G-56

仮想マシンを活用した NoSQL インジェクションの実践的演習環境の開発

Development of Practical Learning Environment for NoSQL Injection using Virtual Machine

岸本 和理†　　井口 信和‡§
Kazuri Kishimoto　　Nobukazu Iguchi

1. 序論

総務省によると，令和元年の不正アクセスの認知件数は 2960 件であり，前年度と比較して 99%増加している[1]．不正アクセスの要因として脆弱性を含んだ Web アプリケーションの存在があり，Web アプリケーションの脆弱性を悪用した攻撃にはクロスサイトスクリプティング（以下，XSS）やインジェクション攻撃がある．そこで、これまでに被害件数の最も多い XSS の脆弱性と，被害影響度の高いインジェクション攻撃に分類される SQL インジェクション（以下，SQLi）の脆弱性に対して，対策学習の支援を目的としたシステムの開発を進めてきた[2]．

データベースインジェクションには、SQLi の他に NoSQL インジェクション（以下，NoSQLi）があり，これに対する攻撃対策も必要である．NoSQLi は MongoDB などの非リレーショナルデータベースシステム（NoSQL）を不正に操作する攻撃を指し、個人情報の漏洩や認証回避などの不正アクセスが可能である．DZone の The Database Evolution というレポートによると，ビッグデータを活用する企業を対象とした調査では「NoSQL データベースを単体で使用している」，「SQL データベースと NoSQL データベースを組み合わせて使用している」と回答した割合を合わせると 72%となっている[3]．また，NoSQL データベースである MongoDB のシェアは年々増加しており，2021 年 7 月において第 5 位となっている[4]．このことから，Web アプリケーション開発者は，NoSQL データベースに対する攻撃への対策手法を学ぶ必要がある．

NoSQLi を含むインジェクション攻撃への抜本的な対策手法としてセキュアプログラミングがある．しかし，開発者の知識やスキル不足，外部ライブラリや古くからあるコードを引き継いで利用している等の理由により，対策が行われていない Web アプリケーションが存在する．また，インジェクション対策として，クライアントからの入力文字数を制限する，特定のキーワードの使用を不許可とする，Web Application Firewall（以下，WAF）を導入する，などの表面的な対策のみを行っている Web アプリケーションも多くある．これらの Web アプリケーションはインジェクション攻撃への対策が不十分であり，脆弱性を内包している場合がある．

OWASP Top10 によるとインジェクション攻撃は，

† 近畿大学大学院総合理工学研究科，Graduate School of Science and Engineering Research, Kindai University
‡ 近畿大学理工学部情報学科, Department of Informatics, Faculty of Science and Engineering, Kindai University
§ 近畿大学情報学研究所，Cyber Informatics Research Institute, Kindai University

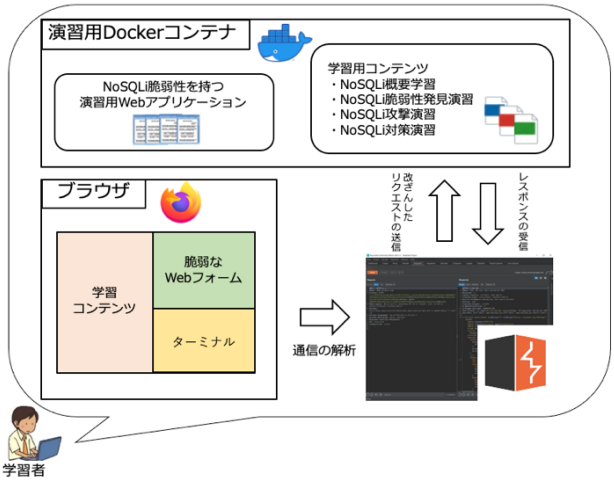

図 1　システム構成図

2010 年度版，2013 年度版，2017 年度版において常に第 1 位となっている[5]．したがって，Web アプリケーション開発者を目指す初学者にとって，実践的なインジェクション攻撃への対策学習は重要である．

我々は，開発中の Web アプリケーションに脆弱性が含まれないようにするためには，攻撃者視点で Web アプリケーションを解析し，施されている攻撃への対策手法や内在する脆弱性を発見するスキルや，そのための知識を習得することが重要であると考えている．

そこで本稿では，Web アプリケーションに対する NoSQLi 対策の実践的な演習を支援することを目的に，脆弱性を発見するという攻撃者視点を取り入れた NoSQLi の実践的演習システム（以下，本システム）を開発する．

本システムは，Docker を用いて演習環境を構築するため，実運用されているネットワークやサーバに影響を及ぼさずに，演習を実施可能である．また，Docker を活用することにより，演習環境の配布が容易になることに加えて，PC の OS に依存することなく演習が実施できる．

学習者は Web ブラウザと Docker 上に構築された演習用 Web サーバとローカルプロキシツールを用いて NoSQLi に関する学習や攻撃者視点を取り入れた演習および対策演習を実施する．

2. 関連研究

高度化，複雑化するサイバー攻撃に対しては机上学習だけでなく，実践形式のサイバー演習が有効であることから，様々な実践型のセキュリティ演習システムが提案，開発されている[6][7][8]．Web アプリケーションへの攻撃を対象とした実践型のセキュリティ演習システムもいくつか開発されている[9][10]．

表 1　NFV の構成要素

	概要	使用技術
VNF	Router,Firewall,WAF など	別研究にて開発中
NFVI	物理リソース 仮想化機構	ワークステーション Docker
NFV MANO	仮想化機構の オーケストレーション	OpenStack on Kubernetes

AppGoat[9]は，情報処理推進機構が提供している，Web 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の脆弱性について学習できる脆弱性体験学習ツールである．AppGoat では，ホスト OS 上に脆弱性を含んだ Web サーバを用意して，脆弱性の体験学習を実施する．そのため，安全上の理由から AppGoat を利用する場合には，PC をインターネットと切り離すよう要請される．また，ホスト OS 上に環境を構築するため，ホスト OS の設定や他のプロジェクトなどの環境を壊してしまう恐れがある．これに対して本システムでは，Docker コンテナ上に Web サーバを用意することで安全性と環境構築の容易さを確保している．

また AppGoat では，NoSQLi に関する演習は提供されていない．これに対して，本システムでは PHP を用いた対策演習だけでなく，mongoose などの MongoDB 向けのライブラリが充実している Node.js を用いた対策演習にも対応している．

Lei Li 氏らによる Developing Hands-on Labware for Emerging Database Security では，NoSQL データベースのセキュリティを学ぶためのハンズオンラボ RELAB の開発を実施している[10]．RELAB では，MongoDB を用いて NoSQL データベースの基本概念と基本操作，認証などのセキュリティ問題，ログ機能を活用した分析方法など MongoDB の利用方法について包括的に学ぶことが可能である．これに対して本システムは，MongoDB の扱いに着目するのではなく，サーバーサイド言語である PHP や Node.js を用いた NoSQLi 対策のためのセキュアプログラミングについてハンズオン演習を実施可能とする．さらに，本研究では攻撃者視点で Web アプリケーションを解析し，対象の Web アプリケーションで施されている攻撃への対策手法や内在する脆弱性を発見するスキルや，そのための知識を習得することが重要であると考えているため，NoSQLi 脆弱性の発見手法に関する演習を重点的にサポートしている．

3. 研究内容

本章では，本研究で開発したシステムについて述べる．

3.1　システム構成

本システムは Network Functions Virtualization（以下，NFV）の仕組みに準じて設計している．NFV とはルーターやファイアウォールなどのネットワーク機器を仮想マシンとしてソフトウェアベース実装する方式のことである[11]．NFV は主に 3 つの要素から構成されており，それぞれ Virtualized Network Function（以下，VNF），NFV Infrastructure（以下，NFVI），Management and Orchestration（以下，NFV MANO）となっている．それぞれの構成要素として検討している技術を表 1 に示す．

NFV の仕組みを活用することによりネットワーク技術者やセキュリティ技術者育成のためのさまざまな演習が実施可能なシステムの実現を進めている．将来的に，本システムをこの NFV 上に組み込むことで，WAF による対策演習

図 2　学習画面の一例

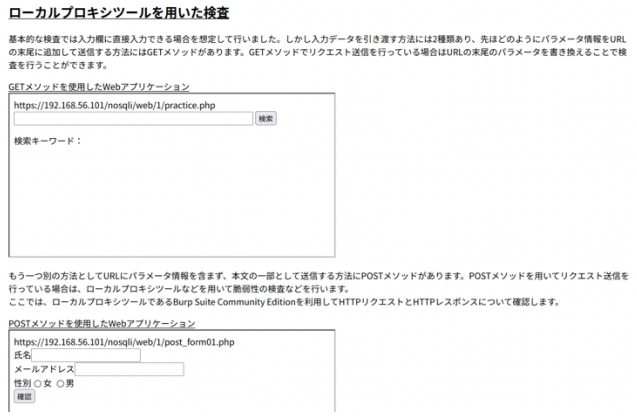

図 3　脆弱性発見演習用の学習コンテンツ

項目の追加を検討している．

本システムは，Web アプリケーション開発者を目指す初学者や，NoSQLi に関する知識が不足している Web アプリケーション開発者を本システムの利用者（以降，学習者）として想定する．

本システムの構成を図 1 に示す．本システムは，Docker 上に構築された演習環境にブラウザから接続することで演習を可能とする．演習用 Docker イメージの構築には Ubuntu18.04 のイメージを基に，演習に必要なソフトウェアを導入することで作成した．それぞれ Web サーバソフトウェアには Apache2.4.29，サーバーサイド言語には PHP7.2.24/Node.js v8.10.0，NoSQL データベースには MongoDB v3.6.3 を導入した．

NoSQLi に関する学習，演習を実施するため，本システムでは演習用 Web サーバに，NoSQLi の攻撃原理や対策手法について学ぶための学習用コンテンツと，これらの演習を実施するための脆弱性を持つ演習用 Web アプリケーションを開発，導入した．これらを使った学習手順については，次節で述べる．また，NoSQLi 脆弱性を発見する演習を実施するために，演習用 Web サーバとブラウザ間の通信を解析するためのローカルプロキシツールである Burp Suite Community Edition v2021.6.2 を導入した．ローカルプロキシツールとは，Web サーバとブラウザ間のデータのやり取りを閲覧および改ざんなどができるツールである．

学習者は，攻撃者視点で Burp Suite を用いた演習を実施することで，対象の Web アプリケーションで施されている攻撃への対策手法や内在する脆弱性を把握，発見するスキルやそのための知識を習得することが可能である．

3.2　演習内容と手順

学習者は，Web ブラウザを使用して，Docker 上に構築された演習用 Web サーバに配置されている学習用コンテン

ツにアクセスし，NoSQLi に関する学習，演習を進める．

学習者が学習に使用する Web ブラウザ上の画面の一例を図 2 に示す．これは，NoSQLi 対策演習用のコンテンツの例である．画面の左側は学習コンテンツ表示部であり，演習用 Web サーバ内にある学習用コンテンツが表示される．画面の右側は演習部であり，演習に用いる．この画面例の場合，演習部はさらに 2 つに分かれており，上部には脆弱性を持つ演習用 Web アプリケーション，下部には NoSQLi 対策を実施するためのコンソール画面が表示されている．学習者は，学習コンテンツに表示される解説や動画を参考にしながら，演習部に表示される脆弱性を持つ演習用 Web アプリケーションやコンソール画面，あるいは Burp Suite を操作することで，演習を進める． なお，簡単な演習内容の場合，学習用コンテンツ中に演習のための Web フォームが含まれる場合もある．

本システムを用いて学習を実施する場合，学習者はまず，NoSQLi 概要学習用のコンテンツを利用して NoSQLi の概要を学ぶ．実際に Web ブラウザ上に表示される Web フォームに，学習コンテンツに従って入力をしながら，NoSQLi の概要を学習する．NoSQLi の概要を学習するために，演算子のインジェクションや Server Side JavaScript Injection などを題材に学習コンテンツを作成した．

続いて，学習者は，NoSQLi 脆弱性発見演習用のコンテンツを利用して，NoSQLi 脆弱性発見手法の学習と演習を実施する．この演習では，NoSQLi 対策が不十分な Web アプリケーションを用いる．学習者は学習コンテンツに従って，Burp Suite を使用して，攻撃者ホストと演習用 Web サーバ間の HTTP リクエストと HTTP レスポンスを解析する．

Burp Suite を用いて HTTP 通信を解析し，脆弱性の発見手法を学ぶための学習用コンテンツの例を図 3 に示す．Burp Suite を用いて改ざんした HTTP リクエストを送信後のレスポンスの解析を繰り返すことで， Web アプリケーションの構造把握を進め，NoSQLi 脆弱性を発見する．これらのハンズオンによる演習を通して学習者は，Web アプリケーションに含まれる脆弱性の発見手法を習得する．

最後に，学習者は NoSQLi 対策演習用のコンテンツを利用して対策手法の学習と演習を実施する．まず，学習者は対策演習用コンテンツ内のテキストや動画を視聴して防御の理論や対策方法について学習する．学習者は Web ブラウザの右下に表示される演習用 Web サーバのコンソール画面（図 2）を通じて，脆弱な Web アプリケーションのソースコードを閲覧，修正する．具体的には，NoSQLi 対策として「受け取ったパラメータを String 型にキャスト」「ホワイトリスト型の入力値検証」「メタキャラクタのサニタイジング」などの処理を追加するためのセキュアプログラミングによるソースコードの修正を実施する．学習者は，ソースコードの修正後，再度 NoSQLi 攻撃を実施して，NoSQLi 脆弱性が解消されたことを確認する．

4. 実験

実験は情報系学部の大学生，大学院生を対象に実施する．実験対象者をインジェクション攻撃について，本システムで学ぶグループと座学で学ぶグループ 2 つに分割し，それぞれ学習の前後にインジェクション攻撃に関する事前・事後テストを実施する予定である．2 グループの事前・事後テストの点数の差から本システムが対策学習の支援ができているかを確認する予定である．

テスト内容は情報処理推進機構による情報処理安全確保支援士試験の参考書[12]と過去問を基に問題を作成する．事後テストでは事前テストと同レベルの別の問題を使用する．問題数はそれぞれ 10 問となっており，1 問 1 点として点数をつける．事前テストで解いた問題の解答は公開せずに，事後テストを実施する予定である．

5. 結論

本研究では，Web アプリケーションに対する NoSQLi 対策の実践的な演習を支援することを目的に，攻撃者視点を取り入れた NoSQLi の実践的演習システムを開発した．本システムを使用することで NoSQLi に対して安全な Web アプリケーションの作成方法を学習できる．

今後の予定として，本システムを NFV の仕組みに組み込むことで WAF を利用した NoSQLi 演習の追加を検討している．また，SQL データベースと NoSQL データベースを組み合わせて使用している環境が多いことから，実際に SQL と NoSQL が混在した環境で運用されているシステムを再現した演習の追加を検討している．

謝辞

本研究は JSPS 科研費 21K12185 の助成を受けたものである．

参考文献

[1] 総務省: 不正アクセス行為の発生状況(2020)，入手先 <https://www.soumu.go.jp/main_content/000671872.pdf>, (参照 2021-07-21)

[2] 岸本和理, 井口信和: Web アプリケーションセキュリティに関する実践的学習環境の開発，インターネットと運用技術シンポジウム論文集，Vol. 2020, pp. 107-108

[3] DZone: The Database Evolution，入手先 <https://www.cockroachlabs.com/guides/database-evolution-trend-report-2020/> (参照 2021-07-09)

[4] DB-Engines: DB-Engines Ranking，入手先< https://db-engines.com/en/ranking> (参照 2021-07-09)

[5] OWASP:OWASP Top10 -2017 入手先 <https://www.owasp.org/www-pdf-archive/OWASP_Top_10-2017(ja).pdf > (参照 2020-07-21)

[6] 八代哲, 高橋和氏, 渡辺亮平, 角田裕太, 田邉一寿, 齋藤祐太，齋藤孝道: 体験型サイバーセキュリティ学習システムの提案と構築，コンピュータセキュリティシンポジウム 2017 論文集，Vol.2017，No. 4, pp. 180-183(2017).

[7] 千川尚人, 小林康浩, 石原学, 白木厚司, 下馬場朋禄, 伊藤智義: サービス拒否攻撃演習システムの実装とそのアクティブラーニングシナリオによるセキュリティ技術教育，電子情報通信学会論文誌，Vol. J103-B, No. 4, pp. 180-183(2020).

[8] 湯川誠人,谷口義明,井口信和: 攻防戦型ネットワークセキュリティ学習支援システム，電子情報通信学会論文誌，Vol.J103-D, No.8,pp. 591-602 (2020)

[9] 独立行政法人 情報処理推進機構 IPA: 脆弱性体験学習ツール AppGoat，入手先 <https://www.ipa.go.jp/security/vuln/appgoat/index.html>, (参照 2020-07-21).

[10] Lei Li, Kai Qian, Qian Chen, Ragib Hasan, Guifeng Shao: Developing Hands-on Labware for Emerging Database Security，SIGITE'16 September 28-October 01,2016,Boston,MA,USA

[11] 富士通: ネットワーク仮想化(NFV)ソリューションを支えるプラットフォーム技術，FUJITSU. 66，6，p. 20-27(11,2015)

[12] 上原孝之，情報処理教科書 情報処理安全確保支援士 2020 年版．翔泳社，2019.