「攻撃者目線」で学べるシステムセキュリティ実践的学習環境の提案

中島 滉介　伊藤 恵　新美 礼彦

本研究では, 脆弱性を含むシステムセキュリティ学習用のシステムを提案する. そのシステムに学習者自身が攻撃を加えることによって,「攻撃者目線」からシステムセキュリティを学べるような実践的学習環境の提案を目標とする. システム開発において, セキュリティを意識した開発を行なうことはとても重要なことである. 近年, 情報システムに対する脅威は多様化しており, 開発者は開発対象のシステムを様々な脅威を防ぐように構築しなければならない. しかしそういったシステムを開発するには開発者自身にセキュリティに関して相応の知識や, それに伴う技術力が必要になってくる. システムセキュリティを攻撃者目線から学ぶことで, 攻撃の仕組みをより詳しく学ぶことができると考えられる. この研究により学習者のセキュリティに対する意識を高められたり, セキュアなコーディングができるようになることを目指す.

In this study, we propose a system for learning of system security, including the vulnerability. This study's goal is to propose learning tool that learner learns system security like "hacker" with learner's attack to this system. In system development, it is important to consider security during development. Recently, threats to the information system have been diversified. So developers have to develop system defending against these threats. However, it is need the knowledge of security and high skill level to the developer to develop system so. By learning system security form actually attacking a system, we think that learner can learn principle of threats more. We aim at to learn system security, and to do secure programming for learner by this study.

1　はじめに

1.1　背景

コンピュータウイルス不正アクセスなどの脅威は年々減少傾向にある. しかし一方で, 手口が高度化し, システムセキュリティの知識が不足している技術者では対応が困難である.IPA（独立行政法人 情報処理推進機構）が発行した 2012 年度コンピュータウイルス・不正アクセス届出状況および相談受付状況 [1] によると,2004 年度から不正アクセスの届出は減少傾向にあることがわかる.(図 1)　しかしその届出種別をみてみると,2011 年度多かった「アクセス形跡（未遂）」の項目件数が大幅に減り, その代わりに「侵入」,「なりすまし」の項目件数が増えている.(図 2) これにより, 今までのセキュリティで防げていた脅威が防げなくなってきていることがわかる. 被害内容としては,「ウェ

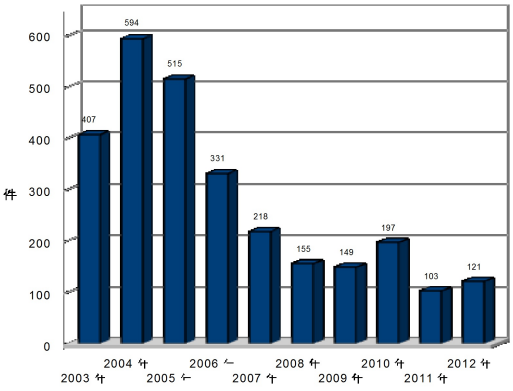

図 1　不正アクセス届出件数の推移

Proposal of An Environment for Practical System Security Learning From the Viewpoint of "Hacker"

Kousuke Nakashima, Kei Ito, Ayahiko Niimi, 公立はこだて未来大学システム情報科学部, Dept. of Information and Computer Science, Future University Hakodate.

届出種別	2012年	2011年
侵入	52	39
メール不正中継	1	1
ワーム感染	0	0
DoS(サービス妨害)	8	5
アドレス詐欺	0	0
なりすまし	32	25
不正プログラム埋込	10	5
その他(被害あり)	2	0
アクセス形跡(未遂)	6	21
ワーム形跡	0	0
その他(被害なし)	10	7
合計 (件)	121	103

図 2　不正アクセス届出の種類

ブサイトの改ざん」,「踏み台として悪用」,「オンラインサービスの不正利用」が多い. システムセキュリティが不十分であると, そのシステムを利用している人だけではなく, 全く関係のない人にまで被害が及んでしまう可能性がある. こういった被害を防ぐためにも, システム開発者はセキュアなシステムを開発する必要がある.

システムセキュリティを学ぶ方法として, 様々な企業が開催しているセキュリティセミナーに参加する方法が考えられる. この方法ではセミナーを開く企業はシステムセキュリティに対して詳しい場合が多く, より実践的なセキュリティ知識を学ぶことができる. しかし, 参加するのに参加費が必要であったり, 開催日時や場所の関係で学生では参加することが容易でない場合がある. 他にもインターネットや書籍から情報を集め学習する方法があると考えられる. この学習方法は多様な情報を手軽に集められる反面, 情報の取捨選択が難しい.

1.2　目的

本研究の目的はセキュリティの知識が不十分なシステム開発者に対して, システム上の脅威を体験し, 学ぶことができるツールの提案と構築である. システムセキュリティにおいて, 攻撃を防げなかった場合, サーバに侵入され Web ページを改変されたり, なりすましによる金銭的な被害を受けることもある. システムの開発者はそのようなことが起こらないように, セキュリティ要件をしっかり盛り込み, セキュアなシステムを開発しなければならない. このツールを使用した学習者が, システム開発において, セキュアなシステムを構築できるようになることが本研究の目標である.

2　関連研究

システムセキュリティの学習環境である,WebGoat,AppGoat の 2 つを関連研究として紹介する.

2.1　WebGoat

WebGoat [2] は OWASP コミュニティが開発した Web アプリケーションのセキュリティを学ぶためのツールである.(図 3)　このツールは実際に学習者が用意された脆弱性を含む環境に攻撃を仕掛け, その仕組みを学ぶといったツールになっている. 特徴としては学習者が「攻撃者目線」でシステムセキュリティについて学べるということである. 自分が攻撃者となり, 仕組みを理解したうえで実際に攻撃してみることで, ただ書籍や講義などで学ぶよりも, 攻撃の仕組みについて理解が深まると考えられる. WebGoat では Web アプリケーションに対する脅威について XSS や SQL インジェクションなど, 大きく分けて 19 種類に分か

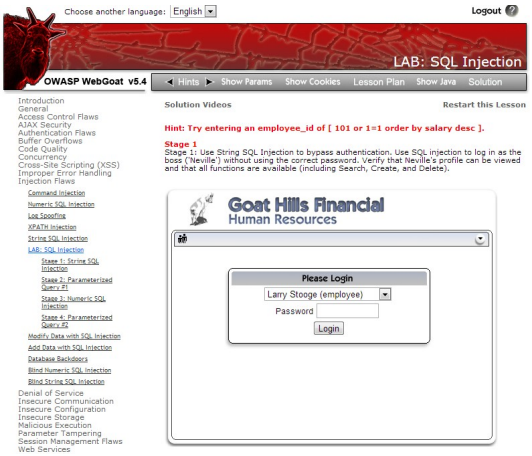

図 3　WebGoat

れており, 脅威について詳しく学ぶことができる.

学習の流れとしては, 学習者はまずどの脅威について学ぶかを選択する. 選択すると課題のページが表示されるので, 実際に攻撃を加えていくという流れになる. 学習者が何をしたらよいかわからなくなった時のために, ヒントを表示させる機能もある. ヒントはいくつかの粒度に分け,, 少しずつ提示していくという工夫もされている. また学習者はヒントだけではなく, このツールの内部で動く Java のソースコードを見ることもできる.(図 4)

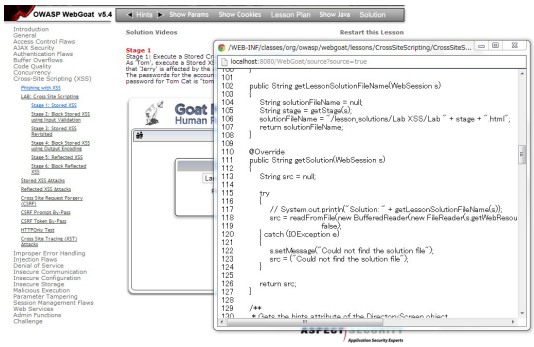

図 4　WebGoat のソースコード

学習ツールとして多くの機能がある WebGoat だが,WebGoat に足りない点として 2 つのことが挙げられる.

1 つ目は「その攻撃によってどういった影響が出るのかという説明がない」ということである.WebGoat ではこの攻撃を受けることでどういった影響があるのかという説明がない. 攻撃を受けることで受ける被害についても正確に知っておくことで, その攻撃の危険性や実際に攻撃を受けたかどうかを判断できるようになると考えられるため, どういった影響を及ぼすかまで, 学習させるべきであると考える.

2 つ目は「WebGoat のソースコードを見ることはできるが, どこに脆弱性があるかまで言及されていない」ことである.WebGoat はソースコードを見れるようにはなっているが, セキュリティに関して知識が不足している人から見ると, どこのコードに脆弱性があるかわかない. ここを改善することで, 具体的にどう対策したらよいか, プログラミングする際にどこに注意したらよいかを学習させることができると考える.

2.2　AppGoat

AppGoat [3] は IPA が開発した脆弱性体験学習ツールである.(図 5) AppGoat は WebGoat と同じように学習者が用意された脆弱性を含む環境に攻撃を仕掛けることで, その仕組みを学ぶといったツールになっている.AppGoat は「ウェブアプリケーション版」と「サーバ・デスクトップアプリケーション版」に分かれており, それぞれ学習できる内容が異なる.

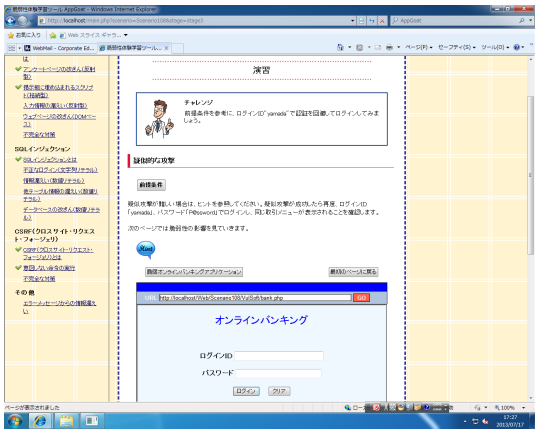

図 5　AppGoat

AppGoat は WebGoat と比べ,「ウェブアプリケーション版」は大きく分けて 4 つ,「サーバ・デスクトップアプリケーション版」は大きく分けて 7 つしかテーマ数がない. その分一つのテーマを複数回に分け, 詳しく学べるような作りになっている. さらに WebGoat よりも攻撃についての説明が詳しい. また WebGoat にはない点として,AppGoat では WebGoat と違い, 攻撃によってどういった影響を及ぼすのかまで解説している.(図 6)

AppGoat の学習の流れとしては, まず学習内容の概要説明, 攻撃の仕組みについて説明した後, 実際に脆弱性を含んだ環境に攻撃を仕掛けるようになっている. 攻撃を仕掛けた後はその攻撃による影響とその攻撃の対策方法を説明している. これが大まかな流れとなっている. これに加え,「サーバ・デスクトップアプリケーション版」では修正例を確認したり, 実際に脆

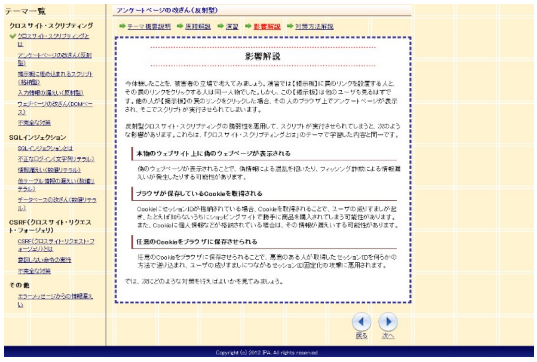

図 6　攻撃による影響の解説

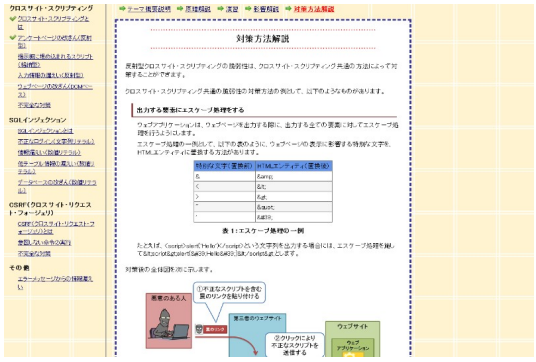

図 7　対策方法の解説

弱性を修正することもできる.

このような学習ツールである AppGoat に不足している点は 2 つある.

1 つ目は「WebGoat に比べ学習できる内容が少ない」ことである. WebGoat が大きく分けて 19 テーマあるのに対し,AppGoat のウェブアプリケーション版は大きく分けて 4 テーマしかない. 多様な Web アプリケーションへの脅威へ対応するためには, 幅広い知識が必要であると考えられる. そうなるとテーマ数 4 つは少ないと考えられる.

2 つ目は「ウェブアプリケーション版では具体的な対策方法が書かれていない」ことである.AppGoat では WebGoat と違い, 脆弱性に対してどのように対策すべきかを解説している. 例えば XSS の対策として,AppGoat ではウェブページを出力する際に要素をエスケープ処理することで対策できるとしている.(図 7) しかしそれは, その攻撃に対する対策であり, 具体的にソースをどのように変更すべきかまでは解説していない. 本研究では具体的にどのように修正すべきかまでを学習者に解説する.

3　提案

本研究ではシステムセキュリティの知識が不足しているシステム開発者に対して, セキュアなシステムを開発できるようになるような, 実践的な学習環境の提案を行う. 学習環境の概要図は図 8 の通りである.　この学習環境において, 仕組みを学ぶだけでなく,「どのように脆弱性をなくすか」までを学習者に学ばせるよ

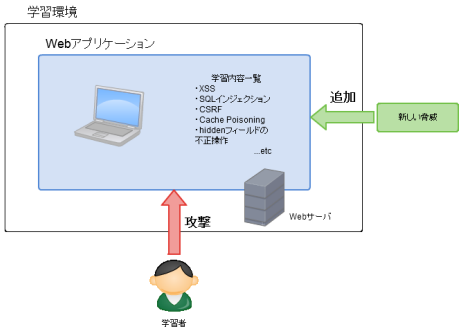

図 8　提案環境の概要図

うなツールを提案する. 具体的には脆弱性がある部分のソースコードを提示し, その部分をどう変更すれば脆弱性がなくなるかまでを学習させる. これにより実践的に学び, 学んだことをどう生かすかまでを学習することができる.

本研究の背景として, 情報システムに対する攻撃の手口が高度化してきていると述べた. 既存の防御方法では防ぎきれなくなってきており, システム開発者はセキュリティに関して常に新しい情報を知っておくことで, それに対応する必要がある. 本研究で提案する学習環境でも, 変化していく脅威に応じて学べる攻撃の種類も追加していけるような仕組みを考える. 攻撃の種類を追加するには, その攻撃を学ぶための情報が必要になる. 本研究では「攻撃の概要」,「攻撃の仕組み」,「脆弱性を含むソースコード」,「攻撃の対策」,「攻撃による影響」について学べる学習環境を提案する. そのため, 新しく攻撃を追加するにはその 5 つの

情報さえあれば, 簡単に攻撃を追加することができる.

3.1 研究スコープ

本研究で提案する学習環境は, システムセキュリティに対しての知識が不十分なシステム開発者を対象としている. 具体的には情報システムを開発する際に, そのシステムにどういった脅威があるのかを知らない, または知っていてもどのように対策すればよいかわからないといった人を考えている. 対象者としては学生や, 経験の浅い SE などが想定される.

また本研究では Web アプリケーションの脅威や脆弱性を学習の対象とする. 学習内容は Web アプリケーション上で考えられる脅威について, その仕組みを学び, 実際にどういった対策を取るべきかを含めて学習できるようにする.

3.2 既存ツールの分析

本研究で提案する学習ツールとして,WebGoat,AppGoat のような既存のツールがある. そこで上で挙げた 2 つの学習ツールを分析し, 足りない部分を考えてみる.

2 つのツールは共に, 攻撃についての説明を読み, ヒントを頼りながら攻撃の仕組みを理解し, その上で, 脆弱性のある環境への攻撃を行っている. このプロセスを経ることで, システムセキュリティについて学ぶことができるとしている. この点についてはただ仕組みを学ぶだけよりも実践的であり, 効果的であると考えられるため本研究でも取り入れるべき点である.2 つのツールに足りない部分として,「どのように脆弱性をなくすか」という点が挙げられる.WebGoat は WebGoat 自体のソースコードを見ることができるが, 実際にどの部分に脆弱性があるのかがわからない.AppGoat はどうしたらその攻撃を阻止できるかを解説してはいるが, ソースコードレベルでの解説はしていない. 以下に各学習環境を比較した図を載せる.

3.3 脆弱性修正の流れ

脆弱性修正の簡単な流れを示す. 単純な例として先ほども挙げた XSS の脆弱性の修正を考えてみる. 図 9 は脆弱性を含んだ PHP のソースコードである. この

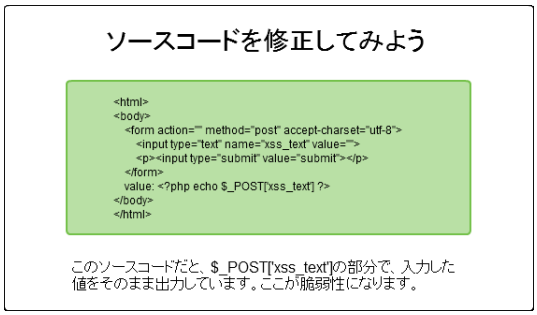

図 10 脆弱性修正前画面

入力フォームに「<script>alert('XSS');</script>」と入力するとスクリプトが実行されてしまう. セキュリティ知識が不足している学習者はこのソースコードを見てもどこに脆弱性があるかがわからない, もしくはわかったとしてもどう修正したらよいかがわからないと考えられる. 本研究では実際に学習者にこうした脆弱性を含むソースコードを提示し, 実際にどう修正すべきかを解説する. 解説した後, 実際に修正したコードを提示することで, 攻撃に対する具体的な対策を学んでもらう. 例の場合だと図 10 のようになる.

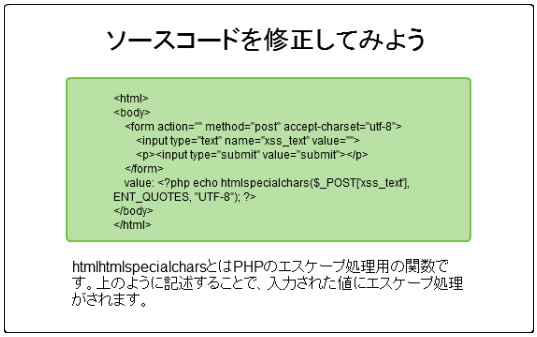

図 11 脆弱性修正後画面

4 まとめと今後の考察

本研究では Web アプリケーションに対する攻撃を「攻撃者目線」から学べる実践的学習環境を提案した. また学習者は脆弱性の含んだ環境に攻撃するだけではなく, その脆弱性を含むソースコードを見て, 脆弱性を修正するところまでを学習範囲とすることで, よ

	攻撃の仕組みの説明	攻撃の種類	脆弱性対策	新しい脅威の対策
WebGoat	簡易な説明しかない	19種類	対策については書かれていない	できない
AppGoat	詳しく書かれている	4種類	対策については書かれているが、具体的ではない	できない
提案環境(予定)	未定	未定	対策について具体的に書いてある	できる

図 9　各学習環境の比較

り実践的な学習環境の提案をした. 今後の課題としてはこの学習環境で具体的に何の脅威について学ぶかを決定することである. 多数ある脅威からどれを学習対象とするかを取捨選択する必要がある. また提案した学習環境を評価する方法について考える必要がある. 脆弱性を含んだ Web アプリケーションを用意し, 学習環境で学習した人, していない人, 既存ツールで学習した人に分けてどれくらい脆弱性を除去できるかを比較し, 評価しようと考えている.

参 考 文 献

[1] IPA(独立行政法人 情報処理推進機構),(2013),「コンピュータウイルス・不正アクセス届出状況および相談受付状況」,http://www.ipa.go.jp/security/txt/2013/documents/summary12all.pdf,2013 年 7 月 11 日アクセス

[2] OWASP(Open Web Application Security Project),「Category:OWASP WebGoat Project」, https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project,2013 年 7 月 11 日アクセス

[3] IPA(独立行政法人 情報処理推進機構),(2012),「脆弱性体験学習ツール AppGoat」,http://www.ipa.go.jp/security/vuln/appgoat/index.html,2013 年 7 月 11 日アクセス