Vol.2012-DPS-150 No.25
Vol.2012-CSEC-56 No.25
2012/2/29

高校生のウェブアプリケーション脆弱性体験学習に関する一考察

増山　一光[†,††]　　佐藤　直[†]

神奈川総合産業高等学校において独立行政法人情報処理推進機構（IPA）が公表している脆弱性体験学習ツール AppGoat を用いてウェブアプリケーションの脆弱性に着目した情報セキュリティ教育を行った．本稿では，この教育実践を通じて生徒らがウェブアプリケーションの脆弱性を認識する過程について検証を行なった．その結果，一般的な情報セキュリティ能力の向上だけではなく，システム開発者としての基礎的な能力が醸成できることが確認できた．さらに，この結果を踏まえて，情報システムの脆弱性対策を中心とした高等学校における情報セキュリティ教育の可能性について検討する．

A consideration of experiential education for high school students on web application vulnerabilities

Kazumitsu Masuyama[†,††]　and Naoshi Sato[†]

We have practiced an experiential education for high school students on web application vulnerabilities, using AppGoat IPA published. Through the education, we examined how the high school students recognize the vulnerabilities and improve their information security ability and basic skills to develop information systems. And we discuss possibilities and issues of deploying this education in high school.

1. はじめに

初等中等教育における情報モラル教育の内容は大きく分けて 2 つに分けられる[1]．まず，情報社会における正しい判断と望ましい態度を育てるという「心を磨く領域」がある．次に，情報社会で安全に生活するための危険回避の方法の理解やセキュリティの知識･技術,健康への意識という「知恵を磨く領域」がある．さらに，その項目として次のような 5 つが示されている．

1.情報社会と倫理
2.法の理解と遵守
3.安全への知恵
4.情報セキュリティ
5.公共的なネットワーク社会の構築

本稿においては，このような初等中等教育における情報モラル教育を踏まえて後期中等教育である高等学校における情報セキュリティ教育に着目することにする．情報セキュリティ教育を展開するときに，情報セキュリティインシデントを解説しその対策を理解させることは可能であっても，これを安全な環境で体験させてその対応を実習として行うことは難しいという問題が存在する．これは，情報セキュリティインシデントの再現性が困難であり，仮に再現できたとしても学習者である生徒らを危険な環境にさらしてしまうという懸念があるためである．

実際に，このような実習を行なうことができれば，情報セキュリティインシデントに対する確かな能力と適切な態度の育成が可能となる．そのため，情報セキュリティに関する学習ツールを用いることを検討した．このような学習ツールとしては，フィッシング詐欺対策のフィッシングフィル[2]や 5 分でできる！情報セキュリティポイント学習[3]などがある．前者はフィッシング詐欺サイトを見破る技術を身に付けるための URL 判別ゲームであり，危険な URL を判別する能力を身に付けることはできるが，情報セキュリティインシデントの全体的な理解ができるような内容ではない．後者は，副題に「～事例で学ぶ中小企業のためのセキュリティ対策～」とあるように，中小企業における経営者，管理者，一般社員を対象にした情報セキュリティ学習ソフトであり高校生にとっても有用な学習ソフトであるが，情報セキュリティインシデントを再現し技術的な学習ができる内容は含まれていない．

そこで，本稿では脆弱性の発見，プログラミング上の問題点の把握，対策手法の学習を対話的に実施できる脆弱性体験学習ツール AppGoat[4]を利用して情報セキュリティ教育の実践の試みる．このような学習ツールを用いることで，具体的な情報セキュリティインシデントの発生する技術的な原因と対策を理解させると共に，その背後に存在する情報モラル上の課題についても明確にさせる．

2. 研究目的

本稿における研究目的は，脆弱性体験学習ツール AppGoat を用いて，高校生がどのようにして脆弱性を認識していくのかという過程に着目しつつ，この教育実践におい

† 情報セキュリティ大学院大学
Institute of Information Security
†† 神奈川総合産業高等学校
Kanagawa Sogo Sangyo High School

Vol.2012-DPS-150 No.25
Vol.2012-CSEC-56 No.25
2012/2/29

て情報セキュリティインシデントの発生の原因と対策の学習が具体的な情報セキュリティに関する能力の向上と適切な情報モラルの形成に寄与したかを検証する．

受講生徒は，AppGoat を用いてウェブアプリケーションの脆弱性について学習し，実際の脆弱性を見つけて，その脆弱性から何が起きるのかを把握するまで，グループによる協働学習活動により試行錯誤を繰り返しつつ問題解決するよう指示した．この過程を分析することで，生徒が身に付けた情報セキュリティに関する能力や情報モラルの特質について明らかにする．

なお，情報セキュリティ教育にはディフェンス面（防御面）とアタック面（攻撃面）がある．AppGoat を用いた情報セキュリティ教育では脆弱性によって生じる問題点を明らかにすることからセキュリティ攻撃を学習することにもなる．高等学校では，こうしたセキュリティ攻撃に関する教育は情報モラルの観点からほとんど行なわれていない．そこで，こうしたセキュリティ攻撃に関する教育実践の適切性も検証する．

3. 講座概要

3.1 概要

AppGoat を利用した情報セキュリティ教育を本校の夏季特別講座の一つとして実施した．本校では脆弱性に着目した情報セキュリティ教育の実施は初めての試みであったため，単位認定を行なう通常の科目としてではなく，希望する生徒が自主的に参加する講座として設定した．講座の実施概要は次のとおりである．なお，本実践の学習環境に関しては，受講人数分の AppGoat をインストールしたパソコンを用意し，インターネットが使用できる状況であった．

実施日程：2011 年 8 月 1 日から 3 日（3 日間）
実施場所：本校　ネットワーク通信実習室
受講人数：7 名（男子 3 名：1 年生 1 名，3 年生 2 名）
　　（女子 4 名：2 年生 1 名，3 年生 3 名）

3.2 脆弱性体験学習ツール AppGoat

本講座で使用した脆弱性体験学習ツール AppGoat は，独立行政法人情報処理推進機構によって 2011 年に公開されたものであり，開発したのは株式会社 フォティーンフォティ技術研究所[5]である．AppGoat は，公開しているサイトからファイルをダウンロードして，そのファイルを解凍してバッチファイルを実行することで学習を始めることができる．実際の講座で使用した際には，他のアプリケーションやシステムに影響を与えることなく使用することができた．

表 1　学習テーマ

○クロスサイト・スクリプティング	
1	クロスサイト・スクリプティングとは
2	アンケートページの改ざん（反射型）
3	掲示板に埋め込まれるスクリプト（格納型）
4	入力情報の漏えい（反射型）
5	ウェブページの改ざん（DOMベース）
6	不完全な対策
○SQLインジェクション	
7	SQLインジェクションとは
8	不正なログイン（文字列リテラル）
9	情報漏えい（数値リテラル）
10	他テーブル情報の漏えい（数値リテラル）
11	データベースの改ざん（数値リテラル）
○CSRF（クロスサイト・リクエスト・フォージェリ）	
12	CSRF（クロスサイト・リクエスト・フォージェリ）とは
13	意図しない命令の実行
14	不完全な対策
○その他	
15	エラーメッセージからの情報漏えい

表 2　学習者の参加動機

学年・性別	参加動機
1年男子	パソコンに興味があり，SEもしくはネットワーク管理者になりたいから
2年女子	後期に学校設定科目「情報セキュリティ」を履修するから
3年男子	コンピュータに興味があり自分の知識を高めたいから
3年男子	ネットワークを学習しており情報セキュリティもその一環で学習したいから
3年女子	情報セキュリティに興味・関心があるから
3年女子	情報セキュリティがこれからの情報社会で必要となる知識だから
3年女子	情報セキュリティについてもっと知識を深めたいこととレベルの高い内容に挑戦したいから

表 3　学習グループの構成

グループ名	人数	構成	平均点
A	2	2年生女子，3年生女子	5.0
B	2	3年生女子2名	4.0
C	3	1年生男子，3年生男子2名	3.6

AppGoat の学習環境は，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の二つがある．今回の講座はウェブアプリケーションに関わることから前者の実習環境を使用した．ウェブアプリケーション実習環境の学習テーマは表 1 の通りである．本講座では，1 日目にクロスサイト・スクリプティング，2 日目に SQL インジェクション，3 日目にクロスサイト・リクエスト・フォージェリとその他を扱った．

3.3 学習者のレディネス

この講座を受講した生徒の参加動機は表 2 の通りである．この表から，受講生徒らの情報セキュリティに関する高い関心が覗える．さらに，希望進路から見て必要である，現在の学習を発展させたい，自らの将来向けた必要性や現在の学習に発展といった動機から受講していることも分かる．

受講者の情報セキュリティの基礎的学習に関しては，1･2 年生が「情報 C」，3 年生が教科工業の「情報技術基礎」において行なっている．前述の学校設定科目「情報セキュリティ」の履修状況については，この科目が 2･3 年生の選択科目であることから，1 年生一人を除いて 3 名がすでに履修済み，残りの 3 名は 2011 年後期に履修している．

本校は単位制専門高校であり，自らが希望する科目を選択することができる．そのため，今回の受講生徒の 3 年生では 10 科目近く履修している生徒もおり，平均的には

Vol.2012-DPS-150 No.25
Vol.2012-CSEC-56 No.25
2012/2/29

5〜6 科目の情報系の科目を履修している．その主な科目内容としては，プログラミング，ハードウェア，ソフトウェア，ネットワーク，情報セキュリティ，３DCG，課題研究など多彩である．こうした本校での学習活動から，脆弱性に関する学習の必要性を強く感じている生徒が多くなっている．

3.4 学習形態

今回の AppGoat を用いた学習では個別学習ではなく，グループによる協働学習により課題解決を目指す形態で実施した．この理由は，個人学習よりもグループ学習の方が脆弱性に対する「気づき」が促進されると考えたからである．

そこで，受講生徒の基礎能力を確認してグループ分けするために，事前に実力テストを実施した．この事前実力テストは大別して HTML，JavaScript，SQL に関する問題で構成し，全 10 問の出題した．その結果，全体の平均点は 4.14 であり，脆弱性を学習するに当たっては十分な学力ではなかった．このため，講座内でも基礎的な内容を復習しながら学習を展開した．

グループ分けに関しては表 3 ように行った．できるだけグループ間の能力差がないように，すなわち，事前実力テストの平均点に大きな差のないようにグループ分けした．なお，次章での学習プロセスの分析において，このグループ名を用いて，それぞれのグループにおける脆弱性の認識の特徴について考察する．

4. 学習フロー

4.1 AppGoat の学習フロー

脆弱性体験学習ツール AppGoat は，脆弱性の原理を理解して，擬似攻撃より脆弱性がもたらす脅威を体験できる．さらに，脆弱性への対策方法を学習することができる．

このため，AppGoat では学習を始める前に図 1 のような注意事項が表示される．こうした注意事項は，脆弱性を学習するに当たっては必要不可欠なものであると同時に，学習者に高いレベルでの情報モラルが求められる．

AppGoat における学習の流れは図 2 の通りである．AppGoat で学習を開始すると，まず，テーマに関する概要説明が表示される．次に図 3 のように脆弱性がなぜ起きるのかについての説明がなされる．この説明は図解により，脆弱性が容易に理解できる工夫がなされている．

脆弱性に関する説明がされた後に，実習用のウェブアプリケーションが表示され脆弱性に対して擬似攻撃を行なうことになる．実際に擬似攻撃が成功すると図 4 のような表示になる．この攻撃に当たっては問題文だけでは脆弱性を特定することが難しい場合もあるので，ヒントが用意されている．このヒントは 2 つ用意されており，1 つ目のヒントは擬似攻撃のポイントを示しており，2 つ目のヒントは実質的な擬似攻撃の手法である解答を示している．

図 1　AppGoat のおける注意事項

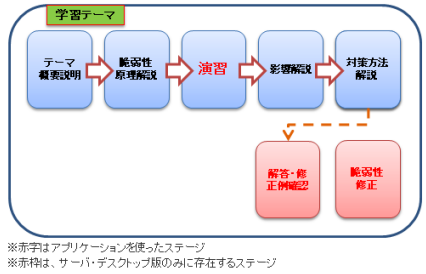

図 2　AppGoat における学習フロー

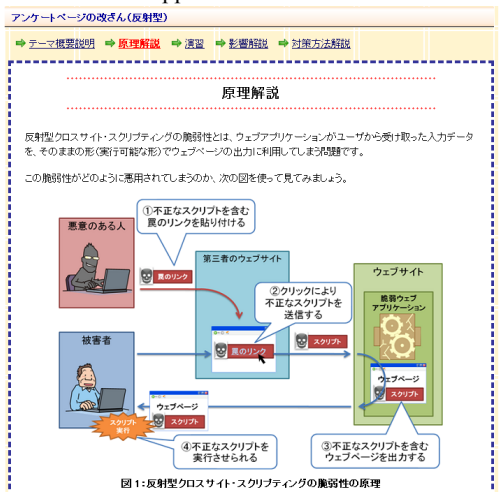

図 3　脆弱性原理解説

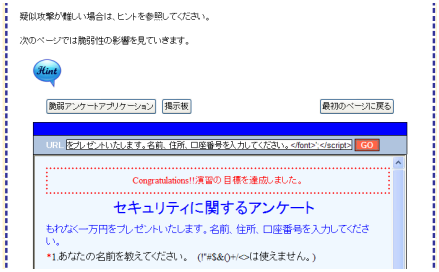

図 4　擬似攻撃の成功時の画面

実習の後には影響と対策方法の解説が行なわれている．影響については，脆弱性に対する擬似攻撃を受けた立場でどのような影響が生じるかを解説している．このような解説を加えることで，アタック側である攻撃者とディフェンス側であるユーザの視点から脆弱性を考えることができる．そして，学習テーマの最後に対策方法の解説が行なわれる．ここでは限られた画面上ではあるが，脆弱性に対する適切な対処法が示されており，安全なウェブアプリケーションを開発する上で不可欠な知識が学べるようになっている．

Vol.2012-DPS-150 No.25
Vol.2012-CSEC-56 No.25
2012/2/29

4.2 本実践における学習フロー

本実践では，基本的に AppGoat における学習フローを踏襲しながら，学習者である高校生が脆弱性に対する学習に関して初学者であることを考慮して学習フローの設計を行なった．具体的な学習フローは図 5 の通りである．

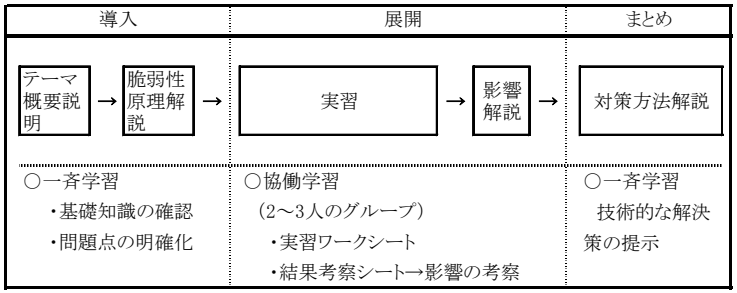

図 5　本実践における学習フロー

まず，学習を進めるにあたって，ウェブアプリケーションの脆弱性に関して教材が必要になる．ここで教材の選定については，高校生が自学自習の可能な教材で，今回利用する AppGoat での学習にある程度準拠している必要がある．そこで，AppGoat においても紹介されている「安全なウェブサイトの作り方」[6]を利用した．

最初の導入の段階では，一斉学習の形態をとり，AppGoat における画面においてテーマ概要説明や脆弱性原理解説を学習した上で，前述の教材を用いて実習目的や事前知識の確認を行なった．特に初学者ということもあり，やや難解な HTML タグや JavaScript の文法を理解していないことがあったので, 基礎知識の確認に関しては重点的におこなった．さらに，脆弱性に対して擬似攻撃を行なうにあたって，何が脆弱性として存在しているのかという問題点の明確化を行った．次の展開では，脆弱性に対する擬似攻撃を行った．ここではグループによる協働学習を中心に行ない，操作上のトラブルが発生しない限りには特段の教授は行っていない．

AppGoat には，各学習テーマの履歴を表示する機能はあるが，個別の実習における学習プロセスに関する履歴を蓄積する機能は有していない．そのため，生徒が試行錯誤をしながら擬似攻撃をするにあたって，そのプロセスを把握させるために実習ワークシートを用意し，記録させるようにした．このワークシートでは，擬似攻撃を行なった内容とその結果を記録させた．これにより客観的な視点から擬似攻撃における因果関係を考察させるようにした．

このような活動を通じて擬似攻撃が成功した際には，結果考察シートに擬似攻撃によってどのようなことができるようになったのかと，なぜそのような結果が持たされたのかを記述させている．このシートの役割としては，高校生による脆弱性の検証に関しては攻撃ができればよいという短絡的な解答の求め方になりがちであるので，こうしたことを是正し何か起きたのかを客観的に考えさせる．さらに，グループで記述したシートと，AppGoat における影響解説を比較して，具体的な脆弱性について検証する．

最後のまとめでは，一斉学習の形態で対策方法の解説を行なった．具体的には，AppGoat による対策方法解説の画面にて概要を学習した上で，教材を用いて技術的解説を行なった．この段階で，講座自体の時間的制約から，対策手法に関する実習を行なうことができなかったが，クロスサイト・スクリプティングや CSRF については多くのサイトで対策がとられているので，これらに関しての検証を行なった．

4.3 クロスサイト・スクリプティングの実践例

ここではクロスサイト・スクリプティングにおける不完全な対策に関する実習を実践例として取り上げ，生徒らの脆弱性を認識するプロセスについてみることにする．

問題文は「ブラウザ上の入力チェックによるクロスサイト・スクリプティングの対策を回避して，【脆弱掲示板】にスクリプトを送信してみましょう.」となっており，ヒントは「次の URL を完成させ，ブラウザのアドレス欄に入力してみましょう．http://localhost/Web/Scenario106/Vulsoft/bbs.php?name=1&title=1&url=&content=<ここに掲示板の内容を書き換えるスクリプトを入れてください>」となっている．なお，脆弱掲示板（一部）は図 6 の通りである．

表 4 は受講生徒が実習の際に記入したワークシートをまとめたものである．この実習ではヒントを参考にして掲示板の改ざんを行なうが，受講生徒はこれまでにアンケートページの改ざん（反射型）や掲示板に埋め込まれるスクリプト（格納型）といった学習テーマに取り組んでいる．

A グループに関しては，これまでの学習から同様の脆弱性があると判断して掲示板のソースを表示させて，JavaScript の document.getElementById を用いて指定 ID のエレメントを取得する方法で掲示板の内容が改ざんできることに気がついている. しかし，これまでの学習テーマで使用した innerTEXT はテキストの書き換えを行なえるが，HTML の書き換えができない. そこで調査を行なって innerHTML に変更することで目標を達成している．

B グループは，当初，<script>タグを用いる方法をあまり理解できておらず，アラートを表示する程度の能力しか身に付けていない状況であった．そこで，これまでの実習をすべて振り返ると同時に，テキストとして配布していた「安全なウェブサイトの作り方」を参考にして脆弱性を発見することができた．このグループはこの実習中によく話し合いを行ない適切なグループワークを実現していた．

C グループはこの実習における脆弱性を理解しており，目標達成に関しては特に問題のない状況であった．このグループは報道されているウェブ改ざんのニュースでは

Vol.2012-DPS-150 No.25
Vol.2012-CSEC-56 No.25
2012/2/29

表示を目立つようにしているということから工夫を加えている．工夫の内容は改ざんした文字のフォントを変更するという比較的簡単なものであるが，これを実行できるためには JavaScript や HTML に関する基礎的な能力を有している必要がある．

クロスサイト・スクリプティングに対する根本的な対策（例．エスケープ処理）は基本的にサーバ側で行なわなければならないが，リソースの制約からサーバでの対策は行なわなかった．代わりに，エスケープ処理に関して机上で説明した．

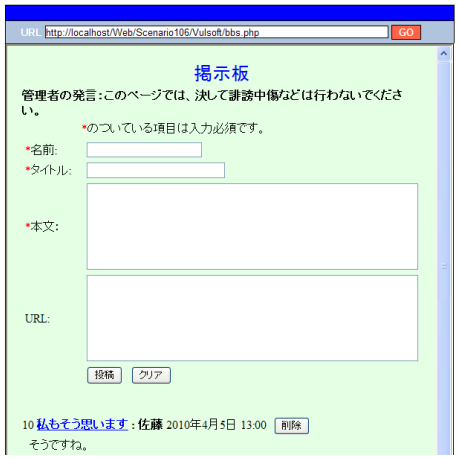

図 6　脆弱性掲示板（一部）

表 4　クロスサイト・スクリプティングの学習プロセス

Aグループ

回数	実行内容	結果
1	http://localhost/Web/Scenario106/Vulsoft/bbs.php?name=1&title=1&url=&content=<script>document.getElementByld("warning").innerTEXT="誹謗中傷してね";</script>	Web上の文字はどこも変化しなかった
2	http://localhost/Web/Scenario106/Vulsoft/bbs.php?name=1&title=1&url=&content=<script>document.getElementByld("warning").innerHTML="誹謗中傷してね";</script>	管理者のメッセージを変更することができた

Bグループ

回数	実行内容	結果
1	http://localhost/Web/Scenario106/Vulsoft/bbs.php?name=1&title=1&url=&content=<script>この掲示板のみ誹謗中傷を許可します．</script>	URLに入力したが失敗
2	http://localhost/Web/Scenario106/Vulsoft/bbs.php?name=1&title=1&url=&content=<script>document."この掲示板のみ誹謗中傷を許可します．";</script>	変わらない
3	http://localhost/Web/Scenario106/Vulsoft/bbs.php?name=1&title=1&url=&content=<script>document.getElementByld("warning").innerHTML=この掲示板のみ誹謗中傷を許可します．";</script>	目標達成

Cグループ

回数	実行内容	結果
1	http://localhost/Web/Scenario106/Vulsoft/bbs.php?name=1&title=1&url=&content=<script>document.getElementByld("warning").innerHTML='存分に罵り合ってください!!　Fight!!';</script>	管理者の発言が改ざんできる

4.4　SQL インジェクションの実践例

ここでは SQL インジェクションによる情報（数値リテラル）漏えいの実習を取り上げ，生徒が脆弱性を認識するプロセスについてみることにする．

問題文は「【脆弱オンラインバンキングアプリケーション】に対し，入出金履歴照会ページから SQL 文を構成する文字列（UNION 句を用いた問い合わせ文）を送信します．入出金履歴照会ページで全てのユーザのログイン ID とパスワードを表示させることを目標に，進めていきましょう．」となっている．

擬似攻撃の手順は次の通りである．

①脆弱オンラインバンキングアプリケーション】にログインして，入出金履歴照会ページが正常に動作することを確認します．ログイン ID「yamada」，パスワード「P@ssword」を利用してログインしてください．

②前提条件を参考に，ユーザテーブルからログイン ID とパスワードを取得する SQL 文を作成します．

③全てのユーザのログイン ID とパスワードが表示されることを確認します．

ヒントは「（手順②）口座番号を表すパラメータ「account_id」が SQL 文を構成するために使われています．」と「（手順②）URL のパラメータに値を入力して次のような SQL 文を実行させてみましょう．SELECT account_history.trade_date, account_history.trade_type, account_history.note, account_history.money_in, account_history.money_out, account_history.balance FROM account, account_history WHERE account.id = 'ユーザ ID' AND account.account_id = account_history.account_id AND account_history.account_id = 好きな数字 UNION SELECT NULL,●●,■■,0,0,0 FROM user」になっている．なお，脆弱オンラインバンキングアプリケーション（一部）の開始画面は図 7 の通りである．

表 5 は受講生徒が実習の際に記入したワークシートをまとめたものである．この実習では，SQL に関する知識を踏まえて，手順，前提条件，ヒントなどを参考にして脆弱性を発見することになる．

A グループと B グループについては，前回の学習ですべてのユーザの口座残高を抽出する実習を行なっており，そこで「account_id=99 OR 1=1」を含む URL を利用しており，今回の実習でも利用できると考えてアプローチしている．しかしながら，今回は SELECT 文の統合する UNION 句を利用したアプローチであることから，user テーブルからデータを抽出して入出金履歴照会処理に統合できるかという視点が必要になる．これに関しては，再度，前提条件とヒントを確認して uesr テープルの構成を確認して目標達成にいたっている．

C グループは，実習に取り組んだ早い段階でその目指している目標を理解していた．具体的には，「yamada」の ID でウェブアプリケーションの動作を確認して，page=4 である入出金履歴照会処理のページでユーザ ID とパスワードを表示するためにはどのようにすればよいのかを考察していた．そこで，ヒントが入出金履歴照会処理の形式であることに気づき，ここに user テープルからデータを抽出することで脆弱性に対する擬似攻撃を成功させていた．

今回の実習では，SQL に関する基礎的な知識不足から，この脆弱性に関しては理解できなかった生徒もいた．さらに，この実習手順や前提条件を読み解く力が試されており，こうした学力を身に付けなければならないことが課題でもあった．

Vol.2012-DPS-150 No.25
Vol.2012-CSEC-56 No.25
2012/2/29

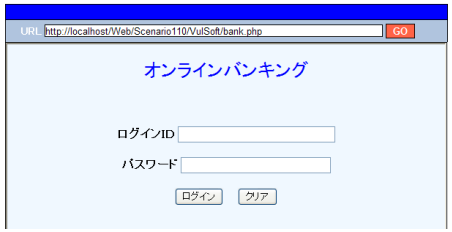

図 7　脆弱オンラインバンキングアプリケーション（一部）

表 5　SQL インジェクションの学習プロセス

Aグループ

回数	実行内容	結果
1	http://localhost/Web/Scenario110/VulSoft/bank.php?page=4&account_id= or 1=1 UNION id,name FROM user	失敗
2	http://localhost/Web/Scenario110/VulSoft/bank.php?page=4&account_id= or 1=1 UNION id,password FROM user	失敗
3	http://localhost/Web/Scenario110/VulSoft/bank.php?page=4&account_id= or 1=1 UNION SELECT NULL,id,password,0,0,0 FROM user	失敗
4	http://localhost/Web/Scenario110/VulSoft/bank.php?page=4&account_id=99 UNION SELECT NULL,id,password,0,0,0 FROM user	成功

Bグループ

回数	実行内容	結果
1	http://localhost/Web/Scenario110/VulSoft/bank.php?page=4&account_id=1000001 UNION SELECT NULL,or 1=1,or 1=1,0,0,0 FROM user	エラー表示
2	http://localhost/Web/Scenario110/VulSoft/bank.php?page=4&account_id=21 UNION SELECT NULL,account,account_history,0,0,0	エラー表示
3	http://localhost/Web/Scenario110/VulSoft/bank.php?page=4&account_id=21 UNION SELECT NULL,id,password,0,0,0 FROM user	目標達成

Cグループ

回数	実行内容	結果
1	http://localhost/Web/Scenario110/VulSoft/bank.php?page=4&account_id=0 UNION SELECT NULL,id,password,0,0,0 FROM user	他人の情報が見れるようになる

4.5　CSRF の実践例

ここでは CSRF（クロスサイト・リクエスト・フォージェリ）における意図しない命令の実行に関する実習を取り上げ，生徒らの学習プロセスから脆弱性を認識するプロセスについてみることにする．

問題文は「意図しないリクエストを送信させる罠のリンクを作成し【掲示板】に投稿します．【脆弱 SNS アプリケーション】にログインしたまま【掲示板】の罠のリンクをクリックすることで，【脆弱 SNS アプリケーション】の個人情報公開設定を変更することを目標に，演習を進めていきましょう．」となっている．

擬似攻撃の手順は次の通りである．

①【脆弱 SNS アプリケーション】にログイン名「yamada」，パスワード「P@ssword」でログインして，設定変更ページに移動し，個人情報公開設定が「非公開」になっていることを確認します．

②HTML ソースを閲覧して，その情報をもとに個人情報公開設定を「非公開」から「公開」に変更するリクエストを作成します．

③　②で作成したリクエストを送信させる罠のリンクを作成し，【掲示板】に投稿します．

④罠のリンクをクリックすることによって，個人情報公開設定が「非公開」から「公開」になってしまうことを確認します．

ヒントは「（手順②）HTML ソースを閲覧して，GET メソッドで送信されるデータを確認してみましょう．」となっている．なお，脆弱 SNS アプリケーション（一部）の開始画面は図 8 の通りである．

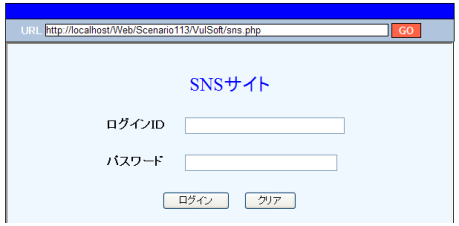

図 8　脆弱 SNS アプリケーション（一部）

表 6　CSRF の学習プロセス

Aグループ

回数	実行内容	結果
1	山田さんのIDとパスワードで入って,公開したときのURL掲示板のリンクに埋め込んだ.	非公開にしていたものが公開になった

Bグループ

回数	実行内容	結果
1	http://localhost/Web/Scenario113/VulSoft/sns.php?page=4&public=1	目標達成

Cグループ

回数	実行内容	結果
1	公開が&public=1，非公開が&public=0である	個人情報が見れるようになってしまう

表 6 は受講生徒が実習の際に記入したワークシートをまとめたものである．まず，生徒らは手順に従って，脆弱 SNS アプリケーションに「yamada」という ID でログインをして，公開と非公開の状態を確認していた．各グループとも，これまで脆弱性に関する学習を重ねてきていることから，状態の変化を的確に把握することでできており，実際に公開と非公開の過程で URL にその状態が表示されていることが理解できている．

各グループの脆弱性発見についてみると，A グループは表示されている手順を忠実に再現して脆弱性を再現している．同様に B グループも公開の状態から URL を導き出し目標が達成できている．C グループは，いち早く公開の状態と非公開の状態が URL に表示されていることを突き止めており，それらの状態を正しく把握して目的を達成している．

この実習から，専門的な知識がなくても，ウェブアプリケーションを利用する過程を観察することで脆弱性を発見できることを学習できた．クロスサイト・スクリプティングの際も同様に，多くの生徒が，URL に注意しないことで個人情報の漏えいにつながることを改めて認識できた．

Vol.2012-DPS-150 No.25
Vol.2012-CSEC-56 No.25
2012/2/29

5. 分析と考察

5.1 アンケート結果の分析

表 7 のアンケートは受講者に対して，質問に最も高い回答（とてもそう考える）を 5 とし，逆に最も低い回答（全くそう考えない）を 1 とした 5 段階評価法を用いて行ったものである．まず，AppGoat の利用に関しては学習意欲の向上がもたらされ，積極的な学習活動を行なったことがわかる（番号 2，6）．しかしながら，AppGoat の活用や学習内容の理解に関しては十分ではないことがわかる．

表 7　受講生徒のアンケート結果

番号	質問	平均値	標準偏差	番号	質問	平均値	標準偏差
1	AppGoatの使い方は理解できた.	3.86	1.07	16	CSRFを理解することができた.	3.86	1.07
2	AppGoatを使用することで学習意欲が向上した.	4.43	0.79	17	CSRFの実習をきちんと行うことができた.	3.71	0.49
3	AppGoatを使用することで課題ごとの学習目標が明確になった.	3.29	1.25	18	CSRFへの対策を理解することができた.	3.57	0.98
4	AppGoatの機能を十分活用することができた.	3.43	1.27	19	本講座での学習を通じて，情報モラルの必要性を感じますか.	4.86	0.38
5	AppGoatの実習内容は満足できるものであった.	3.86	1.35	20	本講座での学習を通じて，技術の適切な利用が大切であると感じますか.	4.86	0.38
6	本講座の学習に全体を通じて積極的に参加することができた.	4.29	0.95	21	本講座での学習を通じて，集中して取り組むことができた.	3.86	1.07
7	実習を行うにあたって，グループを意識して取り組むことができた.	3.71	0.95	22	本講座での学習を通じて，教員の説明が理解でき実習に活かすことができた.	4.29	0.76
8	主体的かつ自主的に実習を行うことができた.	3.86	0.90	23	本講座での学習を通じて，課題解決に向けて努力することができた.	3.71	1.50
9	実習ができたときに達成感が得られた.	3.86	1.46	24	本講座で学習したことは，私自身にとって大切な知識である.	4.86	0.38
10	クロスサイトスクリプティングを理解することができた.	3.71	0.95	25	本講座で学習したことは，情報セキュリティの知識として役立つ.	4.71	0.49
11	クロスサイトスクリプティングの実習をきちんと行うことができた.	3.86	0.69	26	本講座で学習したことは，他の情報系の授業にも役立つ.	3.57	1.51
12	クロスサイトスクリプティングへの対策を理解することができた.	3.57	0.98	27	本講座で学習したことは，たのしかった.	4.29	1.11
13	SQLインジェクションを理解することができた.	3.71	0.95	28	本講座に対して参加するに当たって，理解できるかどうか不安であった.	4.14	0.69
14	SQLインジェクションの実習をきちんと行うことができた.	3.86	0.69	29	本講座の課題に取り組むに当たって，できるかどうか心配であった.	3.71	1.11
15	SQLインジェクションへの対策を理解することができた.	3.57	0.98	30	本講座で得た知識を適切に利用する心構えを持つことができた.	4.43	0.79

これに関しては，AppGoat の取り扱っている脆弱性の学習内容が初学者である高校生にとっては高度であったことが影響している．具体的には,学習内容であるクロスサイト･スクリプティング（番号 10～12），SQL インジェクション（番号 13～15），CSRF(番号 16～18)についてはどの項目ほぼ同じような値を示す結果となっている．特に，それぞれの学習内容に関する対策についての理解が低かった．この原因として，対策については実習が伴わず理解不足になったためと考えている．

情報モラルに関しては，講座全体を通じて必要性が理解できており，技術の適切な利用に関しても把握できている（番号 19，20）．これは，今回の脆弱性体験学習が技術に立脚した情報モラルを問う内容でもあったことから，受講生徒らは講座の趣旨を十分に理解していた．加えて，情報セキュリティに関する知識を得られたことによる講座の効果を認識している（番号 25）．

5.2 受講生徒の感想の分析

表 8 は受講生徒による感想の一部抜粋である．受講生徒らに共通に見られるのは，情報セキュリティインシデントが身近に存在しているという認識である．これは AppGoat で擬似攻撃が成功することにより，ウェブアプリケーションが抱える脆弱性を実体験として知ることができたからである．このように脅威を直視させることで，情報セキュリティ意識が形成され，自ら情報セキュリティ対策に役立てようとしていると考えられる．

なお，受講生徒の多くが本校における情報系の科目を学習しており,プログラミングやシステム開発の基礎を学習している者が多い．そのため，今回の講座が開発者として情報モラルや情報セキュリティの育成にも役立っていることが覗える．具体的には，自分が作成したウェブアプリケーションで脆弱性がないかを確認するや，脅威に立ち向かう創意工夫が確認できた，などの感想が示されている．

また，受講生徒らの感想から，今回の講座に対する満足度が高いことがわかる.これは，受講生徒が高等学校における教育のなかで，初めて脆弱性に関する学習に取り組んだことによって興味･関心が高くなったことと，ウェブアプリケーションの脆弱性の学習活動を通じて情報システムに関する多角的な学習ができたことが要因となっている．このことから，脆弱性体験学習のような教育実践は，高校生に対する情報モラルや情報セキュリティに関する動機付けと技術の向上を目指すためにも有効であると考えられる．

表 8　受講生徒の感想（一部）

1年男子	最近のWebアプリケーションやWebサイトに，今回の実習のような脆弱性があると大変危険だとわかりました．万が一自分の作成したプログラムにこのような脆弱性がないかを確認するために，この講座で得た知識を役に立てたいと思います.
2年女子	ほとんど知識がないまま参加したので，いまだに理解はあと一歩だなと思うところがある．しかし,参加して本当によかったと思う．後期に情報セキュリティの授業をとっているので,今回知ったことをAppGoatで復習し，授業で得た知識を補強して理解を深めたいと思いました.
3年男子	自分の身近なところにも情報セキュリティインシデントが起こる危険性が潜んでいたり，悪意のある人々や脅威に立ち向かおうとする人々の創意工夫があるということが印象に残りました.
3年男子	この実習をしてみて，こういう風にしてプログラムを実行させることができ，相手をコントロールできるんだと知りました．この実習を通じて安易に危険なWebページを開いてはいけなと思いました.
3年女子	今回の講座で，何度か失敗したり，エラーともたくさん遭遇しましたが，擬似攻撃が成功した時，本当に恐ろしいなと思いました.
3年女子	受講前,全くこの分野に対しての知識がなく，実習のときついていけるかどうかと思ったか，何とか終えられてよかった．たとえ一部だとしても,今回学んだことを少しでも自分の情報を守るために活かしていけるようにしていきたいし，当たり前だが悪用はしないように気をつけて生きたいと思った.
3年女子	今回この講座をやってみて，何も対策を行なっていないWebサイトは本当に危険だと思いました．対策としてはエスケープ処理，暗号化とかセッションIDを乱数で表すなどいろいろ学ぶことができたので，将来Webサイトを作るときに役立てたいと思います.

Vol.2012-DPS-150 No.25
Vol.2012-CSEC-56 No.25
2012/2/29

6. 脆弱性体験学習の課題

6.1 高等学校での脆弱性体験学習の可能性

今回の実践では，ウェブアプリケーションに関する脆弱性体験学習を試みとして行ってきた．システム開発教育の視点から考えれば，ウェブアプリケーションの脆弱性を学習するに当たっては，その対象であるウェブアプリケーションに関する開発能力を有しているか，もしくは学習途上にあることが求められるはずである．

現行の高等学校の教育課程では，システム開発教育が十分には行なわれていないために，その延長上に脆弱性対策に関する教育を実現させることは難しいのが現状である．そのため，こうした教育アプローチとしてはシステム開発を中心とする情報系の大学等で実践することの方が現実的である．

しかし，高校生に対して情報セキュリティインシデントをより具体的に学習させるという視点からは，脆弱性体験学習は有効である．高等学校段階における情報セキュリティ教育の主眼は安全教育に置かれおり，脆弱性体験学習はフィッシング詐欺を未然に防いだり，危険なウェブアプリケーションを判別させるのにも有効である．

ここで考慮しなければならないことは，今回の実践で活用した AppGoat のような学習ツールを高等学校の現場でどのように扱うかという点である．たとえば，脆弱性学習が生徒一人ひとりに行なうことが難しければ，教員が一斉教育で脆弱性を紹介することもできる．また，あるテーマを限定して脆弱性に注目して，実習型授業として展開することも可能である．いずれにしてもこうした学習ツールを利用した授業展開を行なうためは，学校や生徒の特質を考慮した学習計画が必要となる．

6.2 脆弱性体験学習と情報モラルの関連性

今回実施した脆弱性体験学習は，基本的にウェブアプリケーションの脆弱性を擬似攻撃することで明確にして，その原理と対策を学ぶものである．本校での実践するにあたり考慮したことは，高校生に対してウェブアプリケーションへの攻撃手法を教授することが適切な教育活動であるかという点であった．そのため，本実践においては，十分な事前指導を通じて講座の趣旨を理解させ，受講者の学習履歴や参加動機などから，学習する内容を不適切に用いないことをあらかじめ確認してから実施している．

しかし，一方では，セキュリティ攻撃の授業展開はその本質を理解するために有用ではあるが，発達途上にある高校生がこれらを悪用してしまう可能性は拭いきれないので，高校生に対して脆弱性体験学習は適切ではないと考えることもできる．

脆弱性体験学習には，情報モラルとの関連からこのようなジレンマが伴う．従って，こうしたジレンマをどのようにして解決していくかが課題となる．一つの方策としては，脆弱性のもたらす影響を生徒らに考えさせることであろう．具体的には，システム開発者や被害者の立場になって，こうした脆弱性で何が問題でどのようなことが発生するのかを考えさせ，自らの実体験から影響を正しく認識させることが重要であると考えられる．

そして，脆弱性体験学習を実施するにあたっては，生徒らのなかにそのような情報モラルが形成されているかを事前に十分検証する必要がある．そのため，検証ができない場合や不適切な情報モラルが形成されているような場合には安易に実習させるべきではなく，教員が AppGoat のようなツールを使って脆弱性や脅威を理解させるだけでも学習目的を達成できると考えられる．

7. まとめ

本稿では，高校生に対して脆弱性体験学習ツール AppGoat を用いた情報セキュリティ教育の実践を行い，脆弱性に関する学習プロセスに注目し，情報モラルや情報セキュリティに関する態度や能力が形成されているかの検証を行なった．

その結果，受講生徒は擬似攻撃を通じて脆弱性を体験することで，情報セキュリティインシデントが身近で発生しうることを認識し，情報セキュリティ意識の形成が促されることも分かった．さらに，開発者としての情報モラルの重要性について理解が深まることも分かった．こうした脆弱性体験学習は有効である反面，学習経験を生徒が悪用してしまう可能性もある．そのため，対象となる生徒の状況などを把握した上で，綿密な学習計画を立案して実施する必要がある．

今後は，プログラム教育の一環として，AppGoat のサーバ・デスクトップアプリケーション実習環境を利用して，バッファオーバーフローや整数オーバーフローといった脆弱性について学習をさせたいと考えている．

参考文献

1) 日本教育工学振興会:すべての先生のための「情報モラル」指導実践キックオフガイド, pp.4-5 (2007)
2) フィッシング対策協議会:フィッシングフィル.http://www.antiphishing.jp/phil/
3) 独立行政法人情報処理推進機構:5 分でできる！情報セキュリティポイント学習.
http://www.ipa.go.jp/security/vuln/5mins_point/index.html
4) 独立行政法人情報処理推進機構:脆弱性体験学習ツール AppGoat.
http://www.ipa.go.jp/security/vuln/appgoat
5) 株式会社フォティーンフォティ技術研究所:開発者向け脆弱性実習ツールの開発.
http://www.ipa.go.jp/security/vuln/appgoat/appgoat_abst.pdf
6) 独立行政法人情報処理推進機構:安全なウェブサイトの作り方.
http://www.ipa.go.jp/security/vuln/documents/website_security.pdf