」の例

4. ヒント編集手順

ここでは、演習のヒント文章（文言）の編集方法について説明します。ヒント文章は全て言語フォルダ内の言語ファイルに保存されています。言語ファイルは JSON という形式を使用しているため、記述の仕方を誤ると、テーマのコンテンツが表示されなくなってしまうことに注意が必要です。

4.1. ヒントボタン

ヒントボタンは図 19 の赤い丸で囲んだ部分に相当します。

図 19 ヒントボタンの位置

4.2. ヒント内容

ヒント内容は図 20 の赤い丸で囲んだ部分に相当します。

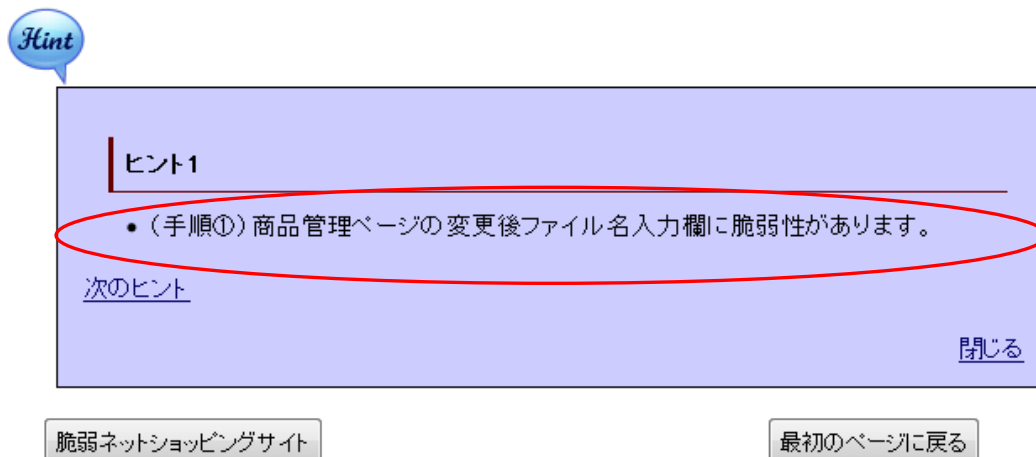


図 20 ヒント内容の位置

ヒント内容を変更したい場合は、シナリオ言語ファイルの「hint[n]」を編集します。

```
...
  "contents" : "...",
  "hint1" : "<ul><li> (手順①) 商品管理ページの変更後ファイル名入力欄に脆弱性があります。
</li></ul>",
  ...
...
```

図 21 ヒント内容

4.3. ヒント内容の記述箇所

ヒントはステージ内でコンテンツと演習アプリケーションの間に位置します。ヒントは1つ以上のヒント項目（hint1～hint[n]）で構成されています。ヒントは演習単位に用意され、ヒントを編集する場合は、ページの「hint[n]」という項目を編集します。ヒント項目は必要に応じて、増やしたり減らしたりできます。

```

...
    "contents" : "...",
    "hint1" : "<ul><li> (手順①) 商品管理ページの変更後ファイル名入力欄に脆弱性があります。
</li></ul>",
    "hint2" : "<ul><li> (手順②) 「Cドライブ直下のファイル名とフォルダ名の一覧を取得する OS
コマンド」は「dir /b c:&yen;」です。</li></ul>",
    "vulsoft1" : {
        "title" : "脆弱ネットショッピングサイト",
        "src" : "VulSoft/netshopping.php"
    }
}

```

図 22 ヒント内容の記述箇所

4.4. ヒント文章の編集ルール

ヒント文章の編集にはいくつかのルールがあります。以下に示す記述ルールに従い編集します。

表 4 ヒント文章の編集ルール

分類	目的	記載コード
ダブルクォート	HTML ソース	¥"
	HTML コンテンツ	"
シングルクォート	HTML コンテンツ	'
円マーク	HTML コンテンツ	\

ダブルクォート

ヒント内容中にダブルクォートをコードとして表示する場合は HTML 特殊文字「"」を使用します。

```
...
"hint1" : "<ul><li> (手順①) アンケートページの名前欄に脆弱性があります。</li>
           <li> (手順②) document.getElementById(&quot;account&quot;).innerHTML の
           値を書き換えると、アンケートページの内容を書き換えることができます。</li></ul>",
...
```

コードとして表示する場合は「"」を使用する。

図 23 ダブルクォートの例

シングルクォート

ヒント内容中にシングルクォートをコードとして表示する場合は「'」を使用します。

```
...
"hint3" : "<ul><li> (手順③) 次の URL を【掲示板】に投稿してみましょう。「http://ホスト
/Web/Scenario1121/VulSoft/enquete.php?page=2&amp;sex=0&amp;old=1&amp;company=&amp;xss=1&amp;
trouble=1&amp;
           content=&name=&lt; script&gt;
           document.getElementById (&quot;account&quot;).innerHTML =
           &#39;&lt;font&nbsp;color=&quot;blue&quot;&nbsp;size=&quot;3&quot;&gt; もれなく一万円をプレゼ
           ント&nbsp;をします。名前、住所、口座番号を入力してください。&lt;/font&gt;&#39;;&lt;/script&gt;」
           </li>
...
「&#39;」を使用する。
```

図 24 シングルクォートの例

円マーク

円 (¥) マークを表示するには、「\」を使用します。コード中の円マークも同じです。

```
...
"hint3" : "<ul><li> (手順②) 次のように変更後ファイル名を入力してみましょう。<br />
「exame.txt &amp; dir /b c:&#92;」 </li></ul>",
...
```

「\」を使用する。

図 25 円マークの例

5. 称号名の設定手順

ここでは習熟度テストで得られる称号名の設定の変更を説明します。称号名は全て言語フォルダ内の言語ファイルに保存されています。言語ファイルはJSONという形式を使用しているため、記述の仕方を誤ると、テーマのコンテンツが表示されなくなってしまうことに注意が必要です。

5.1. 称号名の設定

称号は図 26 の赤い丸で囲んだ部分に相当します。称号の説明が記載されているツールチップが存在し、それは図 26 の噴出し部分に相当します。また、学習の進め方にも図 27 のように称号の説明が記載されています。

IPA 脆弱性体験学習ツール AppGoat

総合メニュー 学習を進める前に 学習環境へ 学習者の管理 演習セットの設定 FAQ 利用者マニュアル AppGoatの終了方法 ログアウト ログインユーザ :

学習状況表示

1つ前のページに戻る

ログインID yamada
氏名 山田
所属 開発部
演習セット 全演習
ログイン状況 未ログイン
称号 **入門者** (現在0問正答しています。あと14問正答すると上位の称号へ進みます。)

①称号名の設定

②ツールチップの変更

称号とは
・習熟度テストの全体の正答率を基に以下の名称を付与しています。
- 卒業(正答率:100%)
- 上級者(正答率:80%以上)
- 中級者(正答率:60%以上)
- 初級者(正答率:40%以上)
- 初心者(正答率:20%以上)
- 入門者(正答率:20%未満)

管理者向け…ウェブサイトの管理者
開発者向け…ウェブアプリケーション開発者

種別	脆弱性	学習対象		正答率(%)
		管理者向け	開発者向け	
				-
				-
				-
				-

図 26 称号名の修正箇所とリンク

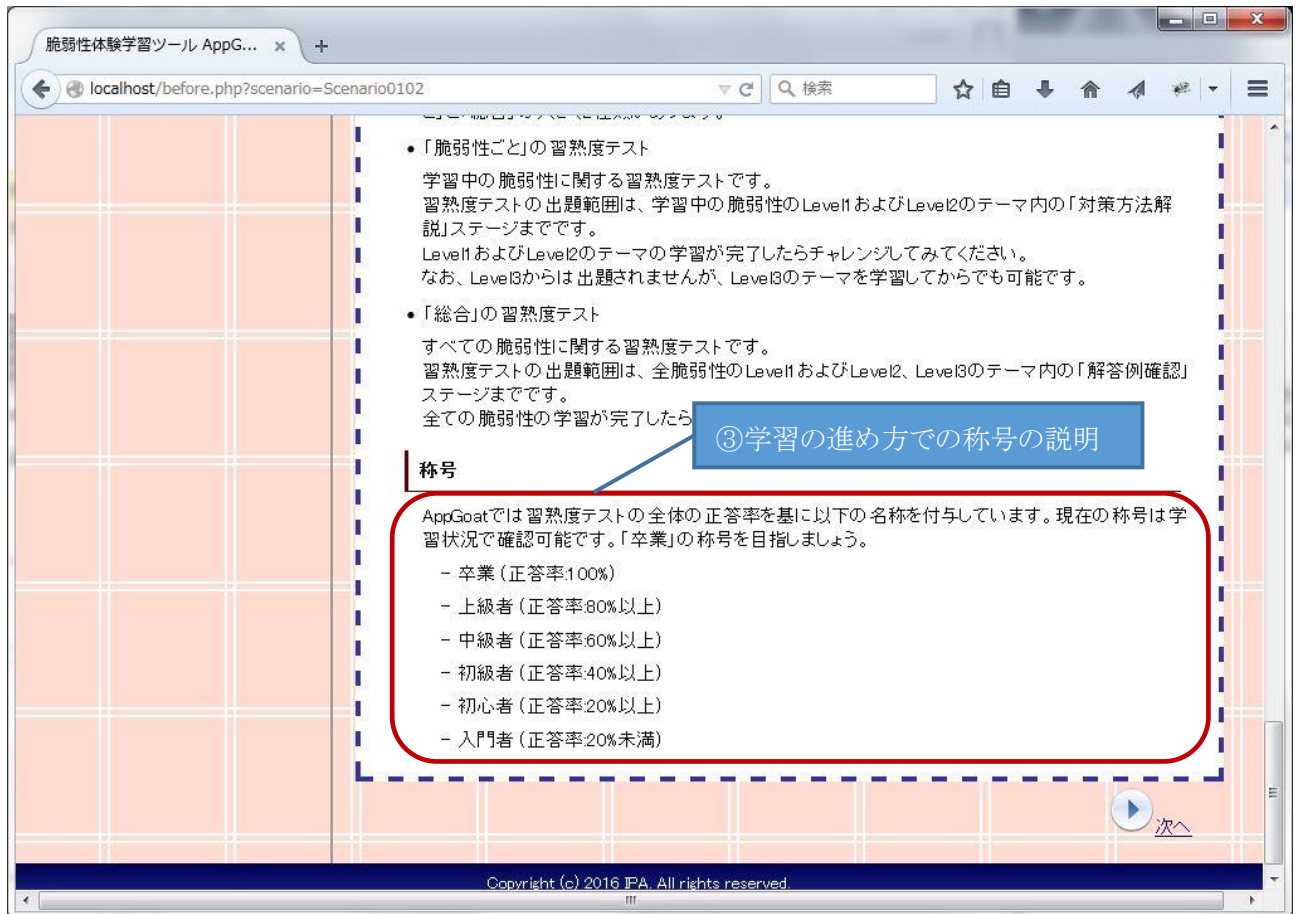


図 27 学習の進め方での称号の説明とリンク

- 図 26 や図 27 のようにデフォルトで称号名が設定されています。称号名を変更したい場合は、
- ① 称号名の設定
 - ② ツールチップの記載
 - ③ 学習の進め方での称号の説明

以上の 3 箇所を変更する必要があります。また、①、③の反映は一括処理の「学習状況の全初期化」では行えません。変更後に作成する学習者アカウントには自動で反映されますが、変更前から存在するアカウントに対しては、一度削除してアカウントを作成しなおすか、「IPATool¥Users¥アカウント名¥Web」のフォルダに対して同様の処理を行う必要があります。

5.1.1. ①称号名の設定

称号名の設定は「IPATool¥Scenarios¥Web¥Lang」の ja.txt ファイルに記載されています。称号名を変更したい場合は、「rank」を編集します。

```
},  
"rank":{  
  "RANK1": "卒業",  
  "RANK2": "上級者",  
  "RANK3": "中級者",  
  "RANK4": "初級者",  
  "RANK5": "初心者",  
  "RANK6": "入門者"  
}  
}
```

図 28 称号の設定

5.1.2. ②ツールチップの変更

ツールチップの内容は「IPATool¥Framework¥Lang」の ja.txt ファイルに記載されています。称号名を変更した場合は、学習者が混乱しないように「tooltip_rank_note」を編集します。

```
{  
  ...  
  "label_progress": "最終学習履歴",  
  "label_no_progress_quiz": "未実施",  
  "tooltip_rank_note": "称号とは<BR>・習熟度テストの全体の正答率を基に以下の名称を付与  
しています。<BR> - 卒業 (正答率: 100%) <BR> - 上級者 (正答率: 80%以上) <BR> - 中  
級者 (正答率: 60%上) <BR> - 初級者 (正答率: 40%以上) <BR> - 初心者 (正答率: 20%  
以上) <BR> - 入門者 (正答率: 20%未満) ",  
  "title_show_practice_set_menu": "演習セット管理",  
  ...  
}
```

図 29 ツールチップ

6. カスタマイズ実施例

ここでは、利用が多そうなカスタマイズをピックアップして修正方法例を紹介します。

6.1. 画像の変更(追加)

ここでは画像を変更・追加する手順を説明します。

URLから修正箇所を判別する。

原理解説

反射型クロスサイト・スクリプティングの脆弱性とは、ウェブアプリケーションがユーザから受け取った入力データを、そのままの形(実行可能な形)でウェブページの出力に利用してしまう問題です。

この脆弱性がどのように悪用されてしまうのか、次の図を使って見てみましょう。

①不正なスクリプトを含む
悪のリンクを貼り付ける

②クリックにより
不正なスクリプトを
送信する

③不正なスクリプトを含む
ウェブページを出力する

④不正なスクリプトを
実行させられる

図 1: 反射型クロスサイト・スクリプティングの脆弱性の原理

- 悪意のある人が、不正なスクリプトを含む悪のリンクを掲示板などの第三者のウェブサイトに貼り付けます。
- 被害者が第三者のウェブサイトを訪ね、悪のリンクをクリックしてしまうことで、不正なスクリプトを今

図 31 カスタマイズ対象の画面

概要 シナリオ ID 「Scenario1111」 の原理解説の画像を変更（追加）する
手順

① 修正するシナリオの画像フォルダを開く

URL に着目すると、「scenario=Scenario1111」となっています。対象となるシナリオフォルダは「IPATool¥Scenarios¥Web¥Scenario1111」となります。画像フォルダである Image フォルダを開きます。

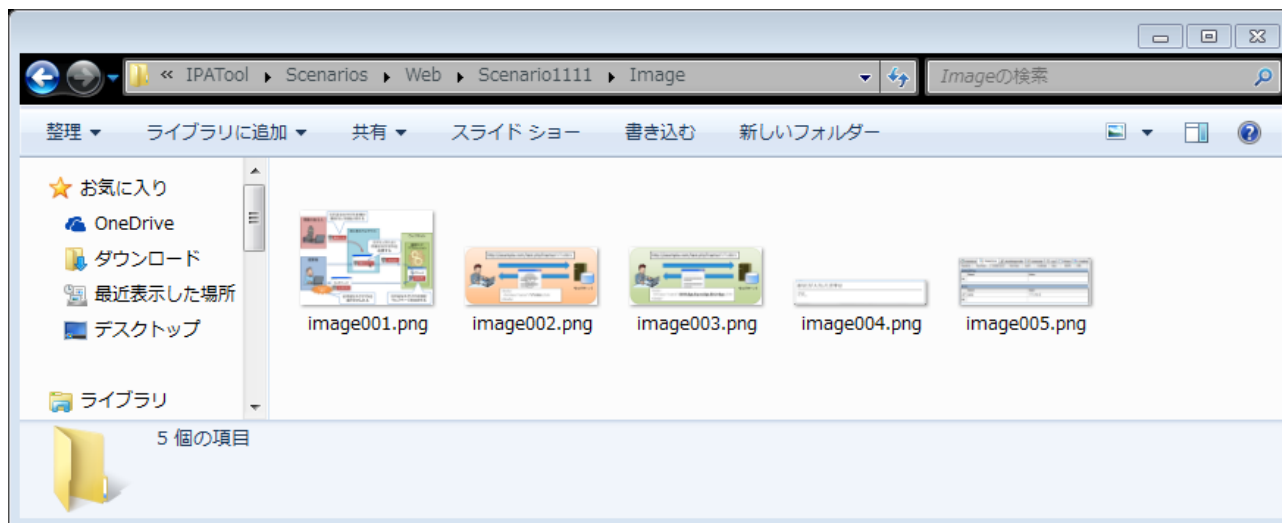


図 32 変更前の画像フォルダ

② 画像ファイルを変更する（画像変更の場合）

変更する画像を変更したい画像に置き換えます。手順⑥に移ります。

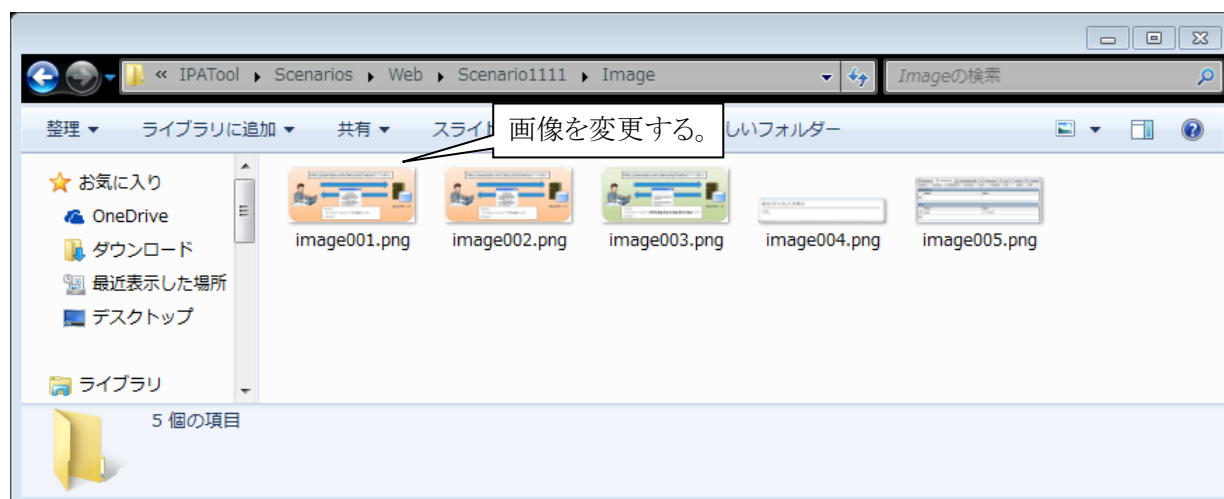


図 33 画像変更後の画像フォルダ

③ 画像を追加する（画像追加の場合）

画像フォルダに追加したい画像を追加します。



図 34 画像追加後の画像フォルダ

④ 言語ファイルを開く（画像追加の場合）

文章（画像）は言語ファイルで記述されているので、シナリオフォルダ配下の **Lang** フォルダの **ja.txt** を開きます。

⑤ 言語ファイルに画像の表示を記載する（画像追加の場合）

URL に着目すると「**stage=stage2**」となっているので **stage2** を確認します。変更前の記述は図 35 のようになっており、画像の表示には赤枠部分を用いています。画像を追加したい場所にこの赤枠部分をコピーします。その後、この場合パスの「**image001.png**」部分を追加した画像ファイル名に置き換えます。また、**width**、**height** の値で横と縦のサイズの調整、図タイトルの変更を行います。

加) 後に作成する学習者アカウントには自動で反映されますが、変更前から存在するアカウントに対しては、一度削除してアカウントを作成しなおすか、「IPATool¥Users¥アカウント名¥Web」のシナリオフォルダに対して②～⑥の処理を行う必要があります。

カスタマイズ結果

脆弱性体験学習ツール AppG... x 新しいタブ

localhost/main.php?scenario=Scenario1111&stage=stage2

脆弱性の概要および発見演習

基礎

クロスサイト・スクリプティング

イントロダクション
クロスサイト・スクリプティングとは

Level1
脆弱性の概要および発見演習

Level2
アンケートページの改ざん(反射型)
入力情報の表示欄に埋め込み(納型)
ウェブページの改ざん(DOMベース)

Level3
不完全な対策
ヘッダ要素へのスクリプト

習熟度テスト
テスト問題 全5問

SQLインジェクション

イントロダクション
SQLインジェクションとは

Level1
脆弱性の概要および発見演習

Level2
不正なログイン(文字列リテラル)
情報漏えい(数値リテラル)
他テーブル情報の漏えい(数値リテラル)
データベースの改ざん(数値リテラル)

Level3

原理解説

反射型クロスサイト・スクリプティングの脆弱性とは、ウェブアプリケーションがユーザから受け取った入力データを、そのままの形(実行可能な形)でウェブページの出力に利用してしまう問題です。

この脆弱性がどのように悪用されてしまうのか、次の図を使って見てみましょう。

図1: 反射型クロスサイト・スクリプティングの脆弱性の原理

図2: カスタマイズサンプル

図1の注釈: 画像が変更された。

図2の注釈: 設定したサイズで画像が追加された。

図 37 カスタマイズ後の画面

6.2. ヒントの追加

ここでは、ヒントの追加およびコンテンツ文章の変更を学習者全体に反映する手順を説明します。

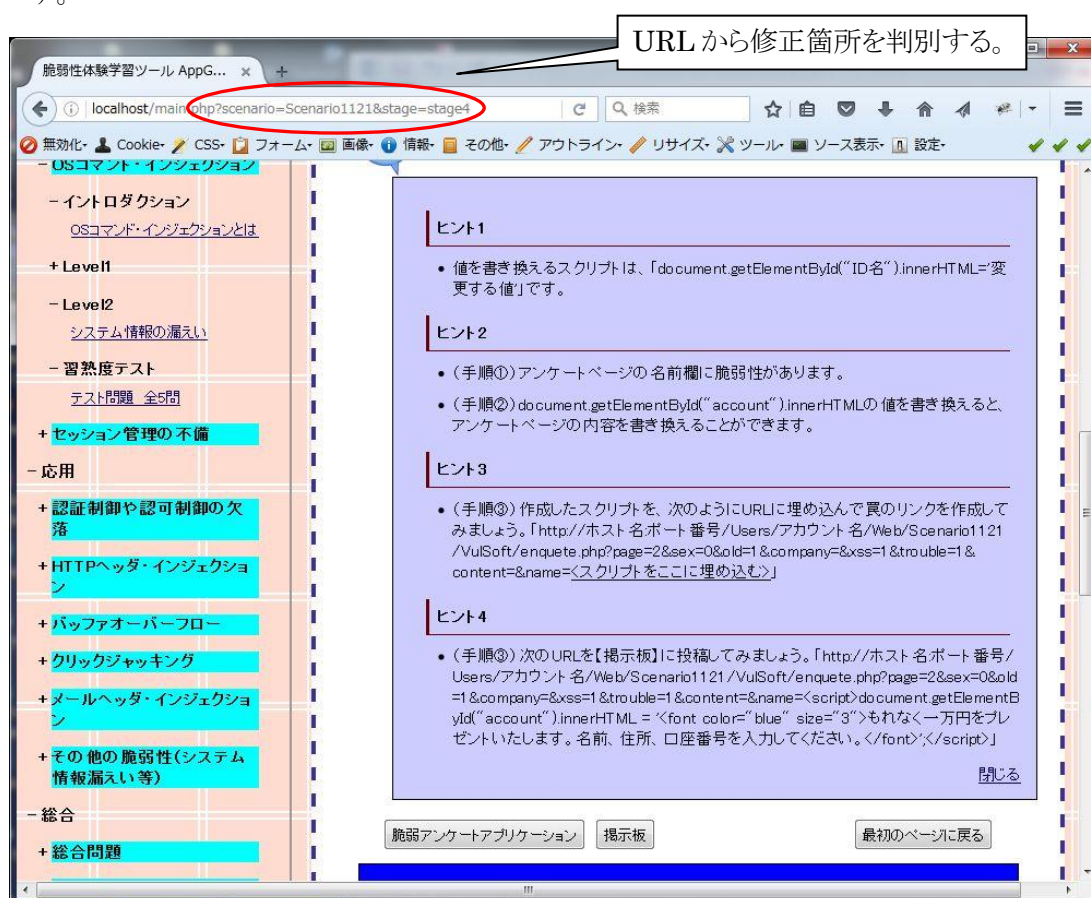


図 38 カスタマイズ対象の画面

概要 シナリオ ID 「Scenario1121」 のヒント 2 を 2 つにわけ、ヒントを 5 段階にする

手順

① 修正する言語ファイルを開く

URL に着目すると、「scenario=Scenario1121」となっています。対象となるシナリオフォルダは「IPATool¥Scenarios¥Web¥Scenario1121」となります。ヒントは言語ファイルで記述されているので、シナリオフォルダ配下の Lang フォルダの ja.txt を開きます。

② 該当箇所を確認する

URLに着目すると「stage=stage4」となっているので stage4 を確認します。ヒントは図 39 のように記載されています。

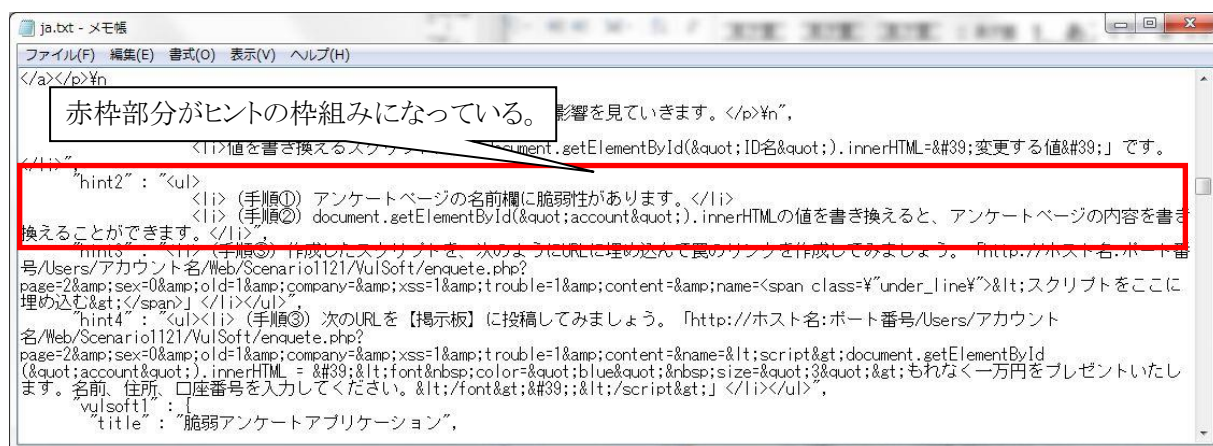


図 39 ヒントの構成

③ ヒントの枠組み追加

図 39 の赤枠部分がヒント 1 つ分に相当します。ヒントを減らしたいときは、赤枠部分を削除することで、ヒントを増やしたいときは、赤枠部分を追加することで行えます。今回はヒントを増やすので赤枠部分をコピーしてすぐ下に貼り付けます。

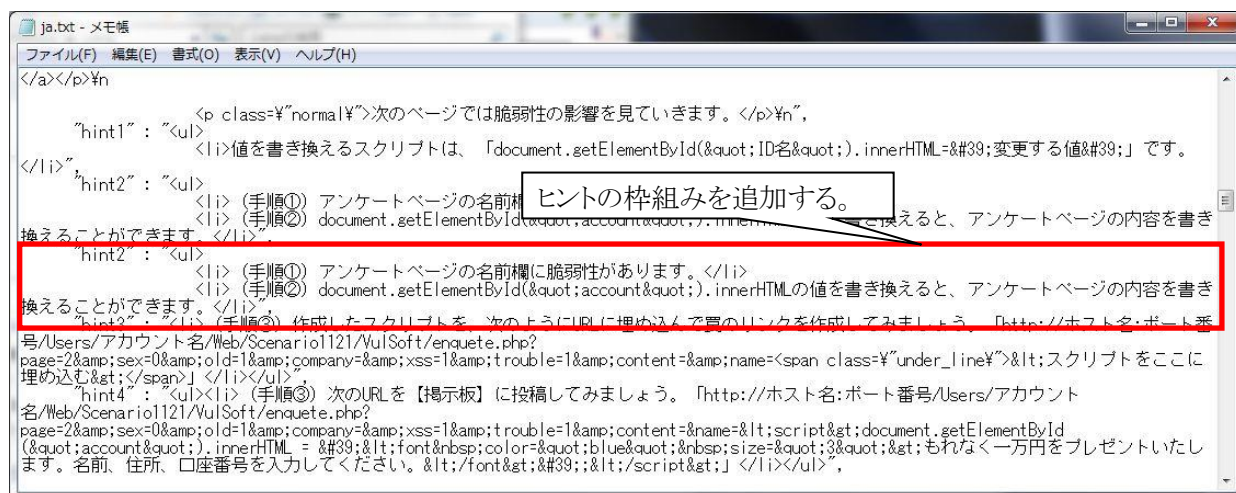


図 40 ヒントの枠組み追加

ヒントの枠組みを削除することでヒントを削除できる。同様にヒント 2,3,4 を削除することでヒントをなくすことができる。

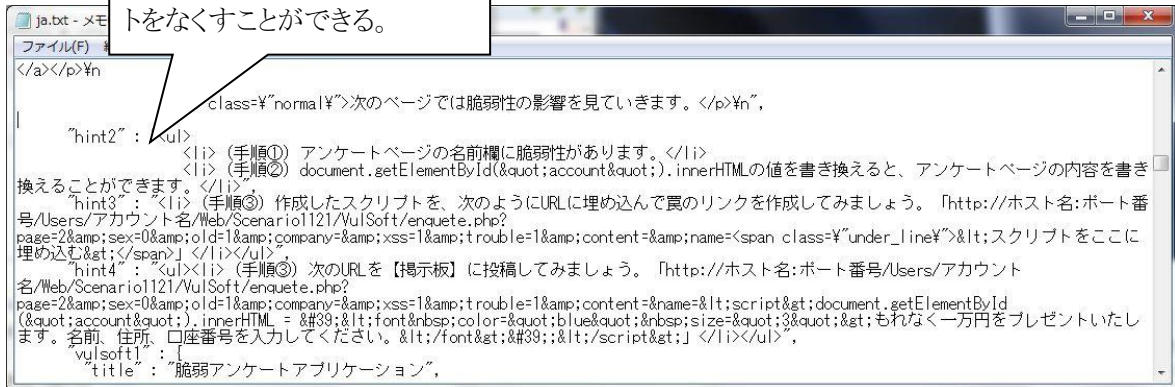


図 41 (参考) ヒント 1 を削除する場合

④ ヒントの編集

ヒントを編集します。hint1、hint2、hint2、・・・となっているので、hint1、hint2、hint3、・・・と変更し、文言を編集します。今回は、hint2 の手順②を hint3 に移動します。

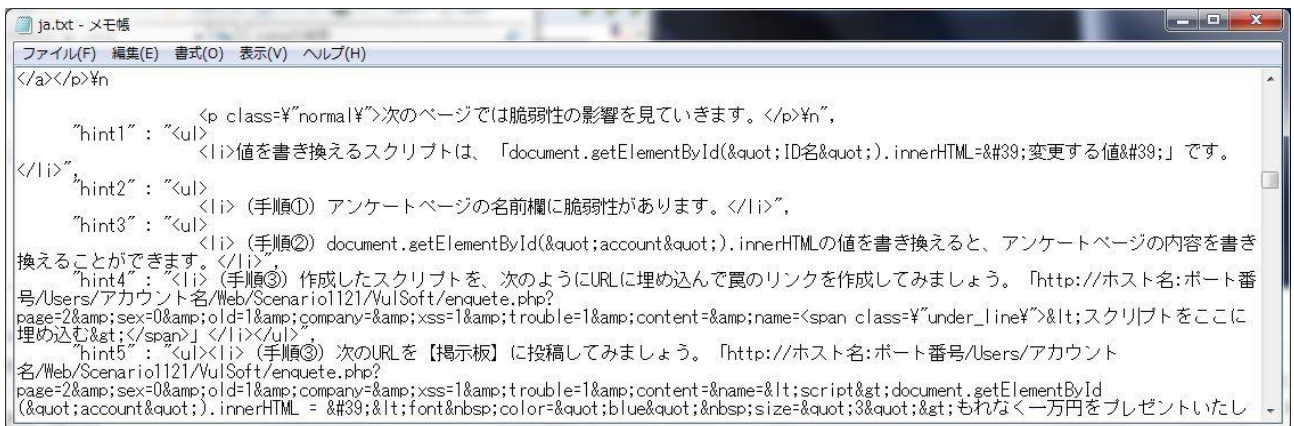


図 42 編集後の言語ファイル

⑤ 学習者への反映

シナリオフォルダ「IPATool¥Scenarios¥Web¥Scenario1121」の変更は終了しましたが、学習者への反映は行われていません。学習者への反映は、学習者管理画面の学習状況の全初期化で行います。

IPA 脆弱性体験学習ツール AppGoat

総合メニュー 学習を進める前に 学習環境へ 学習者の管理 演習セットの設定 FAQ 利用者マニュアル AppGoatの終了方法 ログアウト ログインユーザ : 田中

学習者管理

学習者情報

新規登録

学習者の学習状況情報

一括処理

- 学習者削除
- 演習セット変更 ---演習セットを選択---
- 学習状況の全初期化

実行

① 変更を反映したいアカウントにチェックを入れる。

② 学習状況の全初期化を選んで実行をクリックする。

ID	所属	氏名	演習セット	ログイン状況	称号	最終学習履歴	学習状況	学習者情報
<input checked="" type="checkbox"/> tanaka (管理者)	所属	田中	全演習	ログイン中	入門者		表示 部分初期化	変更
<input checked="" type="checkbox"/> user1	開発部	学習者1	全演習	未ログイン	入門者		表示 部分初期化	変更 削除
<input checked="" type="checkbox"/> user2	開発部	学習者2	全演習	未ログイン	入門者		表示 部分初期化	変更 削除

図 43 学習状況の全初期化

カスタマイズ結果

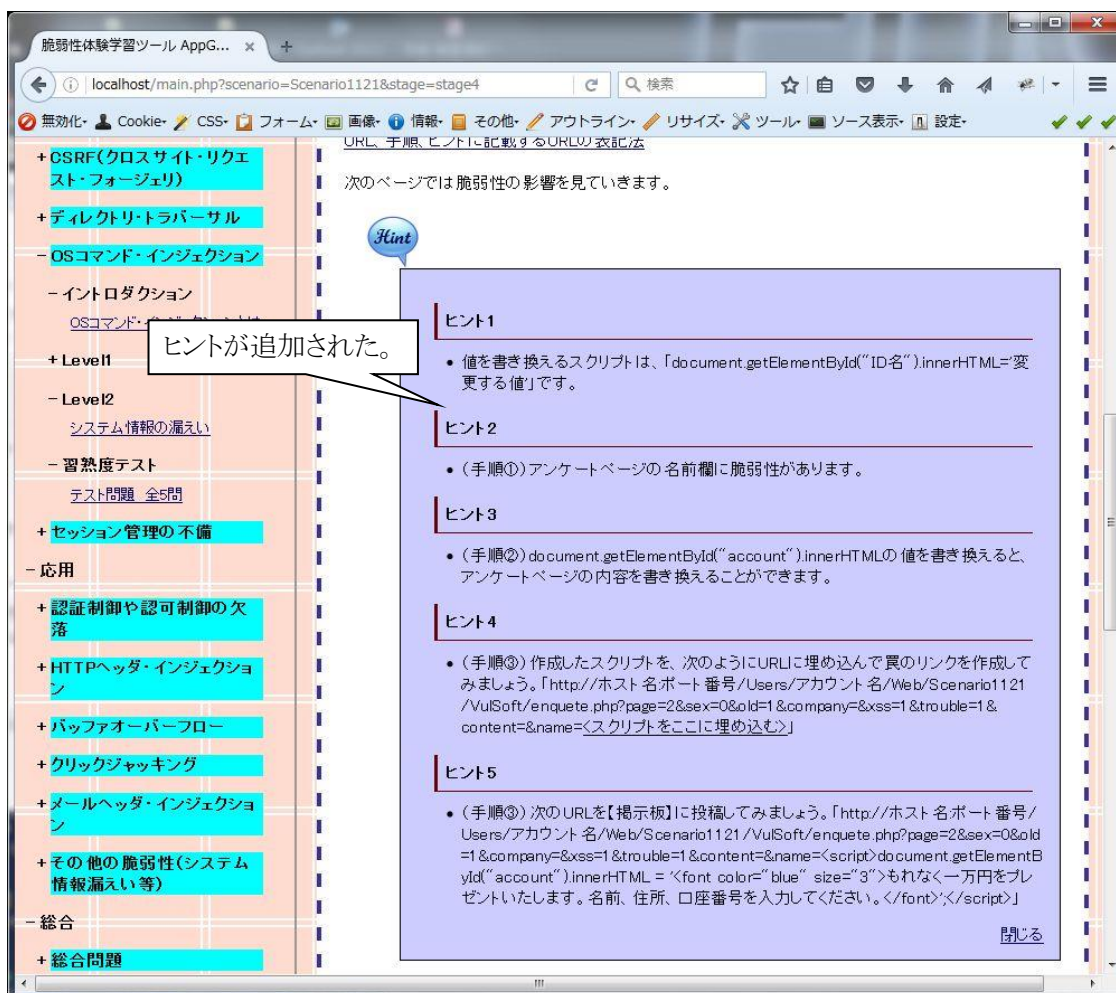


図 44 カスタマイズ後の画面

6.3. 説明用コンテンツの記載内容修正(学習環境に合わせたファイルパスの追記)

ここでは、説明用コンテンツの記載内容修正(学習環境に合わせたファイルパスの追記)を通して、個別の学習者に対してコンテンツ文章を変更する手順を説明します。



図 45 カスタマイズ対象の画面

概要 学習者アカウント「user1」のシナリオ ID「Scenario1311」の画面(図 45)の赤枠部分のパスに具体的な値を入れる

(教育者が起動させる環境は次のとおりです。ホスト名: 172.10.20.30、ポート番号: 82)

手順

① 修正する言語ファイルを開く

URLに着目すると、「scenario=Scenario1311」となっています。

今回はアカウント名「user1」へのみの編集なので、対象となるシナリオフォルダは「IPATool¥Users¥user1¥Web¥Scenario1311」となります。文言は言語ファイルで記述されているので、シナリオフォルダ配下のLangフォルダのja.txtを開きます。

② 該当箇所を確認する

URLに着目すると「stage=stage4」となっているのでstage4を確認します。該当箇所は図 46 のように記載されています。

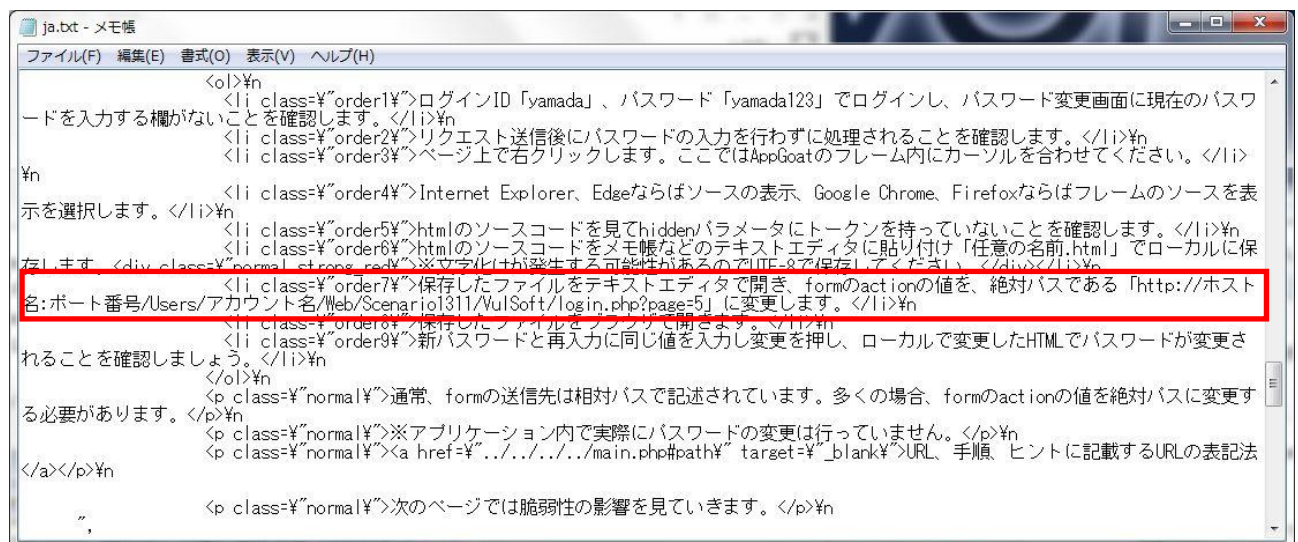


図 46 該当箇所

③ パスの編集

パスを具体的な値に書き換えます。

今回は「http://172.10.20.30:82/Users/user1/Web/Scenario1311/VulSoft/login.php?page=5」となります。また、アカウント名「user1」が参照しているフォルダを直接編集したので、画面への反映がすぐに行われます。

```
ja.txt - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
<ol>%n
<li class=%order1%>ログインID「yamada」、パスワード「yamada123」でログインし、パスワード変更画面に現在のパスワードを入力する欄がないことを確認します。</li>%n
<li class=%order2%>リクエスト送信後にパスワードの入力を行わずに処理されることを確認します。</li>%n
<li class=%order3%>ページ上で右クリックします。ここではAppGoatのフレーム内にカーソルを合わせてください。</li>%n
<li class=%order4%>Internet Explorer、Edgeならばソースの表示、Google Chrome、Firefoxならばフレームのソースを表示を選択します。</li>%n
<li class=%order5%>htmlのソースコードを見てhiddenパラメータにトークンを持っていないことを確認します。</li>%n
<li class=%order6%>htmlのソースコードをメモ帳などのテキストエディタに貼り付け「任意の名前.html」でローカルに保存します。<div class=%normal_strength%>※文字化けが発生する可能性があるためUTF-8で保存してください。</div></li>%n
<li class=%order7%>保存したファイルをテキストエディタで開き、formのactionの値を、絶対パスである「http://172.10.20.30:82/Users/user1/Web/Scenario1311/VulSoft/login.php?page=5」に変更します。</li>%n
<li class=%order8%>保存したファイルをブラウザで開きます。</li>%n
<li class=%order9%>新パスワードと再入力に同じ値を入力し変更を押し、ローカルで変更したHTMLでパスワードが変更されることを確認しましょう。</li>%n
</ol>%n
<p class=%normal%>通常、formの送信先は相対パスで記述されています。多くの場合、formのactionの値を絶対パスに変更する必要があります。</p>%n
<p class=%normal%>※アプリケーション内で実際にパスワードの変更は行っていません。</p>%n
<p class=%normal%><a href=%.././.././../main.php#path% target=%_blank%>URL、手順、ヒントに記載するURLの表記法</a></p>%n
<p class=%normal%>次のページでは脆弱性の影響を見ていきます。</p>%n
```

図 47 編集後の言語ファイル

カスタマイズ結果

脆弱性体験学習ツール AppG... x +

localhost:82/main.php?scenario=Scenario1311&stage=stage4

アンケートページの改ざん(反射型)
入力情報の漏えい(反射型)
掲示板に埋め込まれるスクリプト(格納型)
ウェブページの改ざん(DOMベース)

- LevelB
不完全な対策
ヘッダ要素へのスクリプト

- 習熟度テスト
テスト問題 全5問

- SQLインジェクション

- イントロダクション
SQLインジェクションとは

- Level1
脆弱性の概要および発見演習

- Level2
不正なログイン(文字列リテラル)
情報漏えい(数値リテラル)
他テーブル情報の漏えい(数値リテラル)
データベースの改ざん(数値リテラル)

- LevelB
ブラインドSQLインジェクション

- 習熟度テスト
テスト問題 全5問

次の手順に従って、検査を進めてみましょう。

- 1 ログインID「yamada」、パスワード「yamada123」でログインし、パスワード変更画面に現在のパスワードを入力する欄がないことを確認します。
- 2 リクエスト送信後にパスワードの入力を行わずに処理されることを確認します。
- 3 ページ上で右クリックします。ここではAppGoatのフレーム内にカーソルを合わせてください。
- 4 Internet Explorer、Edgeならばソースの表示、Google Chrome、Firefoxならばフレームのソースを選択します。
- 5 htmlのソースコードを見てhiddenパラメータトークンを持っていないことを確認します。
- 6 htmlのソースコードをメモ帳などのテキストエディタに貼り付け「任意の名前.html」でローカルに保存します。
※文字化けが発生する可能性があるためUTF-8で保存してください。
- 7 保存したファイルをテキストエディタで開き、formのactionの値を、絶対パスである「http://172.10.20.30:82/Users/user1/Web/Scenario1311/VulSoft/login.php?page=5」に変更します。
- 8 保存したファイルをブラウザで開きます。
- 9 新パスワードと再入力に同じ値を入力し変更を押し、ローカルで変更したHTMLでパスワードが変更されることを確認しましょう。

通常、formの送信先は相対パスで記述されています。多くの場合、formのactionの値を絶対パスに変更する必要があります。

※アプリケーション内で実際にパスワードの変更は行っていません。

URL、手順、ヒントに記載するURLの表記法

次のページでは脆弱性の影響を見ていきます。

脆弱パスワード変更画面

最初のページに戻る

URL http://localhost:82/Users/user1/Web/Scenario1311/VulSoft/login.php GO

図 48 カスタマイズ後の画面

7. 終わりに

本マニュアルは、AppGoat を使用して集合教育を行う管理者が、教育用コンテンツを変更するための方法を記載したものです。管理者が説明しやすいように文言の一部を修正する際にご利用ください。

ただし、既存のコンテンツ内では HTML の文法や AppGoat 上で作成されたクラスが多用されています。コンテンツの変更内容によっては、画面レイアウトが崩れる、演習が動かなくなる、想定外の脆弱性が生まれる、といったことが起こり得ます。

コンテンツ変更によるトラブルについて、独立行政法人情報処理推進機構は関知いたしませんので、全て集合教育モードの管理者の自己責任で行って頂くようにお願いします。

補足資料

表 5 シナリオ番号一覧

種別	カテゴリ	レベル	テーマ	シナリオ番号
-	学習を進める前に	-	学習の対象者と目的について	0101
		-	学習の進め方 ～概要～	0102
		-	学習の進め方 ～演習～	0103
基礎	クロスサイト・スクリプティング	イントロダクション	クロスサイト・スクリプティングとは	1101
		Level1	脆弱性の概要および発見演習	1111
		Level2	アンケートページの改ざん(反射型)	1121
			掲示板に埋め込まれるスクリプト(格納型)	1122
			入力情報の漏えい(反射型)	1123
			ウェブページの改ざん(DOM ベース)	1124
		Level3	不完全な対策	1131
			ヘッダ要素へのスクリプト	1132
	習熟度テスト	習熟度テスト	1141	
	SQL インジェクション	イントロダクション	SQL インジェクションとは	1201
		Level1	脆弱性の概要および発見演習	1211
		Level2	不正なログイン(文字列リテラル)	1221
			情報漏えい(数値リテラル)	1222
			他テーブル情報の漏えい(数値リテラル)	1223
データベースの改ざん(数値リテラル)			1224	
Level3		ブラインド SQL インジェクション	1231	
習熟度テスト		習熟度テスト	1241	
CSRF(クロスサイト・リクエスト・フォージェリ)	イントロダクション	CSRF(クロスサイト・リクエスト・フォージェリ)とは	1301	
	Level1	脆弱性の概要および発見演習	1311	

種別	カテゴリ	レベル	テーマ	シナリオ番号	
		Level2	意図しない命令の実行	1321	
			不完全な対策	1322	
		習熟度テスト	習熟度テスト	1341	
	ディレクトリ・トラバーサル	イントロダクション	ディレクトリ・トラバーサルとは	1401	
		Level1	脆弱性の概要および発見演習	1411	
		Level2	ファイル情報の漏えい	1421	
		習熟度テスト	習熟度テスト	1441	
	OS コマンド・インジェクション	イントロダクション	OS コマンド・インジェクションとは	1501	
		Level1	脆弱性の概要および発見演習	1511	
		Level2	システム情報の漏えい	1521	
		習熟度テスト	習熟度テスト	1541	
	セッション管理の不備	イントロダクション	セッション管理の不備とは	1601	
		Level1	脆弱性の概要および発見演習	1611	
		Level2	セッション ID の推測	1621	
			セッション ID の漏えい	1622	
		Level3	セッション ID の固定化	1631	
		習熟度テスト	習熟度テスト	1641	
	応用	認証制御や認可制御の欠落	イントロダクション	認証制御や認可制御の欠落とは	2101
			Level1	脆弱性の概要および発見演習	2111
			Level2	認証不備	2121
認可不備				2122	
習熟度テスト			習熟度テスト	2141	
HTTP ヘッダ・インジェクション		イントロダクション	HTTP ヘッダ・インジェクションとは	2201	
		Level1	脆弱性の概要および発見演習	2211	
		Level2	Cookie 値の変更	2221	
		習熟度テスト	習熟度テスト	2241	

種別	カテゴリ	レベル	テーマ	シナリオ番号
	バッファオーバーフロー	イントロダクション	バッファオーバーフローとは	2301
		Level1	脆弱性の概要および発見演習	2311
		Level2	制限を越えた文字列の入力	2321
		習熟度テスト	習熟度テスト	2341
	クリックジャッキング	イントロダクション	クリックジャッキングとは	2401
		Level1	脆弱性の概要および発見演習	2411
		Level2	他サイトへのリクエスト送信	2421
		習熟度テスト	習熟度テスト	2441
	メールヘッダ・インジェクション	イントロダクション	メールヘッダ・インジェクションとは	2501
		Level1	脆弱性の概要および発見演習	2511
		Level2	メール送信先の追加	2521
		習熟度テスト	習熟度テスト	2541
	その他の脆弱性(システム情報漏えい等)	Level1	脆弱性の概要および発見演習	2611
		Level2	エラーメッセージ	2621
		付録	罠サイトへの誘導(オープンリダイレクト)	2631
		習熟度テスト	習熟度テスト	2641
総合	総合問題	習熟度テスト	習熟度テスト	3101
	脆弱性検査	-	演習 1(ネット証券)	3201
		-	演習 2(ネットショッピング)	3301
		-	演習 3(グループウェア)	3401
補足	Fiddler の使い方	-	Fiddler の使い方	4101
	Smarty について	-	Smarty について	4102
	脆弱性検査ツール(OWASP ZAP)の使い方	-	脆弱性検査ツール(OWASP ZAP)の使い方	4103
	脆弱性の修正例(Java・Ruby)	-	はじめに	4201
		-	クロスサイト・スクリプティング	4202
		-	SQL インジェクション	4203

種別	カテゴリ	レベル	テーマ	シナリオ番号
		-	クロスサイト・リクエスト・フォージェリ	4204
		-	ディレクトリ・トラバーサル	4205
		-	OS コマンド・インジェクション	4206
		-	セッション管理の不備	4207
		-	認証制御や認可制御の欠落	4208
		-	HTTP ヘッダ・インジェクション	4209
		-	クリックジャッキング	4210
		-	メールヘッダ・インジェクション	4211
		-	エラーメッセージからの情報漏えい	4212
			オープンリダイレクト	4213