

# 開発者向け脆弱性実習ツールの開発

株式会社フォティーンフォティ技術研究所

## 1 概要

サーバ・デスクトップアプリケーション及びウェブアプリケーションには、これまで多くの脆弱性が発見、公表され、それぞれに対策が施されてきたが、こうした脆弱性の対策、検証方法を学ぶツールが整備されていなかった。そのため、今回、セキュアなアプリケーション構築ができる開発者や情報セキュリティ技術者を育成する環境を整備することを目的とし、脆弱性の発見方法や対策について実習形式で学ぶ実習ツールの開発を行った。

## 2 背景

サーバ・デスクトップアプリケーション及びウェブアプリケーションには、これまで多くの脆弱性が発見、公表され、それぞれに対策が施されてきた。それら脆弱性の多くは、既に知られているものと同種の脆弱性であり、対策方法も存在するが、こうした脆弱性の対策、検証方法を学ぶツールが整備されていなかったことが、新規に開発したソフトウェアに再び同種の脆弱性を作り込む1つの要因となっている。

そのため、セキュアなアプリケーション構築ができる開発者や情報セキュリティ技術者を育成するためにツールを開発し環境を整備することが必要となる。

## 3 目的

開発経験の浅い初心者から上級者まで広く利用できる、脆弱性の発見方法や対策について実習形式で体系的に学べるツールを提供することを目的として「開発者向け脆弱性実習ツール（以降、実習ツール）」の開発を行った。

## 4 実習ツール概要

### 4.1 利用形態

実習ツールは、クライアント上で動作する1つのウェブアプリケーションとして提供される。実習者はIPAのウェブサイトから実習ツールをダウンロードし、利用しているPCにインストールして利用する。実習者はウェブブラウザを使ってコンテンツ

を読みながら実習を進める形式をとる。実習ツールには「ウェブアプリケーション実習環境」と「サーバ・デスクトップアプリケーション実習環境」という2つの実習用コンテンツを用意し、それぞれの分野の脆弱性について学べるようになっている（図1）。

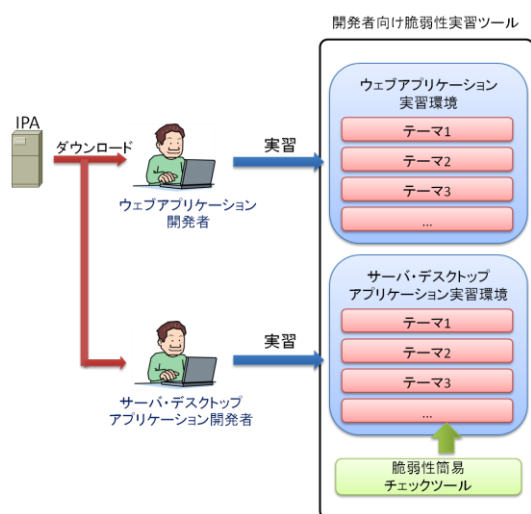


図 1 利用全体像

実習ツールには「脆弱性簡易チェックツール」と呼ばれるプログラムの脆弱性を確認するツールが含まれており、サーバ・デスクトップアプリケーション実習環境において、これらを使いながら脆弱性の発見手法や確認方法も学ぶ。

それぞれの実習環境には、実習テーマが複数用意され、ナビゲーションに従い、実習を順に進めることで体系的に脆弱性について学ぶ。

この実習ツールの大きな特徴として、実際に脆弱性が含まれるアプリケーションを

操作し、脆弱性によってどのような問題が発生するかを確認する演習が各テーマに含まれている点あげられる。これにより、テキストを読むだけでは得られない、脆弱性についてのより深い理解が得られるようになっている。

## 4.2 動作環境

実習ツールの動作に必要な環境を以下に示す。

- Windows XP SP3, Windows Vista SP2, Windows 7 (32bit)のいずれかのOSの動作するPC
- IPv4のネットワーク接続
- Internet Explorer 7 または Firefox 3.6以上のブラウザ
- Adobe Acrobat Reader 9以上
- 1024 × 768ピクセル以上の解像度のモニタ

## 5 実習内容

「サーバ・デスクトップアプリケーション実習環境」には13のテーマ、「ウェブアプリケーション実習」には15のテーマが存在し、それらはカテゴリ分けされている。表1、表2にその一覧を示す。

テーマはこれらの表に示した順で実習を行うことで、体系的に脆弱性について学習することができる。

表 1 テーマ一覧  
ウェブアプリケーション実習環境

クロスサイト・スクリプティング
クロスサイト・スクリプティングとは
アンケートページの改ざん(反射型)
掲示板に埋め込まれるスクリプト(格納型)
入力情報の漏えい(反射型)
ウェブページの改ざん(DOM ベース)
不完全な対策
SQL インジェクション
SQL インジェクションとは
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)
CSRF
CSRF(クロスサイト・リクエスト・フォージェリ)とは
意図しない命令の実行
不完全な対策
その他
エラーメッセージからの情報漏えい

表 2 テーマ一覧  
サーバ・デスクトップアプリケーション実習環境

バッファオーバーフロー
バッファオーバーフローとは
アーカイブソフトの異常終了
FTP プロキシソフトの異常終了
ウェブサーバの異常終了(ヒープ領域)
ディレクトリ・トラバーサル
ディレクトリ・トラバーサルによる情報漏えい
リソースリーク
プログラミングエラーによるリソースリーク
整数オーバーフロー
整数オーバーフローによる異常終了
フォーマット文字列
フォーマット文字列による異常終了
認証・認可
本人認証の不備
権限管理の不備によるファイルの漏えい
その他
ジャンクションへの考慮不足の問題
TOCTOU による検証の迂回
暗号の不適切な利用

## 6 実習の流れ

### 6.1 実習開始

実習ツールには、実習中に脆弱性の動作を再現するためにウェブサーバやデータベースが内蔵されているが、それらは「開始プログラム」という 1 つのプログラムにまとめられているため、実習者は簡単に環境を整えて実習を開始することができる。

実習ツール起動後、総合メニューが表示

され学びたい実習環境を選択する（図 2）。

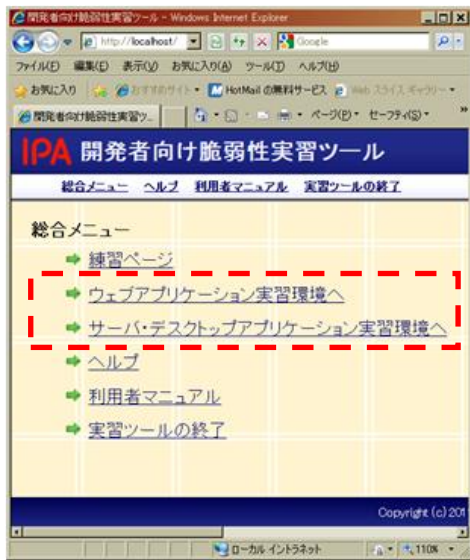


図 2 総合メニュー

実習環境選択後、実習者はテーマを選択し、ナビゲーションに従って実習を進める（図 3）。



図 3 実習画面

テーマ一覧（左）とコンテンツ領域（右）

## 6.2 テーマ構成

各テーマは統一された流れを持っており、実習者が迷いなく進めるように構成されている。これはステージと呼ばれる学習の

トップを统一的に設け、すべてのテーマでその順に解説を行うことで実現している。

図 4 にテーマのステージ構成を示す。

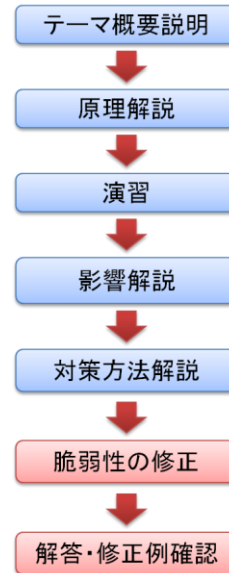


図 4 ステージ構成

（赤色のステージはサーバ・デスクトップアプリケーション実習環境のみ）

それぞれのステージで、実習者は以下に示す内容を学習する。

### テーマ概要説明

テーマの概要を理解する

### 原理解説

テーマで学ぶ脆弱性の原理やその前提となる知識を学ぶ

### 演習

意図的にアプリケーションに埋め込んだ脆弱性を実習者自身が操作しながら探し出し、その影響を確認しながら、原理解説で学んだことの理解を深める

### 影響解説

テーマの脆弱性によって引き起こさ

れる影響について、現実的な例などを含めて学ぶ

### 対策方法解説

テーマの脆弱性について、どのように対策することで問題を回避することができるかを学ぶ

### 脆弱性の修正

脆弱性を含むソースコードを書き直し、脆弱性の修正を行うことで、テーマの脆弱性の対策方法について理解を深める（サーバ・デスクトップアプリケーション実習環境のみ）

### 解答・修正例確認

脆弱性の修正ステージで行った修正の解答や修正例を読み、正しく対策方法を理解できているかを確認する（サーバ・デスクトップアプリケーション実習環境のみ）

## 6.3 演習ステージ

演習ステージは実習の中で最も重要なステージである。演習ステージでは、それまでに学んだことを利用し、脆弱性を含んだウェブアプリケーションやサーバ・デスクトップアプリケーションを操作して、疑似的に攻撃するなどをし、脆弱性による影響を体験する。

演習ステージは、課題形式になっており、最初にある課題が与えられる。課題はソースコードから脆弱性を探したり、脆弱性を突くことでアプリケーションに実際に問題を起こさせたりするものなどがある。課題

をクリアするには、脆弱性の本質的な理解が必要になっているため、そこまでに学習したことが理解できているかを確認することができる。

また、演習には複数のヒントが設けてあるため、課題をクリアすることが難しい場合には、ヒントに従うことで演習を進め、その場で理解できるようになっている。

### ウェブアプリケーション実習環境の演習

ウェブアプリケーション実習環境では図 5 に示すように、実習画面内に埋め込まれたウェブアプリケーションを実際に操作し、クロスサイト・スクリプティングや CSRF など、テーマに沿った脆弱性によってどのような問題が起こるのかを実際に体験する。



図 5 演習画面

ウェブアプリケーション実習環境



図 6 演習画面

サーバ・デスクトップアプリケーション実習環境

サーバ・デスクトップアプリケーション実習環境の一部の演習には脆弱性簡易チェックツールを用いて、脆弱性の発見や確認を行うものが含まれる。脆弱性の発見には、たとえば、ファジングという手法が存在するが、脆弱性簡易チェックツールに含まれるファイルフォーマットファザージャやプロトコルファザージャを用いて、その基本的な概念から、実際にプログラムに適用する方法まで学ぶことができる。

さらに、サーバ・デスクトップアプリケーション実習環境のテーマでは脆弱性の修正も行う。それまでに学習した内容をもとに、脆弱性の含まれているソースコードを修正し、コンパイルして脆弱性が修正されていることを確認する。

## サーバ・デスクトップ アプリケーション実習環境の演習

サーバ・デスクトップアプリケーションも同様に脆弱性を含んだアプリケーションを実際に操作しながら演習を行う。Windows上で動作する代表的なアプリケーションを模したプログラムを動作させ、脆弱性による影響を体験する（図 6）。

## 7 実習ツールによる効果

ここまで説明したように、実習ツールは統一された流れに沿って体系的に脆弱性に

についての学習ができるツールとなっている。

開発経験が浅く、まだプログラムの複雑な動作について理解が十分でない実習者は、この実習ツールを通して、脆弱性による影響を自ら体験することができる。これを、より深く脆弱性について理解し、その対策方法を学習していくきっかけとすることができる。

開発経験が十分ある実習者は、解説や実際のプログラムの動作を通じて、その原因や対策について、プログラム内部の動作を含めて理解することができる。原因、対策を原理からしっかり理解することで、その後の開発時に脆弱性を作り込むことを未然に防ぐことが期待できる。

また、脆弱性簡易チェックツールを通じた演習では、脆弱性の発見、確認の基本的な概念、手法について学習し、実際の開発時に適用することで、より安全なプログラムの開発を行えるようになると期待できる。

## 8 今後の課題

### 8.1 テーマの充実

今回の開発では全部で 28 のテーマを用意し実習できるようにした。今後はこのテーマを増やし、より多くの脆弱性について学習できるツールとしていくことが望まれる。

### 8.2 多言語対応

実習ツールは多言語化をサポートしており、今回は日本語版のみの開発であったが、英語版などの開発も比較的簡単に行える仕組みを持っている。他の言語での利用も行えるようにすることで、より多くの開発者のスキル向上につなげることができる。