サイバー演習の防御演習時におけるシナリオ進行の自動化システムの提案

井上　拓哉 [1], Razvan Beuran[1]

Abstract

サイバー人材の育成だけでなく，多くの人にサイバー空間の脅威について知ってもらうためにもサイバー演習が必要となる。その中でも，脅威の認識および対処のため，サイバー攻撃に対する防御演習が重要である。しかし，既存の防御演習には教育に関する機能が欠けている。本稿では，防御演習によるセキュリティ教育を普及させるため，防御演習の進行と指導を自動化するためのシステム”DeTMan”を提案する。まずは既存の防御演習について考察し，教育システムとして用いるにあたり必要な機能について検討する。次に，検討された機能をいかにして実装するか検討し，概念実装を行う。最後に，実装した DeTMan について，教育システムとしての防御演習の観点から検証する。

Keywords: サイバーセキュリティ, サイバー演習, 防御演習, サイバーレンジ

1. はじめに

現在，技術の発展に伴い，企業にとっても個人にとってもセキュリティリスクは深刻なものとなっている。しかし，IT 技術を適切に扱うための教育が発展に追いついていない。セキュリティ人材の不足 [1] や，社会のセキュリティに関する意識 [2] が大きな問題となっている。そのため，本稿では，セキュリティ人材の育成に大きな役割を果たしている防御演習に注目した。防御演習では，受講者は与えられた環境に対し実行される攻撃に対処することで，サイバー攻撃の脅威と対策について学ぶ。

無料で参加可能な防御演習として，Hardening[3] と Micro Hardening[4] がある。防御演習の開催には，セキュリティに関する高度な専門知識が必要となる。そのため，現在の主流な防御演習は，特定の人物や団体によって運営されるに留まっている。結果として，防御演習の機会は限られてしまい，需要を満たせていない。防御演習を普及には，より簡単に開催できることが重要である。また，現在の防御演習は演習による訓練に重きが置かれており，演習の参加者に対する指導なども提供される。

本稿では，Hardening と Micro Hardening を参考として防御演習による教育システムについて考察する。そして，教育としての防御演習を提供するサイバー防御演習進行管理システム”DeTMan”を提案する。

[1]北陸先端科学技術大学院大学

2. 既存の防御演習

教育としての防御演習を提供するにあたり，既存の防御演習である Hardening と Micro Hardening を例に考察する。

2.1. *Hardening* の紹介

Hardening とは，Web Application Security Forum(WASForum) が主催するセキュリティ堅牢化の競技大会である。Hardening では，脆弱性を持つ EC サイトの運営して，チーム対抗で売り上げを競う。Hardening における売り上げとは，クローラーによる EC サイトでの自動購入によって成立する。運営側からの攻撃に対して，参加者はシステムを堅牢化することでサービスを維持し，売り上げの最大化を目指す。

防御演習の開催には，図 1 に示す 3 つのステップが必要となる。想定する攻撃者の行動や攻撃パターンなどを策定する (1) 防御演習シナリオの作成，シナリオに沿った防御演習の環境を構築する (2) 防御演習環境の構築，参加者に攻撃を与える (3) 防御演習の実施である。

防御演習シナリオの作成は，防御演習において最も重要な部分である。どのような演習を行うのか，どのような攻撃を行うのか (どのような脆弱性を埋め込むのか)，どのように進行するのかについて決定する。Hardening では様々な分野の専門家が集まり，演習シナリオを作成する。

防御演習環境の構築では，作成した防御演習シナリオに基づいて演習環境を作成する。防御演習では実際に攻撃するため，仮想環境で演習を実施する必

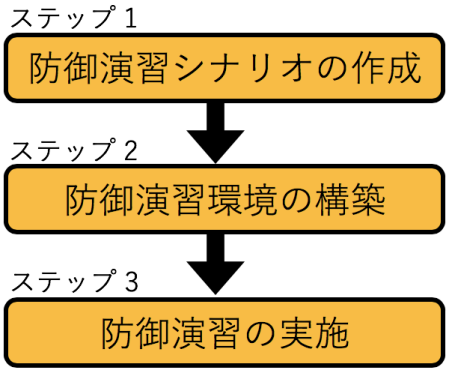

図 1: 防御演習の 3 ステップ

要がある。Hardening における演習環境は，Alfons[5] と呼ばれる環境構築システムを用いて演習環境を構築している。

最後に，実際に演習を実施する。Hardening では，運営側の攻撃はすべて手動で実行される。運営側は会場に設置されたカメラで参加者の様子を確認しながら攻撃するため，チームに合わせて攻撃を調整することができる。そのため，順調に進んでいるチームにはより高度な攻撃を，低調なチームには攻撃をしないといった，柔軟な進行が可能である。

また，Hardening では実環境を想定した仮想ネットワークを使用するだけでなく，顧客対応や上役への報告，さらにはマーケットプレイスと呼ばれる企業のサービス導入なども競技の中で行われる。技術だけなく，サイバーセキュリティに携わる上で必要になると考えられる様々な知識やスキルを学ぶことが可能である。

2.2. *Micro Hardening*

Micro Hardening は株式会社川口設計の川口洋氏によって提供される競技形式の勉強会である。Hardening Project のサブプロジェクトとして誕生した。Hardening と比較して，カジュアルな演習になっている。競技時間は 1 セット 45 分であり，1 度の演習で，3 セット以上同じ内容を繰り返す。簡素化のために，顧客対応や上役への報告といった，技術的な対策以外の要素は省かれている。

Micro Hardening は防御演習環境の構築から攻撃の実行まで，全て自動化されている。そのため，川口氏 1 人だけでも運営が可能であり，Micro Hardening は日本各地で頻繁に開催されている。

Micro Hardening では，攻撃の実行は時間によって動作するタイムドリブン方式により自動化されている。そのため，すべてのセットにおいて同じタイミングで同じ攻撃が実行される。参加者は，攻撃について調査し，次のセットで対策を施すといった試行錯誤を，1 度の演習の中で繰り返すことができる。

3. 既存の防御演習が持つ教育上の課題

3.1. 開催の困難さ

シナリオの作成は演習において，最も重要なステップである。シナリオは，演習の目的に応じて検討する必要がある。そのため，専門家の知見を活用して作成する。また，作成したシナリオは，演習の目的が同じであれば再利用が可能である。

演習環境の構築には，CyRIS[6] などのサイバーレンジ構築ツールや，Ansible などの構成管理ツールを用いることを推奨する。環境の構築が簡単になるだけでなく，同じ演習を開催する場合に演習環境を簡単に構築できるためである。上記のツール群は，演環境の構築に設定ファイルを用いる。設定ファイルの再利用により，何度でも同じ演習環境を作成することができる。

演習において実行される攻撃は多種多様であるため，様々な分野の技術者が必要となる。また，演習の参加者は，運営側の人数よりも多い。そのため，手動で攻撃する場合には運営側に重い負担がかかる。教育の一環として防御演習を普及させるためには，特別な人材が必要であることや運営側に重い負担を強いることは問題である。

シナリオと演習環境の構築は再利用が可能である。しかし，演習の実施は再利用ができない。そのため，防御演習において演習の実施が負担となっている。

Micro Hardening は，演習の進行を自動化することによりたった 1 人でも開催可能である。そのため，防御演習を教育として提供する場合には自動化による負担軽減が必要である。

3.2. 受講者に応じた演習の進行

Micro Hardening はタイムドリブン方式により自動化されている。しかし，タイムドリブン方式では受講者全員に同じ内容の演習を提供することになる。そのため，演習が基準としているレベルから離れている人は対象外となってしまう。

教育としての防御演習では，Hardening のように受講者それぞれの状況に合わせて演習を自動で進行させる必要がある。

3.3. 受講者に対する指導の不足

DoS 攻撃のような例外を除き，サイバー攻撃とはコンピュータの所有者に気付かれないように実行される。Hardening や Micro Hardening は，実際の環境に近い形で行うため，攻撃されたことに気付くこともまた，演習の一部である。そのため，攻撃に気付

くことなく演習が終了する事態も十分に想定される。これは，教育としては問題である。

文部科学省の高等学校学習要項 [7] において，「基礎的・基本的な知識及び技能を確実に習得させ，これらを活用して課題を解決するために必要な思考力，判断力，表現力その他の能力をはぐくむとともに，主体的に学習に取り組む態度を養い，個性を生かす教育の充実に努めなければならない」と記載されている。また，「個々の生徒の特性等の的確な把握に努め，その伸長を図ること」と記載されている。つまり，教育として防御演習を行うには，受講者の進捗に応じて，セキュリティについて確実に習得させることが必要である。

Hardening や Micro Hardening では演習後の解説により，どのような攻撃されたのかについては知ることができる。しかし，演習後の解説ではいつ・どのように攻撃されたのかはわからない。そのため，攻撃された際にコンピュータはどのような反応を示すのか，知ることができない。教育としては，演習中に指導する必要がある。

また，攻撃について教えるだけでは不十分である。例えば，受講者がどのログファイルを確認するべきなのか知らなければそれ以上の情報について調査できない。加えて，ログ保存の設定が適切でなければ，そもそも確認するための情報が存在しない。

防御演習における指導では，演習中に，どのような攻撃かだけでなく，検知方法や対策まで指導する必要がある。

4. 教育としての防御演習のための自動化システム

防御演習を教育として普及させる場合には，運営の負担を軽くするために進行の自動化が重要である。しかし，既存の自動化された防御演習である Micro Hardening は，Hardening にあった柔軟さが失われている。また，既存の防御演習は教育的な指導が不足している。そのため，教育としての防御演習には以下の点が重要になる。

- 受講者の状況に応じた進行をどのように自動化するか
- どのように指導するか
- どのような振り返りを提供するか

4.1. 受講者の状況に応じた進行

受講者の進捗に合わせるためには，受講者が攻撃に対処するまで待機することと，受講者の状況に応じて異なる攻撃を実行することが必要になる。攻撃を待機させるシステムは，以下の 4 つの機能により実装できると考えられる。

- 進行の独立
- 進行の分岐
- 死活監視機能
- イベントドリブン方式による進行機能

4.1.1. 進行の独立

受講者の状況に応じて進行させるためには，演習の進行を受講者ごとに独立させなければならない。進行を独立させることにより，他の受講者による影響を受けることなく自分でペースで進行可能になる。

4.1.2. 進行の分岐

受講者の状況は，チームごとに異なる。受講者に応じた進行には，順調な受講者にはより高度な攻撃を，低調な受講者には簡単な攻撃や防御に失敗した攻撃を繰り返すといった進行が必要である。そのため，演習の進行を分岐させる必要がある。

4.1.3. 死活監視機能

受講者が管理するネットワークにおいてサービスが停止している場合は，何らかのアクシデントが発生していると考えられる。そのため，サービスが停止している場合は攻撃をするべきではない。

4.1.4. イベントドリブン方式による進行機能

死活監視機能だけでは，受講者が緊急の処置としてサービスを再起動をしただけの場合でも攻撃を再開する。攻撃に対する調査する時間を確保するためにも，死活監視機能以外にも攻撃を待機する機能が必要になる。

イベントドリブン方式を用いることにより，特定のイベントが発生するまで，進行を待機させることが可能である。

4.2. 指導方法

演習によるセキュリティ教育として以下の 3 点が必要だと考えられる。

- 段階的な攻撃の通知
- 受講者の理解を確認
- 演習の振り返り

4.2.1. 段階的な攻撃の通知

本稿の目的は，演習による教育である。1 度に攻撃に関するすべての情報を通知しては受講者自らが考える機会が失われてしまう。そのため，異常の発生・調査すべきファイル・実行された攻撃・対策方法と段階的に受講者に通知する機能が必要である。

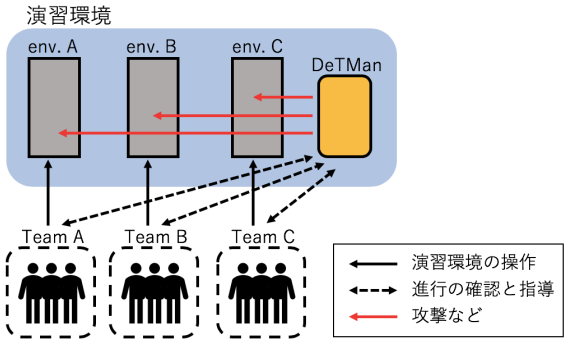

図 2: DeTMan の概要

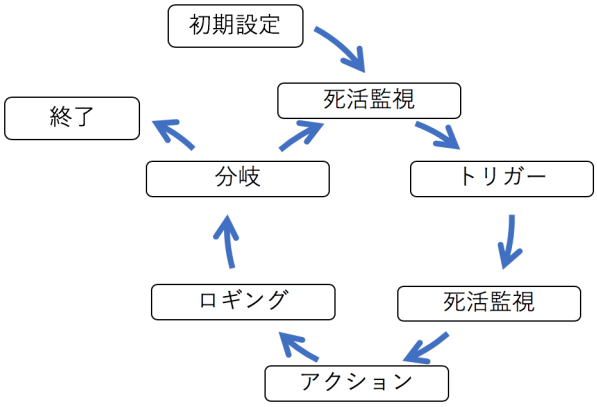

図 3: DeTMan の動作

4.2.2. 受講者の理解を確認

システムに言われるがままに，何も理解せず演習を進めるような事態は避けなければならない。受講者の理解状況を確認する機能が必要である。

4.2.3. 演習の振り返り

攻撃について調査するには以下の情報が必要になる。

- いつ攻撃したのか
- どのような攻撃をしたのか
- 結果はどうだったのか

攻撃のタイミングに関する情報により，各種のログを調査する際に，調査する範囲を限定することが可能である。実行された攻撃の種類による情報により，何を調査するべきか特定可能である。攻撃の成否に関する情報により，受講者の施した対策の効果を知ることが可能である。

また，攻撃に関して任意に調査可能にするため，受講者が任意のタイミングでこれらの情報を確認できる必要がある。

5. 提案システム DeTMan

4 章では，防御演習による教育システムが持つべき機能が明らかとなった。本章では，防御演習自動進行管理システム DeTMan(Defense Training progress Management system) の概念実装を行う。

5.1. *DeTMan* の概要

DeTMan の概要を図 2 に示す。DeTMan は，攻撃と指導をすべて自動で実行する。そのため，演習の実施において，運営側に人を必要としない。

DeTMan は演習の進行を独立させるために，受講者数 (用意された演習環境数) と同数の子プロセスを生成する。演習の進行は子プロセスが担う。1 つの環境について，1 つの子プロセスを割り当てることにより，進行の独立に成功した。

DeTMan の動作を図 3 に示す。DeTMan の各動作において，初期設定以外は子プロセスが担当する。以降の項において，DeTMan の各動作について説明する。

5.2. *DeTMan* の動作

5.2.1. 初期設定

DeTMan ではチームファイルとシナリオファイルという 2 つの設定ファイルを使用する。チームファイルには受講者名 (チーム名) と攻撃対象となるサーバについて記述する。シナリオファイルには演習の具体的な流れについて記述する。シナリオファイルのサンプルを図 4 に示す。

DeTMan において，シナリオファイル内の 1 つのまとまりをステップと呼ぶ。つまり，シナリオファイルとはステップの集合体である。

DeTMan はアクションの成否に応じて動作を変化させることが可能である。シナリオファイルの success と failure には，アクションが成功または失敗した場合について記述されている。また，DeTMan には防御演習に競技要素の持ち込みを可能とするため，ポイント機能がある。success と failure には，next に次のステップを，point にアクション終了後に加減算されるポイントを記述する。

初期設定では，まず，2 つの設定ファイルを読み込む。シナリオファイルでは，trigger，success，failure は省略可能である。ファイルの読み込み後，省略された部分を補完し，DeTMan 用に再構成する。次に，データベースに関する設定を行う。防御演習の進行

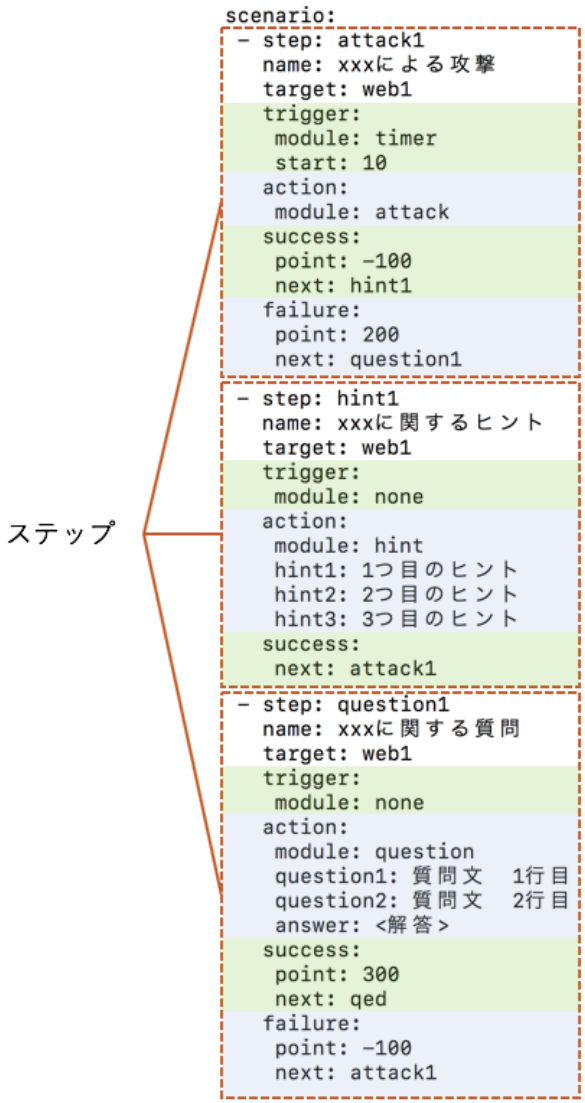

図 4: シナリオファイルのサンプル

状況を，外部から参照可能とするためにデータベースを用いる。データベースには，以下の 4 つのテーブルを作成する。

- 演習環境に関する情報を保管する state テーブル
- 現在実行中のステップとポイントの情報を保管する progress テーブル
- アクションの実行結果に関する情報を保管する log テーブル
- シナリオに関する情報を保管する scenario テーブル
- 受講者への指導に関する情報を保管する board テーブル

最後に，子プロセスを生成する。以降は，子プロセスの終了を待機する。

5.2.2. 死活監視

DeTMan は，通常は ping を用いたネットワークの疎通確認により死活監視を行う。しかし，ネットワークの疎通情報のみでは，攻撃を待機するかの判断には不十分である。そのため，チームファイルに記された演習環境において，サーバ名に web という単語が含まれていた場合に，HTTP リクエストによる HTTP サーバの動作確認を行う。今後，動作確認を行うサービスを追加する。死活監視の結果を state テーブルに書き込む。疎通確認または動作確認が失敗した場合，死活監視が成功するまで攻撃を待機する。

死活監視は，トリガー前と，アクション前に実行する。これにより，トリガーとアクションの際には，演習環境が正常に動作していることが保証される。

5.2.3. トリガー

DeTMan では，イベントドリブン方式におけるイベントをトリガーと呼ぶ。現ステップの trigger に記されたモジュールが実行され，トリガーが発生するまで動作を待機する。

DeTMan は，攻撃を待機するための機能としてイベントドリブン方式を採用した。しかし，非同期 I/O を採用すると同時に複数の攻撃に対処しなければならない事態が想定される。そのため，1 度に 1 つのトリガーのみを待機する。フィッシングやリバースシェルのようなマルウェアを用いる演習は困難になるが，DeTMan では対応しない。

5.2.4. アクション

アクションにおいて，DeTMan は受講者に対して能動的に動作する。現ステップの action に記されたモジュールが実行される。主なアクションとして，攻

撃の実行を想定している。他にも，メールの送信なども想定している。

アクションは，実行の結果とコメントを戻り値として返す。実行の結果は，アクションの成否であり，success または failure である。しかし，例えば，アクションとして実行された攻撃の成功と，メール送信の成功は意味が反対である。攻撃が成功した場合は，受講者に防御をさせるために，次の攻撃には進まない。メール送信が成功した場合は，次の攻撃に進む。次節で説明するロギングされたデータを受講者が見た場合に混乱する。そのため，コメントとして受講者から見た際の結果について返すことにより，混乱を防ぐ。

また，演習中の指導もアクションとして行う。DeTMan はアクションの成否により進行を分岐する。そのため，攻撃が成功した場合にのみに実行するアクションとして，指導が可能である。

DeTMan では，指導のためのアクションとして hint と question を用意した。hint では，board テーブルに情報を格納する。図 4 を参考に説明する。本シナリオでは，hint1，hint2，hint3 の 3 つのヒントが記述されている。DeTMan は，ステップ hint1 が 1 度目に実行された場合，hint1 を格納する。2 度目に実行された場合には hint2 を，3 度目以降は hint3 を格納する。これにより，段階的に受講者に対して情報が提示することが可能である。

question も同様に，board テーブルに情報を格納する。本シナリオでは，question1，question2 の 2 つの質問が記述されている。hint とは異なり，すべての情報が同時に格納される。board テーブルには，hint や question などを区別可能な情報も格納されるため，区別可能である。

5.2.5. ロギング

DeTMan は，実行したアクションに関する情報を log テーブルに格納する。ポイントはロギングの際に計算される。格納される情報は以下の 4 つである。

- 現在の時間
- チーム名
- 実行されたステップの step
- アクションのコメント
- アクション実行後のポイント

また，演習終了後には log テーブルのデータを csv ファイル形式により出力することができる。

表 1: 防御演習の比較

	Hardening	Micro Hardening	DeTMan
演習の用途	訓練	訓練	教育
進行の柔軟さ	◎	×	○
開催難易度	×	◎	○
リアリティ	◎	○	×

5.2.6. 分岐

アクションの成否に応じて，実行ステップを変更する。実行ステップが qed であった場合は演習を終了し，そうでなければ死活監視を行う。そして，実行ステップが qed となるまで，繰り返す。

5.3. WEB UI

データベース内のデータを可視化するために WEB UI を用いる。これにより，受講者の状況を確認可能である。

また，board テーブルのデータも可視化するため，指導にも WEB UI を用いる。データを可視化するだけでなく，未解答の question があった場合は解答フォームも作成する。この解答フォームを用いることにより question に対して解答が可能である。

5.4. 既存の防御演習との違い

表 1 に，Hardening，Micro Hardening，DeTMan の違いをまとめる。既存の防御演習と DeTMan の最大の違いは，演習の用途である。既存の防御演習はスキルアップを目的として開催されるが，DeTMan は教育を目的とする。そのため，DeTMan は既存の防御演習にはなかった教育用の機能を複数持つ。

DeTMan は簡単に柔軟な演習を実施可能だが，演習中に指導を行うためリアリティは損なわれている。実際のインシデントは，DeTMan のように攻撃について教えてくれることはない。そのため，DeTMan は初心者を対象とする。

6. 自動化された演習進行および教育の実証実験

6.1. 実証実験の概要

図 4 に示すシナリオファイルを用いて DeTMan の実証実験を行う。シナリオのフローを，図 5 に示す。本シナリオでは攻撃 attack1 を実行し，防御に成功した場合，attack1 に関して受講者に質問する。攻撃 attack1 の防御に失敗した場合はヒントを表示し，受講者に質問する。受講者が正しく解答した場合は演習を終了し，誤答した場合はもう 1 度攻撃を実行する。

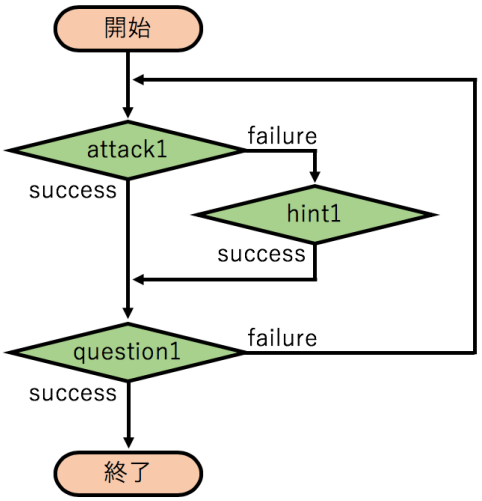

図 5: 演習フロー

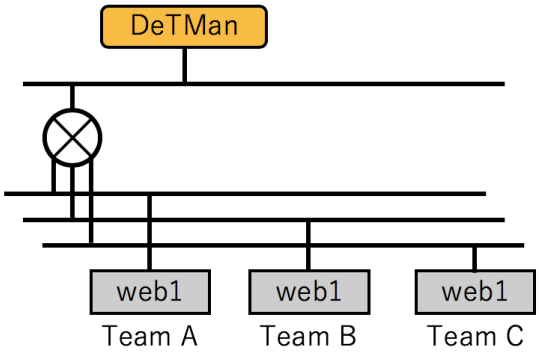

図 6: 演習環境

また，本実験は，図 6 に示す演習環境で実施した。DeTMan は 3 チームが管理する合計 3 台のサーバに対して攻撃する。

Team A は上級者，Team B は中級者，Team C は初心者とし，表 2 のように進行すると定義する。質問には，防御に成功した場合に正答する。例として，Team B は 2 度目に防御に成功するため，質問に 1 度目は誤答し，2 度目に正答する。また，Team B は 2 度目の attack1 におけるトリガー待機中に HTTP サーバが停止し，Team C の演習終了後に再起動する。

6.2. 受講者の状況に応じた進行

演習中のある瞬間において，WEB UI により可視化した progress テーブルの一部を図 7 に示す。progress テーブルより，チームごとに異なる進行をしていることが分かる。

表 2: 各チームにおける演習の進行

チーム	進行
Team A	1 度目に防御成功
Team B	2 度目に防御成功
Team C	5 度目に防御成功

Team	Step
Team A	Complete
Team B	attack1
Team C	question1

図 7: progress テーブルの抜粋

図 8，図 9 に，log テーブルから Team A，Team B に関して抜粋したものを示す。

Team A に関する log テーブル内の情報より，Team A は 1 度目から攻撃 attack1 の防御に成功したため，hint は実行されていないことが分かる。また，question1 にも 1 度で正答したため，そのまま演習が終了した。Team B に関する log テーブル内の情報より，Team B は 1 度目の攻撃 attack1 の防御に失敗したため，次に hint1 が実行されていることが分かる。また，1 度目の question1 に誤答したため，attack1 が再度実行されている。2 度目の attack1 は防御に成功したため，hint1 は 1 度しか実行されていない。DeTMan は，受講者の進捗に応じて進行を変化させてることが確認できた。

また，Team B に関する log テーブル内の情報より，2 度目の attack1 はシナリオ通りならば question1 の 10 秒後に実行されるはずであるが実行されていないことが分かる。しかし，ログには Unavailable と記され，attack1 はシナリオ通りに実行されていない。

この時の可視化された state テーブルから Team B に関して抜粋したものを図 10 に示す。Team B では question1 の後に HTTP サーバが停止したため，DeTMan は HTTP サーバが再起動されるまで攻撃を待機した。

Time	Team	Step	Comment	Point
2018/11/16 07:52:36	Team A	sys	START	1000
2018/11/16 07:52:46	Team A	attack1	Defense success	1200
2018/11/16 07:52:56	Team A	question1	Success	1500
2018/11/16 07:52:56	Team A	sys	COMPLETE	1500

図 8: log テーブルの Team A に関する抜粋

Time	Team	Step	Comment	Point
2018/11/16 07:52:36	Team B	sys	START	1000
2018/11/16 07:52:46	Team B	attack1	Defense failed	900
2018/11/16 07:52:46	Team B	hint1	Success	900
2018/11/16 07:53:06	Team B	question1	Wrong...	800
2018/11/16 07:53:16	Team B	sys	Unavailable	0
2018/11/16 07:54:57	Team B	attack1	Defense success	1000
2018/11/16 07:55:17	Team B	question1	Success	1300
2018/11/16 07:55:17	Team B	sys	COMPLETE	1300

図 9: log テーブルの Team B に関する抜粋

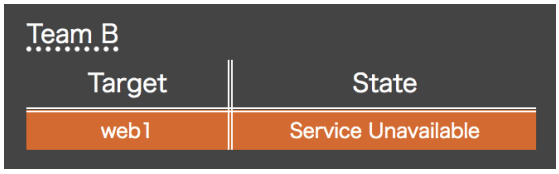

図 10: state テーブルの Team B に関する抜粋

以上の点より，DeTMan は，受講者の状況に応じて演習を進行させていることが確認できた。

6.3. 受講者に対する指導

図 11 に，WEB UI により可視化した board テーブルから Team C に関して抜粋したものの一部を示す。また，図 12 に WEB UI の解答フォームを示す。WEB UI では，対話的であることを強調するため，SNS ライクに表示する。緑のコメントがヒント，赤のコメントが質問，橙色のコメントは受講者の解答である。図 11 において，hint1 が何度目の実行かにより，表示されるヒントが変化していることが確認できる。

DeTMan では，対話的に受講者に対して指導可能であることが確認できた。

6.4. 演習の振り返り

図 7 や図 10 は演習中に撮影したものである。つまり，演習中に受講者は WEB UI により，現在実行されているステップや演習環境の稼働状況について知ることができる。過去に実行された攻撃についても，図 8 や図 9 のように参照可能である。受講者は，WEB UI に表示される情報を参考に演習環境の調査を行う。

また，これらの情報はデータベースに保管されているため，演習後にも残る。DeTMan は log テーブルの内容をチーム別にレポートとして出力する機能を持つため，演習後も振り返りが可能である。

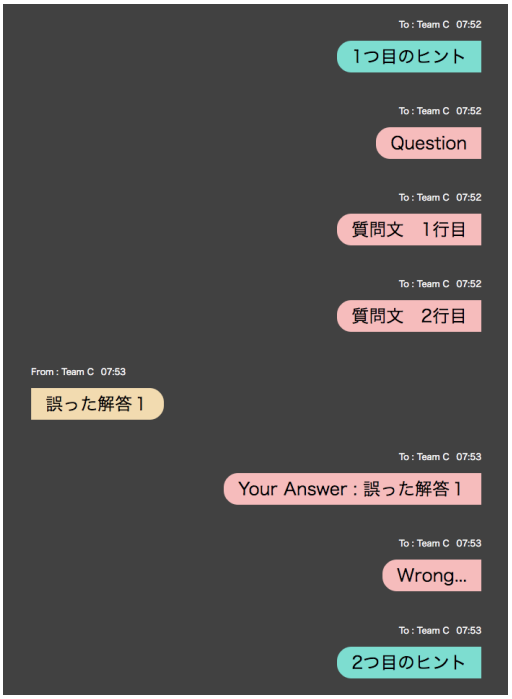

図 11: board テーブルの Team C に関する抜粋

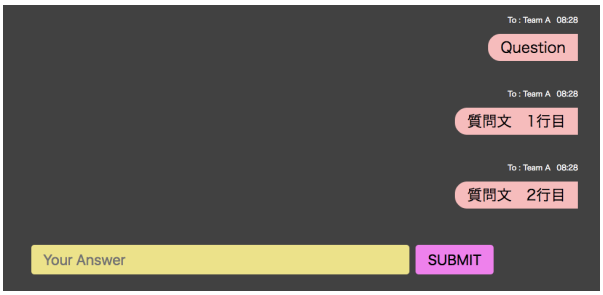

図 12: board テーブルの解答フォーム

7. CyRIS との連携

我々のプロジェクトではサイバー演習統合フレームワーク CyTRON[8] を開発している。その中にサイバーレンジを作成するツール，CyRIS が含まれている。CyRIS において，受講者それぞれに割り当てられる演習環境をインスタンスと呼ぶ。CyRIS は，KVM を用いて作成された基本となるインスタンスを必要なだけ複製し，サイバーレンジを作成する。図 13 に CyRIS と DeTMan の連携について示す。

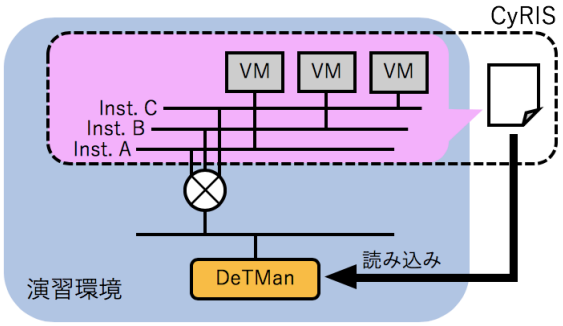

図 13: CyRIS との連携

CyRIS は，作成した仮想マシンに一定のルールに基づいて IP アドレスを割り振り，サイバーレンジの環境情報が記述された yaml ファイルを出力する。このファイルには，DeTMan のチームファイルに必要となる，攻撃対象の IP アドレスなどの情報が格納されており，DeTMan はチームファイルの代わりにこの yaml ファイルを利用可能である。本機能により，CyRIS を用いて演習環境を作成した場合は，チームファイルを作成する手間を省略できる。

8. 今後の課題

8.1. 攻撃側の *IP* アドレスが固定

現在の DeTMan には，攻撃側の IP アドレスを変更する機能がないため，受講者のファイアウォールによる対策により，攻撃がすべて防がれてしまう。 そのため，攻撃者の IP アドレスをランダムで変更する機能が必要である。

8.2. ステップ間の連携

現在の DeTMan は 1 つ 1 つステップが独立してため，あるステップにより情報を奪取しても，別のステップではその情報を活かすことができない。そのため，ステップ同士を連携させるために，攻撃によって得られた情報を保管するための手段が必要になる。

8.3. WEB UI のアクセス制限

現在の WEB UI では，WEB UI にアクセスしたすべての人物が，すべての情報にアクセス可能である。そのため，進行に関する情報が他の受講者に公開されるだけでなく，Board ページで別の受講者に対する質問に解答することも可能である。何らかの形でアクセスを制限することにより，他の受講者に関する情報を閲覧できないようにする必要がある。

9. さいごに

本稿では，既存の防御演習である Hardenig と Micro Hardening を参考に，教育としての防御演習について考察した。考察を元に，教育としての防御演習に必要な機能をどのように実装するか検討した。そして，自動で防御演習の進行と教育をするためのシステム”DeTMan”を実装した。DeTMan が，教育としての防御演習に必要な機能を満たしているか検証した。今後は，DeTMan の完成度を高め，実際に演習を実施することにより検証を継続する。

謝辞

本研究は JSPS 科研費 17K00478 の助成を受けたものです。

参考文献

[1] 経済産業省. IT 人材の最新動向と将来推計に関する調査結果. <http://www.meti.go.jp/press/2016/06/20160610002/20160610002-7.pdf>.

[2] 諏訪博彦, 原賢, 関良明. 情報セキュリティ行動モデルの構築—人はなぜセキュリティ行動をしないのか. 情報処理学会論文誌, Vol. 53, No. 9, pp. 2204–2212, September 2012.

[3] Hardening project. <http://wasforum.jp/hardening-project/>.

[4] 川口 洋. <https://microhardening.connpass.com>.

[5] 安田真悟. Alfons: A mimetic network environment construction system. In *11th EAI International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, Jun 2016.

[6] Razvan Beuran, Pham Cuong, Tang Thanh Dat, Ken ichi Chinen, Yasuo Tan, and Yoichi Shinoda. Cybersecurity education and training support system: Cyris. *IEICE Transactions on Information and Systems*, Vol. E101-D, No. 3, pp. 740–749, March 2018.

[7] 文部科学省. 高等学校学習指導要領. <http://www.mext.go.jp/component/a_menu/education/micro_detail/__icsFiles/afieldfile/2011/03/30/1304427_002.pdf>.

[8] Razvan Beuran, Tang Thanh Dat, Pham Cuong, Ken ichi Chinen, Yasuo Tan, and Yoichi Shinoda. Integrated framework for hands-on cybersecurity training: Cytrone. In *Elsevier Computers & Security*, Vol. 78C, pp. 43–59, June 2018.