修士論文

学習用サイバー防御演習の進行管理自動化に関する研究

1710021　井上　拓哉

主指導教員　Razvan Beuran 特任准教授
審査委員主査　Razvan Beuran 特任准教授
審査委員　篠田 陽一 教授
丹 康雄 教授
知念 賢一 特任准教授

北陸先端科学技術大学院大学
先端科学技術研究科
(情報科学)

平成 31 年 2 月

概 要

情報通信技術は急速な発展を遂げ，社会のあらゆるものがネットワークに依存するようになった。IoT による第 4 次産業革命が盛んに取り上げられ，インターネットへの依存はさらに加速していく。しかし，情報通信技術が身近なものになるほど，サイバー攻撃によるリスクは身近かつ深刻なものとなる。サイバー空間は非常に便利であるが，同時に非常に危険でもある。悪意ある攻撃者は常にインターネットを探索し，脆弱なコンピュータを探索している。他にも，迷惑メールやフィッシングサイト，強制リダイレクトされる WEB サイトなど，悪意あるサイバー空間上の罠は日々増加している。しかし，これらの攻撃は適切な対策を行うことにより，そのほとんどを防ぐことができる。また，サイバー攻撃による被害にあっても，慌てずに適切な対応を行えば，被害を抑えることが可能である。そのため，セキュリティ教育が重要となる。

しかし，現状では技術の急速な発展に教育が追いついていない。そのため，セキュリティ人材の不足だけでなく，社会におけるセキュリティリテラシーも十分ではない。そこで，本研究ではサイバー防御演習に注目した。サイバー防御演習では，受講者に与えられた演習環境に対して運営側がサイバー攻撃を行う。受講者は演習環境に対してセキュリティ対策を行うことでサイバー攻撃を防ぐとともに，インシデント発生時にはインシデントハンドリングを行う。サイバー防御演習により，サイバー攻撃の脅威を体験することで対策の重要性について学ぶと同時に，実際に被害にあってしまった際の予行演習とすることで，現実でも慌てずに対処できるようになることが期待される。しかし，実際にサイバー攻撃を受けるような技術的な演習は，費用や労力の問題により開催が困難であり，限られた人しか受講することができない。また，既存の技術的なサイバー防御演習には学習要素が薄く，演習の目的がサイバー攻撃の体験になるか，対策および対処の学習になるかは，個人の能力に依存する。そのため，学習用のサイバー防御演習を簡単に実施可能とすることが重要である。

本研究は，学習用サイバー防御演習の進行管理を自動化することにより，誰でもサイバー防御演習を実施可能とすることを目的とする。また，演習の進行は，手動によってサイバー攻撃が実行されるサイバー防御演習のように，受講者の能力と演習状況に合わせた柔軟な進行を自動化により再現することを目的とする。自動化により，誰でもサイバー防御演習が実施可能となるだけでなく，いつ・どのような攻撃を行ったのが明確となるため，振り返りも容易となる。柔軟な演習の進行を実現するためには，受講者の能力を判断する機能と，演習の進行を待機する機能が必要になると考えた。一般に，サイバー防御演習の運営にとって受講者の能力は未知である。そこで，演習中に行なったサイバー攻撃の成否により受講者の能力を判断する。また，手動によりサイバー攻撃を行う演習では，演習中の受

講者の様子や演習環境の状況を伺いながら演習を進行させる。そのため，柔軟な進行を実現は，実行したサイバー攻撃の成否に応じて演習の進行を分岐させる機能と受講者のサービスの稼働状況に応じて演習の進行を待機させる機能，受講者が特定の動作を行うまで演習の進行待機させる機能により実現可能である。

本研究では，これらの機能を持つサイバー防御演習進行管理システム DeTMan を提案し，その設計と実装を行なった。DeTMan は用意された演習シナリオに従い，演習を進行させる。演習シナリオには，実行する攻撃と実行するタイミング，攻撃の成否に応じた進行先が記述されている。また，攻撃の実行前などにサービスの監視を行い，異常が検知された場合は受講者に何らかのアクシデントが発生していると判断して演習の進行を待機する。他にも，攻撃の代わりに受講者に対して WEB UI を通じてメッセージを表示したり，クイズを出題したりといった学習支援も可能である。DeTMan を用いて，実際に学習目的のサイバー攻撃を行うことで DeTMan の有用性について実験を行った。実験により，様々なサイバー攻撃を再現可能であり，同時に，既存の防御演習と比較して少ない労力と費用でサイバー防御演習が開催可能であることがわかった。また，既存の防御演習とは違い，演習中に受講者に対して学習要素を提供可能であることも確認できた。DeTMan を用いることにより，サイバー攻撃について口頭で説明を受けるだけでなく，実際に体験することが可能となる。サイバー攻撃とは特別なものではなく，その脅威がいつ襲いかかってきてもおかしくないことを学ぶことができ，セキュリティ教育のさらなる充実に寄与する。

目 次

図 目 次

表 目 次

第1章　はじめに

本章では、研究の背景、目的、論文の構成について述べる。

1.1　背景

現在，情報通信技術は急速な発展を遂げている。個人の利用するサービスだけでなく，産業や行政においても情報通信技術の導入が進み，様々なもののデジタル化が進行中だ。また，IoT による第 4 次産業革命が始まり，社会のあらゆるものがネットワークと接続する時代が訪れようとしている。しかし，情報通信技術が身近なものになるほど，サイバー攻撃によるリスクは深刻なものとなる。近年の攻撃は，従来の個人が愉快犯的に行って攻撃とは異なり，金銭をはじめとする様々な目的のために組織的に行うものが増えてきている。昨今では，サイバー攻撃による機密情報の流出が大きな問題となっている。流出した情報は，スパムメールや不正アクセスだけでなく，個人情報の売買などに使用される。さらに，近年では仮想通貨の普及により，仮想通貨の匿名性を悪用した，不正送金や不正マイニング，脅迫による金銭の要求といった，より金銭に直結した攻撃が横行している。また，ショッピングサイトを代表とするサービスは，インターネットを通してサービスを提供することにより収益を得ている。そのため，DoS 攻撃などによりサービスが停止した場合には，経済的な損失が発生する。サイバー攻撃に対して脆弱であることは，自分だけの問題ではない。DDoS 攻撃に代表されるように攻撃者の踏み台になり，自らが加害者となってしまうことも頻繁に発生している。これらの被害は，標的型攻撃のような高度な攻撃によるものだけでない。多くの被害が単純なパスワード [1] や設定の不備，アップデートのしていないといった，初歩的なミスによって引き起こされている。

サイバー攻撃による被害の防止や軽減には，情報通信技術を適切に扱うための教育が重要となる。しかし，現状では情報通信技術の急速な発展に教育が追いついていない。結果として，セキュリティ人材の不足 [2] や，社会のセキュリティリテラシー不足 [3] が大きな問題となっている。そこで，サイバーセキュリティの啓蒙や学習支援のために大きな効果が期待できるサイバー防御演習に注目した。サイバー防御演習では，与えられた環境に対して運営側から実行される攻撃に対処することでインシデントハンドリング [4] について学ぶ。また，サイバー防御演習では実際にサイバー攻撃を受けるため，サイバーセキュリティの啓蒙にも適して

いる。しかし，サイバー防御演習の開催には多大な費用や労力を必要とするため，サイバー防御演習は特定の人物・組織によって開催されるにとどまり，限られた人数しか演習を受講できない。

1.2 目的

一般に，サイバー演習はサイバー攻撃などの通信や，使用されるマルウェアが実環境に悪影響を及ぼすことを防ぐため，サイバーレンジと呼ばれる仮想環境にて行われる。我々，CROND Project にて開発している CyTrONE[5] は，サイバーレンジの構築とクイズ形式によるフォレンジックなどの静的な演習の実施を可能とする。しかし，サイバー防御演習のような刻一刻と状況の変化する動的な演習はできない。

本研究は，サイバーセキュリティの啓蒙と学習支援のために，演習の進行を自動化することにより，演習の実施に高度な知識を要求される動的な演習を誰でも実施可能とすることを目的とする。本論文では，動的な演習の中でもサイバー防御演習における演習の進行管理に焦点を当て，技術的な学習用サイバー防御演習の実施を簡単にするための自動進行管理システムについて提案し，設計と実装を行う。防御演習において実行されるサイバー攻撃は，演習の目的とシナリオに応じて多種多様であり，すべての要求を満たすことは困難だ。提案システムは，モジュール構造により容易にサイバー攻撃の追加を可能とし，用意された演習シナリオに従って多様なサイバー防御演習を，受講者の演習状況に応じて自動で進行させることを可能とする。

1.3 本論文の構成

本論文は，本章を含めて 5 章で構成される。2 章では，既存のサイバー防御演習を紹介し，学習用サイバー防御演習に求められるものを明らかにする。3 章では，提案システムの設計と実装について述べる。4 章では，提案システムが演習シナリオに従い，様々なサイバー防御演習が提供可能であることを検証する。5 章では，本論文をまとめ，今後の課題について検討する。

第2章　既存のサイバー防御演習

本章では既存のサイバー防御演習について紹介し，既存の演習をもとに学習用サイバー防御演習に必要なものについて検討する。

2.1　Hardening

Hardening[6] とは，Web Application Security Forum(WASForum) によって年に 2 回開催されるセキュリティ堅牢化の競技大会である。競技時間が 8 時間を超えることもある過酷な競技である。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。

Hardening の競技環境は，Alfons[7] を用いて StarBED[8] 内に構築される。また，Hardening におけるサイバー攻撃は，運営がすべて手動で実行している。運営は，参加者の演習環境の動作状況と，会場に設置されたカメラにより参加者の様子を確認しながら攻撃するため，参加者の状況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。

Hardening のコンセプトとして「衛る」技術というものがある。本競技では，複数の仮想マシンにより構築された実環境を模した演習環境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。これにより，技術だけなくセキュリティ業務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[9] 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 セット以上の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，技術的なもの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。

2.3 CYDER

CYDER とは，政府のサイバーセキュリティ戦略等に基づき NICT ナショナルサイバートレーニングセンターによって開催する実践的なサイバー防御演習である。事前オンライン学習と実習とグループワークからなる演習によって構成され，インシデントレスポンス能力の向上を目的とする。サイバーセキュリティ基本法に規定される国の行政機関、地方公共団体、独立行政法人、重要社会基盤事業者等を対象としている。実機を用いてログ解析やフォレンジックによってインシデントを発見し，初動対応や各所への指示や報告，そしてベンダーへの調査依頼などを体験することができる。エンジニアを対象としたセキュリティ技術の向上ではなく，インシデント発生時の事務対応が中心となる。

2.4 セキュリティ系ボードゲーム

様々なセキュリティベンダーなどから，ボードケームやカードゲーム形式のサイバー防御演習が提供されている。例として，Trend Micro のインシデント対応ボードゲーム [10]，Kaspersky の Kaspersky Interactive Protection Simulation[11]，JNSA 教育部会のセキュリティ専門家　人狼 [12] などがある。これらは，CYDER よりもサイバーセキュリティにおける事務対応に特化しており，セキュリティ技術は必要とされず，ゲーム形式により提供される。また，紹介したもののうち，インシデント対応ボードゲームとセキュリティ専門家　人狼は無料でダウンロード可能であり，独自に実施することも可能であるため，Hardening などと比較して取り組みやすい。

2.5 CyTrONE

CyTrONE とは，セキュリティ技術者を目指す初心者から中級者を育成することを念頭に置いて，JAIST の CROND Project によって開発されたサイバーセキュリティ演習用のフレームワークである。CyTrONE の概要を図 2.1 に示す。CyTrONE は，サイバーレンジを作成する CyRIS と，クイズ形式によるサイバー演習を提供

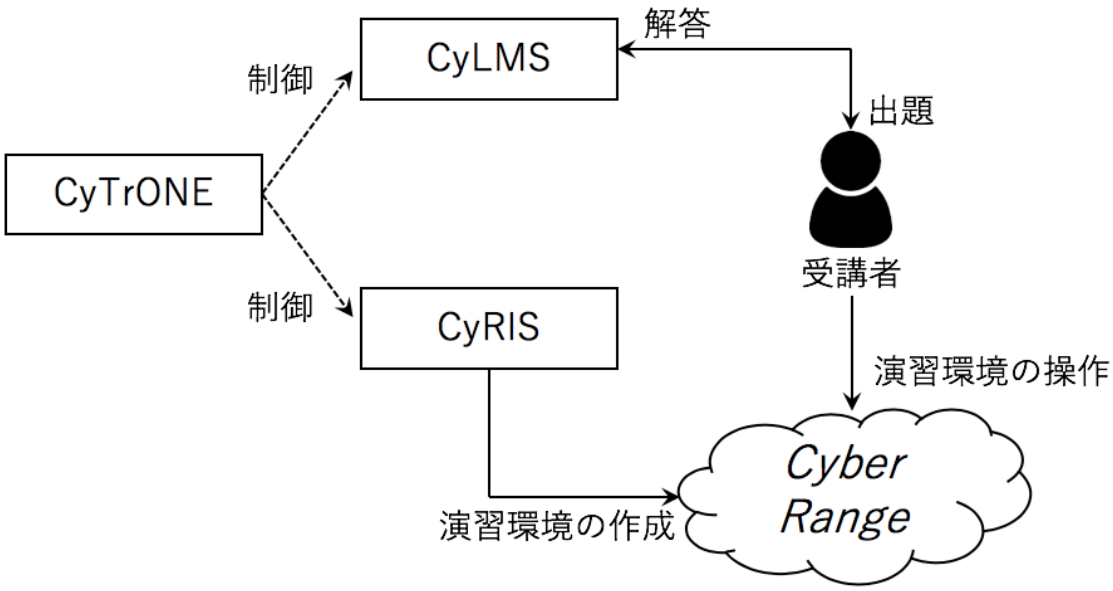

図 2.1: CyTrONE の概要

する CyLMS によって構成される。CyRIS は設定ファイルの記述に従って KVM を用いて仮想マシン作成し，作成したイメージを複製することにより演習環境を作成する。その際に，IP アドレスは CyRIS が自動で設定する。CyLMS は，YAML 形式で記述された問題文，ヒントおよび解答を学習管理システム用に変換し，学習管理システムの 1 つである Moodle[13] を用いてクイズ形式による演習を提供する。これにより，イメージファイルを用意し，設定ファイルを記述するだけでサイバー演習を実施することが可能である。

2.6 技術的なサイバー演習の比較と学習用サイバー防御演習の課題

Hardening や Micro Hardening は技術を重視した演習であり，CYDER やセキュリティ系ボードゲームはインシデント発生時の事務対応を重視した演習である。事務対応を学習するためのサイバー防御演習はすでに無料で配布され，誰でも開催可能となっている。しかし，技術を学習するための重視した演習は，特定の個人・組織によって開催されるにとどまっている。そのため，技術的な演習も誰でも開催可能となる必要がある。

表 2.1 に既存の技術的な演習の特徴を示す。Hardening は，運営側が手動で攻撃を実行しているため，柔軟な進行が可能である。しかし，Micro Hardening と CyTrONE は決められた通りにしか動作しない。Hardening は，多くのスポンサー企業およびエキスパートの協力により成り立っているため，開催が困難だ。逆に，

	Hardening	Micro Hardening	CyTrONE
進行の柔軟さ	◎	×	×
演習進行の負担	×	◎	◎
開催の容易さ	×	◎	◎
学習要素	△	○	◎
演習の用途	訓練	訓練	学習
演習の自由度	×	×	◎

表 2.1: 既存の演習の特徴

Micro Hardening や CyTrONE は演習環境の作成から演習の実施まで 1 人でも実施可能なように簡単化されているため，演習進行の負担が軽く，また開催も容易だ。Hardening や Micro Hardening は訓練を目的として開催される。そのため，参加者が自らの知識や技術を試すための場であり，新たな技術を学ぶ場ではない。CyTrONE は，クイズ形式での演習により，受講者に対して学習の場を提供することが可能だ。また，Hardenig や Micro Hardening と違い，CyTrONE は無料で配布されており，誰でも自由に，様々な演習を作成することが可能だ。ゆえに，セキュリティ教育のためには，Hardening の柔軟な進行を Micro Hardening のように自動化し，CyTrONE のような自由度と学習機能を持つ学習用サイバー防御演習が必要である。

第3章　DeTManの構成

本章では，サイバー防御演習進行管理システム (cyber Defense Training progress Management system) の設計について述べる。DeTMan は，サイバー防御演習において手動による柔軟な演習進行を自動化すると同時に，学習用の機能を追加することにより，学習用サイバー防御演習を誰でも実施可能とすることを目的とする。DeTMan は，図 3.1 に示すようにサイバーレンジ上で動作し，受講者に対してサイバー攻撃を実行するために，各受講者の環境の外側で動作する。

3.1　柔軟な進行の実現

本章では，DeTMan において，柔軟な演習進行をどのように実現するかについて述べる。

3.1.1　適切なアクションの選択

柔軟な進行の実現には，まず受講者に応じて実行するサイバー攻撃を選択する必要がある。DeTMan は 2 つの手法を用いて，適切なサイバー攻撃を選択する。

アクションの成否に応じた進行の選択

一般のサイバー防御演習において，運営は受講者の能力を正確に把握していない。そのため，運営は実際に攻撃を行い，その成否により受講者の能力を判断し，受講者の能力に合わせた難易度の攻撃を選択する。ゆえに，DeTMan は，攻撃の成否に応じて次に実行する攻撃を選択する必要がある。

また，サイバー攻撃には順序がある。ここでは，Linux サーバを攻撃対象とした SSH における辞書攻撃を例にあげる。攻撃の順序を図 3.2 に示す。まずは，リスト攻撃によりログイン可能なユーザが存在するか調査する。ログイン可能なユーザが発見できなければ，攻撃を終了する。ログイン可能なユーザを発見した場合，ログイン可能なユーザが sudo コマンドを使用する権限を持つか調査する。sudo 権限があれば sudo を用いて攻撃 A を実行し，sudo 権限がなければログインユーザの権限により攻撃 B を実行する。ログインに失敗した場合は，以降の攻撃は無意味であり，sudo 権限がないにも関わらず sudo を使用したり，sudo 権限があるにも

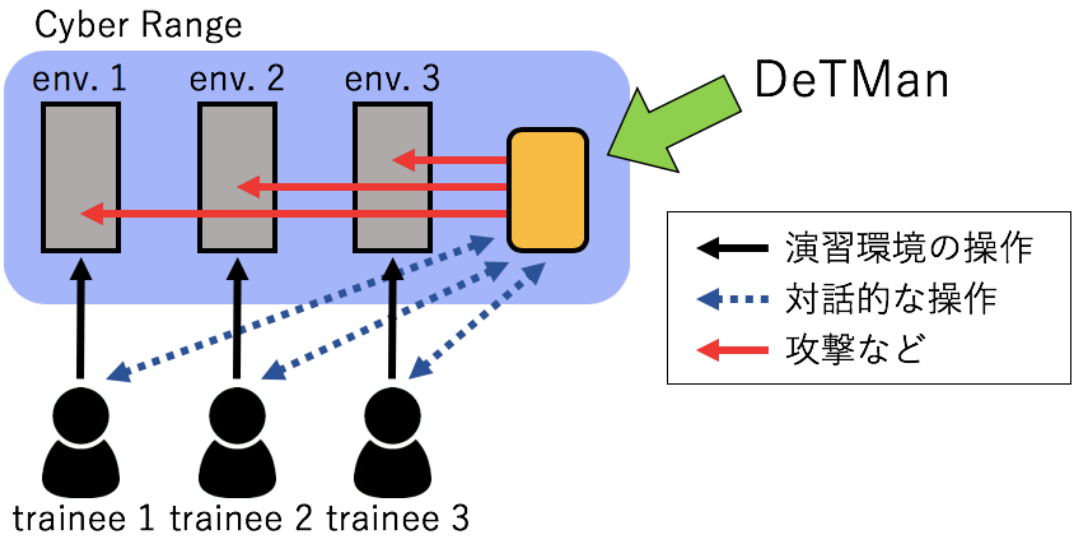

図 3.1: DeTMan

関わらず使用しないこともまた，無駄である。アクションの成否に応じた進行の分岐により，受講者の能力に合わせた攻撃の選択だけでなく，無駄なアクションの実行も防ぐことが可能となる。

サイバー攻撃群の分割

サイバー防御演習内では多種多様なサイバー攻撃が実行される。そして，すべてのサイバー攻撃が関連性を持つということは少ない。本論文では，前述した辞書攻撃を発端とした一連のサイバー攻撃を，1 つの攻撃群と定義する。他にも，例として HTTP サーバに対するサイバー攻撃を取り上げる。同じ HTTP サーバへのサイバー攻撃でも，SSH などの別のサービスへの攻撃や apache や nginx への攻撃，WordPress のような CMS への攻撃などがある。演習シナリオは複数の攻撃群によって構成され，実行される攻撃群は攻撃群のリストからランダムで選択される。これにより，実行される攻撃が受講者間で異なるため，カンニングのようなことはできなくなると同時に，次に実行される攻撃が予測できなくなる。また，演習シナリオを作成する際に，攻撃群を用意するだけで良くなるため，作成者の負担を軽減することができる。

防御に成功した攻撃群はリストから削除され，防御に失敗した攻撃群はリストに残る。そのため，受講者が防御に失敗した攻撃群は，防御に成功するまで繰り返し実行される。これにより，失敗することがわかっている攻撃群 (受講者が防御に成功した攻撃群) の実行を防ぐことができるため，演習の密度をあがることが期待できる。

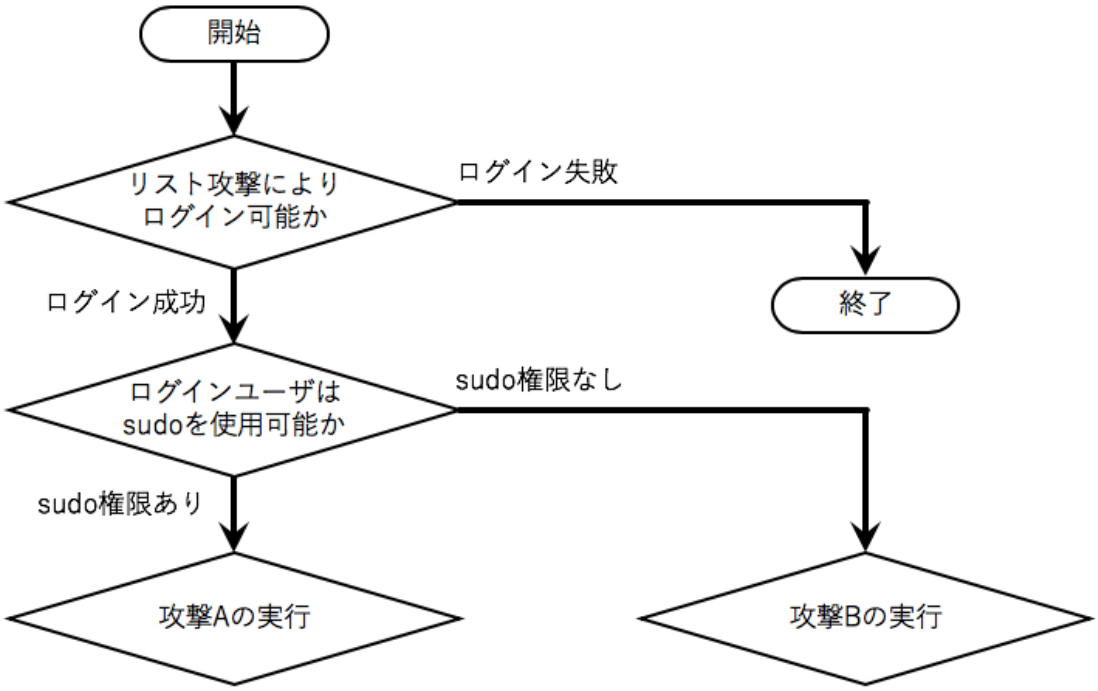

図 3.2: 進行の分岐

3.1.2 サイバー攻撃を実行するタイミングの調整

柔軟な進行の実現には，適切な攻撃を適切なタイミングで事項する必要がある。図 3.3 に，概要を示す。サイバー攻撃を連続で実行した場合は，攻撃による被害からの復旧が終わる前に次の攻撃が実行されてしまう。これは，柔軟な進行ではないと同時に，停止しているサービスに対して攻撃を実行する可能性があるなど，無駄が多い。そのため，復旧が完了するまで攻撃の実行を待機する必要がある。しかし，柔軟な進行の実現には，これだけでは不十分だ。ゆえに，サービスの復旧後からさらに，任意のタイミングまで攻撃を待機する必要がある。

3.2 DeTMan の設計

本節では，DeTMan の設計について説明する。DeTMan は，専門知識を持たない人であっても 1 人でサイバー防御演習が開催可能とするため，システムの導入や使用が簡単となるように設計する。DeTMan は Python が使用可能であり，Windows および macOS，Ubuntu，CentOS 上で動作する。そのため，サーバ上にまとめて構築された仮想マシン環境を用いて行われていた従来の防御演習に対して，DeTMan は図 3.4 に示すように，監督者のコンピュータ上で実行し，受講者のコンピュータ上に作成された仮想マシンに対して攻撃を行うといった演習の開催が可能である。これにより，サーバなどの高価な機材を用いることなくサイバー防御演習を開催

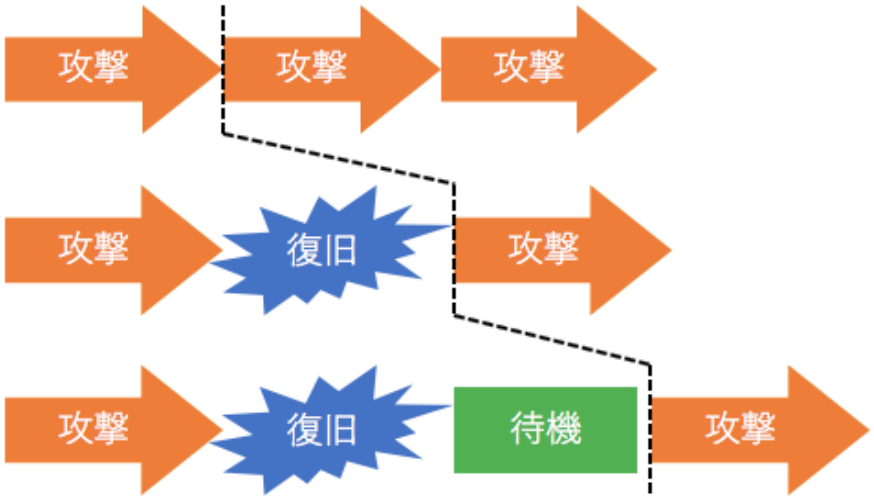

図 3.3: サイバー攻撃を実行するタイミングの調整

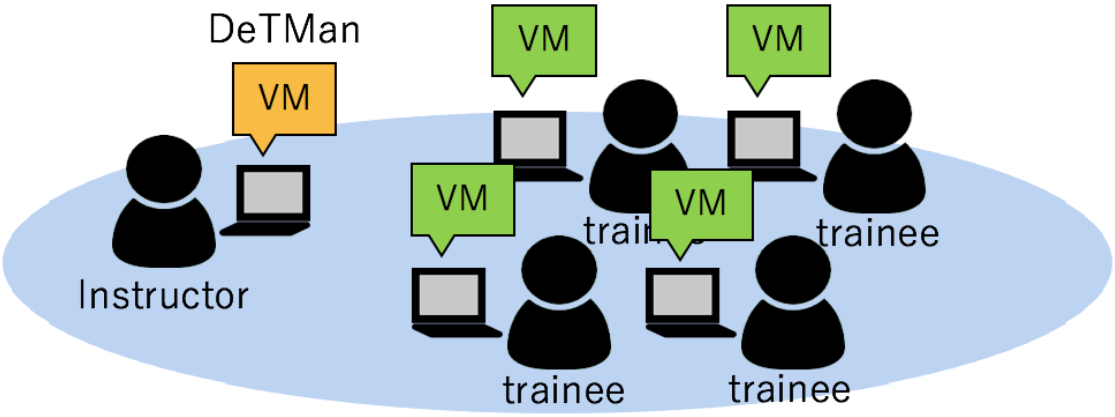

図 3.4: 演習環境

することが可能となる。ただし，受講者側から DeTMan に対する通信が発生するため，何らかの悪影響が発生することも考えられる。そのため，DeTMan は監督者のコンピュータ上に構築された仮想マシン上で動作させることを推奨する。

DeTMan の基本構造を図 3.5 に示す。DeTMan はトリガー，アクション，そしてアクションの結果による進行の分岐を進行管理の基本とする。また，トリガーおよびアクション実行直前にサービスの監視を行う。この 1 サイクルをステップと定義する。

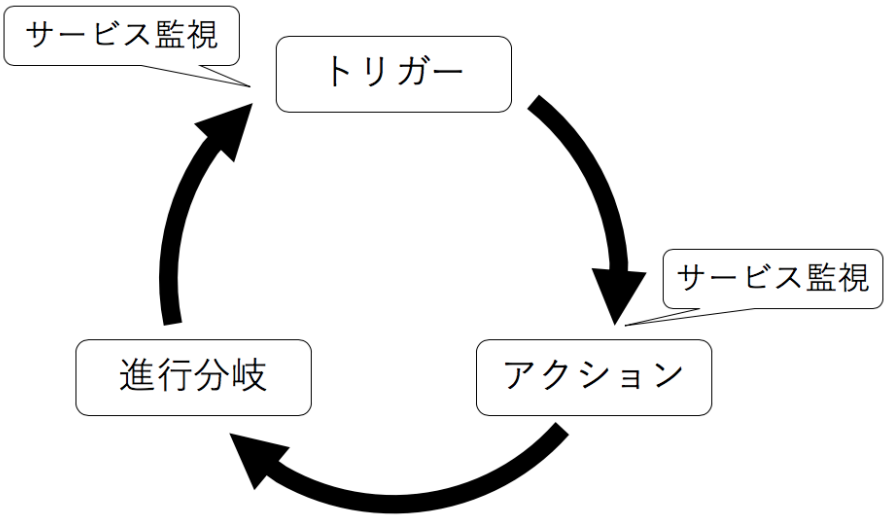

図 3.5: DeTMan の基本構造

3.3 システム構成

DeTMan の概念図を図 3.6 に示す。受講者の持つ能力や，演習中における行動は受講者それぞれで異なる。受講者ごとに進行をさせるため，DeTMan は受講者と同数の training プロセスを生成し，生成された training プロセスが演習の進行を管理する。training プロセスは互いに独立しているため，ある受講者の演習が他の受講者の演習進行に影響を及ぼすことは基本的にない。

3.4 main プロセス

main プロセスの動作を図 3.7 に示す。main プロセスは，主にサイバー防御演習の実施における初期設定を行う。

3.4.1 設定ファイル

DeTMan は，演習シナリオが記述されたシナリオファイルと，受講者および攻撃対象の情報が記述されたターゲットファイルの 2 種類の設定ファイルを持つ。

シナリオファイル

シナリオファイルには，ステップを YAML 形式によって記述される。シナリオファイルのサンプルを図 3.8 に，また，その動作を図 3.9 に示す。シナリオファイルには，より簡単に記述できるように省略可能な項目があり，省略された項目を

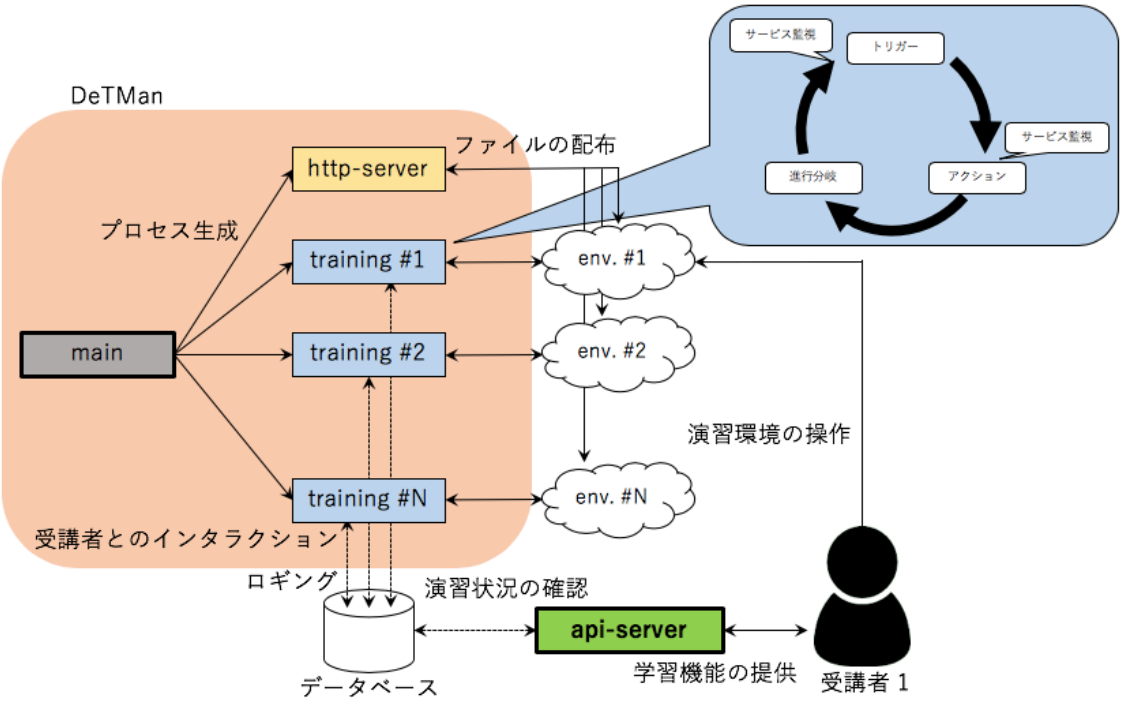

図 3.6: DeTMan の概念図

main プロセス内で補完してファイルに書き出し，完全なシナリオファイルを作成する。省略可能な項目は trigger，success，failure の 3 つである。trigger が省略された場合，trigger として実際には何も動作しない none モジュールが補完される。success が省略された場合は，次のステップとして通常は同じシナリオファイル内の 1 つ下の step が補完される。ただし，省略したステップがシナリオファイルの最下部であった場合は SUCCESS が補完される。failure の場合も，success と同様に補完されるが，省略したステップがシナリオファイルの最下部であった場合は FAILURE が補完される。他にも，同じファイル内で同じことを繰り返し記述する手間を省くため，action 内のオプションは次ステップへの継承が可能である。

DeTMan では，シナリオファイルは特定のファイルではなくディレクトリを指定し，ディレクトリ内にある拡張子が yml であるすべてのファイルを読み込む。1 つのシナリオファイルが，前述の 1 つの攻撃群に該当する。1 つのシナリオファイルにすべてのステップを記述する場合では，図 3.10 に示すように攻撃が失敗するまで次の攻撃に移ることができない。DeTMan は，これをシナリオファイルを複数持ち，次のステップに SUCCESS または FAILURE を使用可能とすることで解決した。図 3.11 のように，シナリオリストの中からランダムでシナリオファイルを選択して実行する。当然，次に実行するシナリオファイルを指定することも可能である。次のステップが SUCCESS の場合は，シナリオリストの中から選択したファイルを削除する。これにより，関連性のない攻撃群間の連携を考えることなくシナリオが作成可能であると同時に，同じ攻撃の繰り返しではなく，防御に

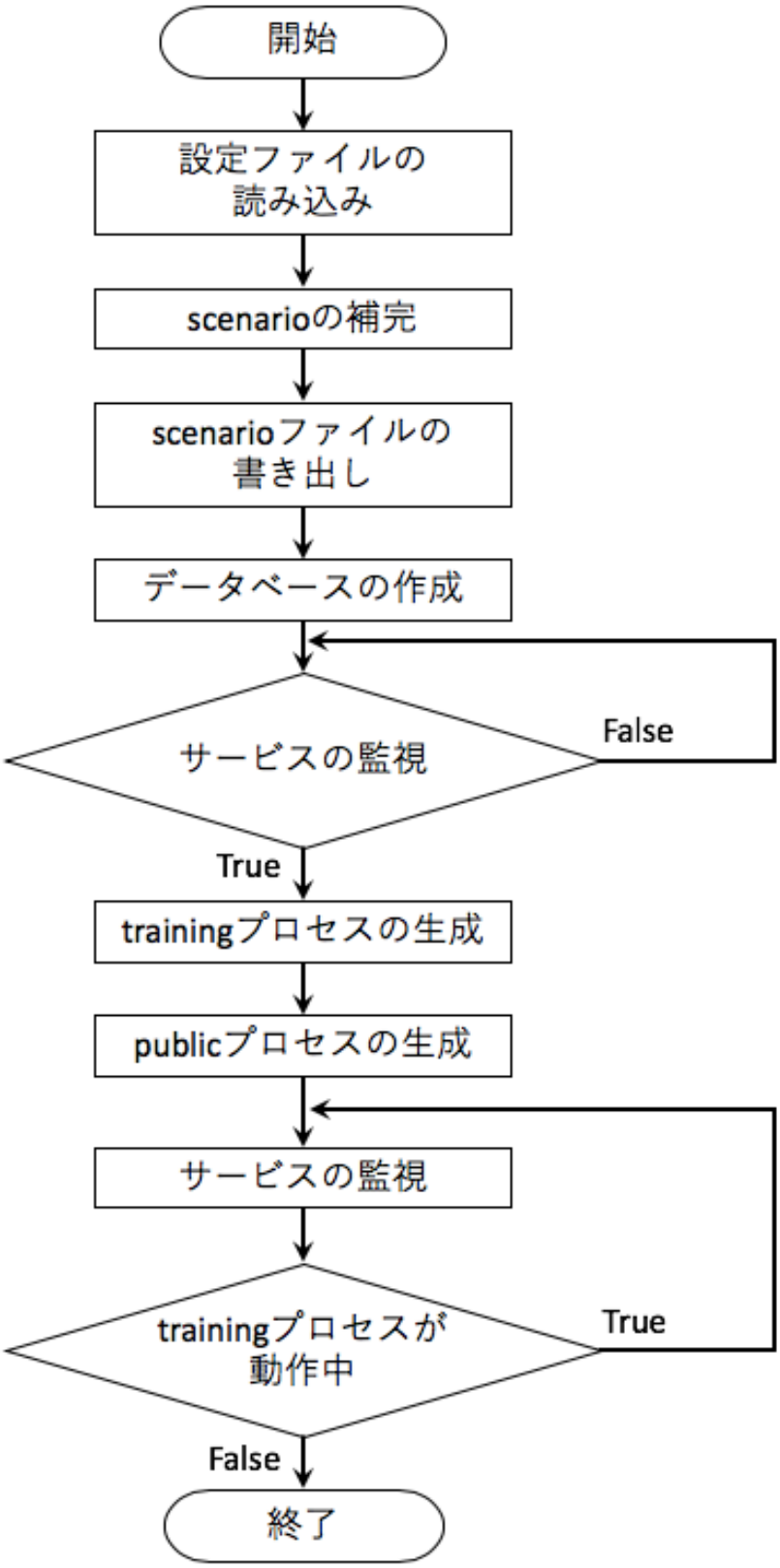

図 3.7: main プロセスの動作

成功するまで柔軟に攻撃を繰り返すことができる。

ターゲットファイル

ターゲットファイルは Python2.7 においてデフォルトで組み込まれている ConfigParser モジュールを利用した INI 形式によって記述される。シナリオファイルのサンプルを図 3.12 に示す。左側の HTTP-server の部分をターゲットと定義する。図 3.8 における target の部分にこのターゲットを記述すると，右側に書かれた IP アドレスに受講者に対応した training プロセスが攻撃を実行する。

3.4.2 データベース

DeTMan は，演習中および演習後の振り返りのためおよび，演習状況を外部から参照可能とするために，演習状況をデータベース内に保管する。データベースは，以下の 4 つのでテーブルを持つ。

- 攻撃対象のサービス稼働状況を保管する state テーブル
- 演習の進行状況について保管する progress テーブル
- アクションの実行履歴を保管する log テーブル
- 学習機能を含めた受講者との連携に用いる情報を保管する board テーブル

board テーブルは情報の保管だけでなく，アクションおよびトリガーとも連携して動作する。データベースには，Python に組み込まれている SQLite を使用した。これにより，追加でソフトウェアをインストールする必要がなくなると同時に，SQLite はデータベースを 1 つのファイルとして持つため，演習結果の移動が容易となる。

3.4.3 サービスの監視

受講者の状況に合わせた進行を実現するためには，受講者の演習状況を把握するために受講者の動作を監視する必要がある。しかし，受講者を監視するために，何らかのソフトウェアを仕込むことは避けなければならない。これは，捉え方を変えるとマルウェアそのものだからだ。受講者が演習環境上で動作する不審なプロセスを発見して，動作を停止した場合，演習が停止してしまう。そのため，DeTMan はサイバーレンジ内かつ受講者が管理する環境の外側からサービスの監視を行うことにより，受講者の演習状況を判断する。

サービスに何らかの異常が発生している場合は，サイバー攻撃からの復旧が未完了または受講者に何らかのアクシデントが発生していると判断可能だ。あるサー

```
scenario:
 - step: attack1
   target: HTTP-server
   action:
    module: attackA
   failure:
    next: QED

 - step: attack2
   target: HTTP-server
   action:
    module: attackB
   failure:
    next: attack4

 - step: attack3
   target: HTTP-server
   action:
    module: attackC
   success:
    next: FAILURE

 - step: attack4
   target: HTTP-server
   action:
    module: attackD
   success:
    next: FAILURE
   failure:
    next: FAILURE
```

図 3.8: サンプルシナリオファイル

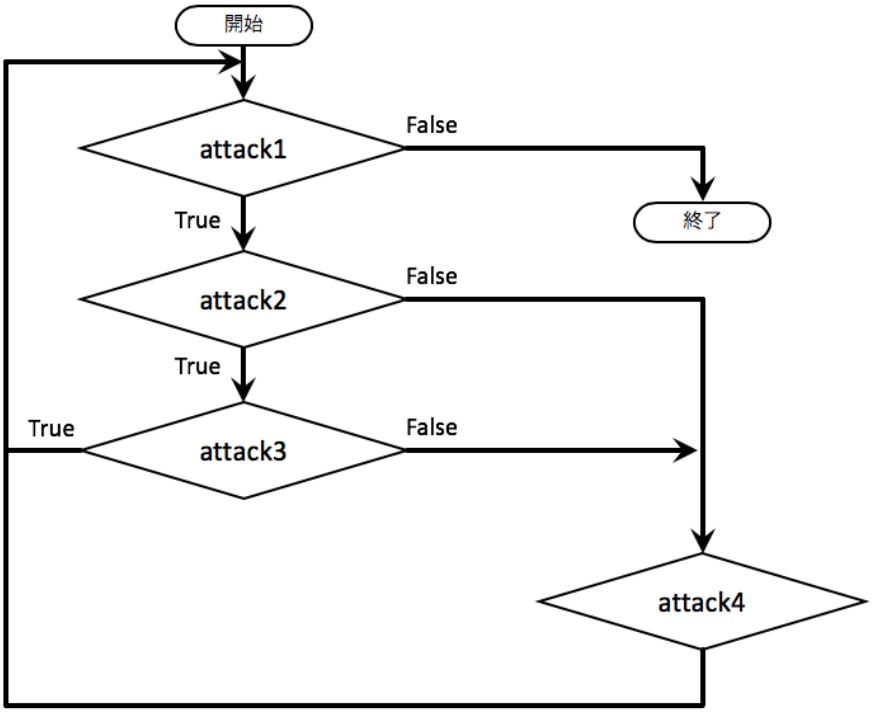

図 3.9: サンプルシナリオの動作

ビスへのサイバー攻撃を行う際に攻撃対象となるサービスが停止していた場合は，サイバー攻撃への対策の有無に関わらず攻撃が失敗する。また，同種のサイバー攻撃を繰り返す場合，例としてサイバー攻撃として WEB ページの改ざんを行う際，前回の攻撃により WEB ページが改ざんされたままではあまり意味がないように，攻撃からの復旧が完了していない場合は効果が薄くなる。そのため，アクションの実行時にサービスが正常に動作していることを保証するため，サービスが復旧するまで演習の進行を待機させる必要がある。また，前章において説明したトリガーの目的の 1 つとして，アクションを実行するタイミングの調整があった。しかし，受講者に何らかのアクシンデントが発生している際にトリガーの待機をしている場合，アクシデントから復旧した瞬間にアクションが実行される可能性がある。これでは，トリガーの意味が薄くなってしまう。よって，トリガーとアクションの直前にサービスの監視を実行する。

DeTMan は，ターゲットファイルに記述されたサービスについて監視を行う。サービスとは，ターゲットファイルに記述されたターゲットの-(ハイフン) の右側に記述されたものである。記述されていない場合は，ネットワークの疎通確認のみを行う。

サービスの確認では，対象となるサービスが正常に動作しているかを確認するために行われる。しかし，サイバー攻撃による被害を一意に定義することはできないため，サービスの正常な状態についてサイバー防御演習の運営以外が定義す

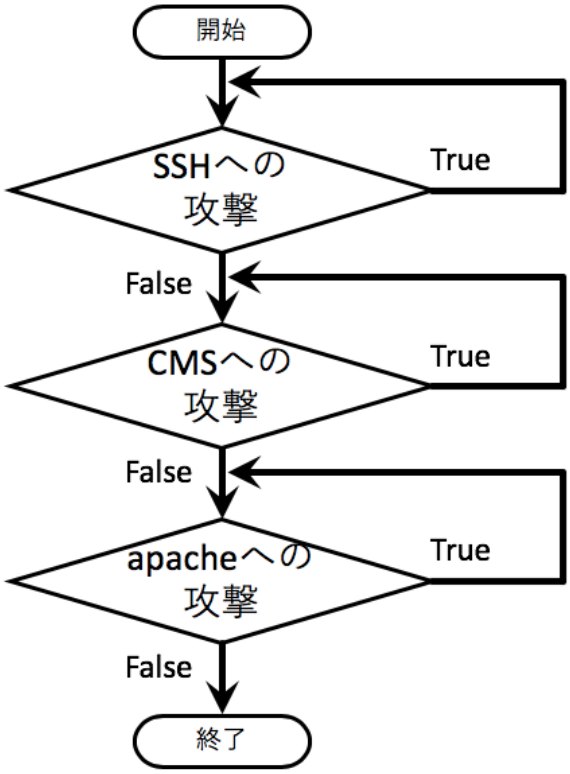

図 3.10: シナリオファイルが 1 つの場合

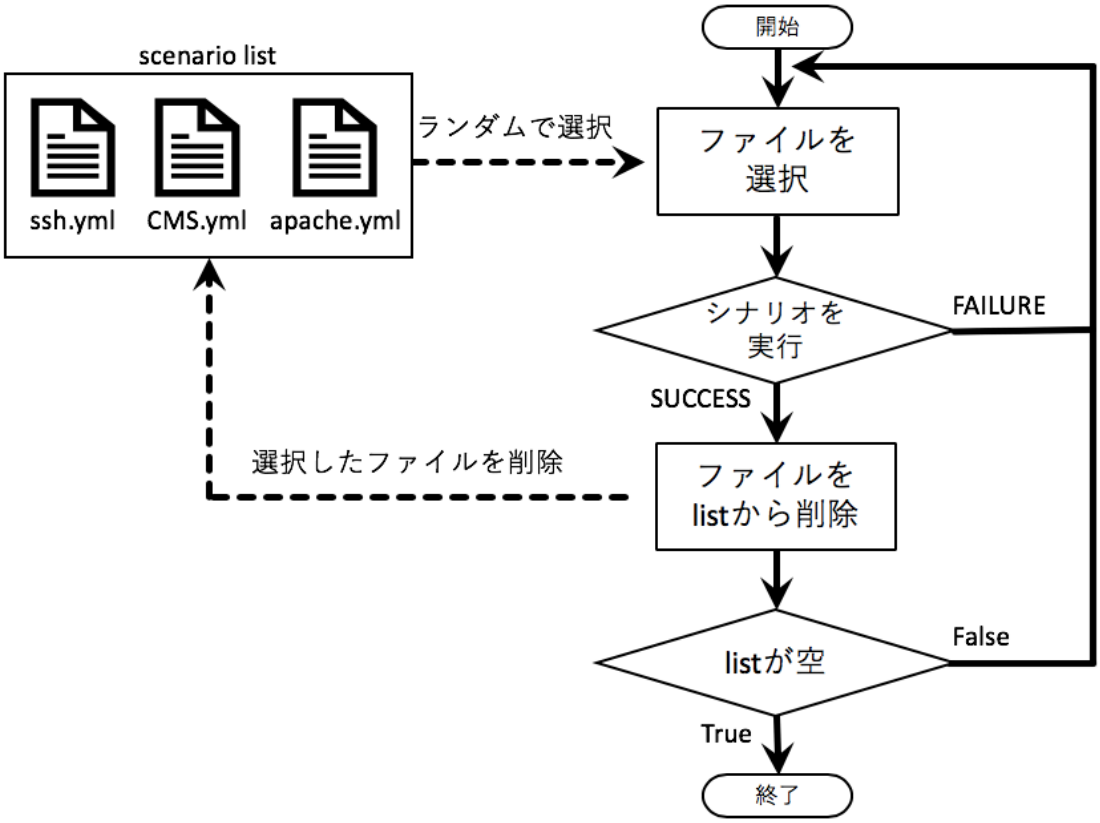

図 3.11: シナリオファイルが複数の場合

```
[trainee 1]
HTTP-server = 192.168.1.101
mysql-server = 192.168.1.101
DNS-server = 192.168.1.102

[trainee 2]
HTTP-server = 192.168.2.101
mysql-server = 192.168.2.101
DNS-server = 192.168.2.102

[trainee 3]
HTTP-server = 192.168.3.101
mysql-server = 192.168.3.101
DNS-server = 192.168.3.102
```

図 3.12: サンプルターゲットファイル

ることは困難だ。DoS 攻撃や受講者の誤操作によりサービスが停止している場合は，死活監視により容易に判断可能だ。だが，例として，WEB ページの改ざんは，サービスの死活監視だけではサービスが復旧していないにも関わらず，復旧していると判断してしまう。また，外側から受講者による書き換えとサイバー攻撃による改ざんの違いを DeTMan にあらかじめ組み込むことも困難である。WEB ページの改ざんにおいて，どの WEB ページをどのように改ざんするのかは，演習の運営次第である。他にも，サイバー攻撃の被害から復旧する前に，次のサイバー攻撃による被害が発生した場合はどこから手をつけていいのか受講者が判断に迷うことが考えられる。そのため，確実に 1 つ 1 つ対処させるために復旧が終わるまで演習の進行を待機させたいと考える運営がいるかもしれない。しかし，トリアージについて訓練するために，逆に進行を待機させたくないと考える運営がいるかもしれない。これもまた，運営次第である。DeTMan では，サービスとプロトコル (TCP・UDP)・ポート番号および監視用関数を辞書形式にて記述することにより，サービスの監視を容易に追加可能である。サービス名が，辞書内のサービス名と一致した場合，辞書内に記述されたプロトコルを用いて，記述されたポート番号に接続する。同時に，監視用関数が記述されていた場合は，その関数を実行する。この関数は，特定のファイルにサービス名と同名の関数を作成するだけで自動でインポートされ，実行される。これにより，運営側が自由にサービスの監視について定義可能となる。

他にも，すべての攻撃対象となるサービスの稼働状況が確認できなければなら

ないのは問題だ。例として，WEB サーバと連携して動作するデータベースサーバを挙げる。データベースは当然攻撃対象である。サービスを正常に提供するためにはデータベースサーバは必ず正常に動作しなければならない。しかし，データベースサーバは外部からアクセス可能である必要はない。ファイアウォールなどにより WEB サーバからのみデータベースサーバへのアクセスを許可した場合，サービスは正常に提供可能であるにもかかわらず，データベースは正常に動作していないと判断される。攻撃対象となるサービスすべての稼働が外部から確認できなければならないことは，進行の待機において問題だ。しかし，これもまた，運営側次第である。そのため，DeTMan はサービス名が大文字で記載されていた場合，必須のサービスと判断し，サービスに何らかの異常が確認された場合は進行を待機する。サービス名が小文字で記載されていた場合，サービスの監視は行うが，何らかの異常が確認された場合も進行を待機しない。これにより，サービス名を大文字で記述するか，小文字で記述するかだけで，容易に対応可能となった。

main プロセスでは，2 つのサービスの監視がある。1 つ目は，ターゲットファイルのエラーチェックおよび，演習環境の最終確認を目的として行われる。これは，training プロセスにおいて行われるサービスの監視と同様のものである。2 つ目は，データベースの更新を目的として行われる。2 つ目のサービスの監視を行わなかった場合，1 つ目のサービスの監視が完了して以降は，図 3.6 に示すように training プロセス内のトリガーとアクションの直前にサービスの監視が行われる。しかし，トリガーまたはアクションに時間がかかる場合は，データベース内に保管されるサービスの稼働状況は現在のものではなくなる。そのため，training プロセス生成後に，すべての training プロセスが終了するまで main プロセスでもサービスの監視を一定間隔で実行し，データベース内の情報を最新のものに保つ。また，この 2 つ目のサービスの監視は，データベースの更新のみを目的とするため，演習の進行に一切の影響を及ぼさない。

3.5 training プロセス

training プロセスの動作を図 3.13 に示す。training プロセスは，受講者と 1 対 1 で動作し，演習の進行管理を行う。

3.5.1 シナリオファイルの選択

training プロセスでは，関連性のないサイバー攻撃間の接続を考慮せずシナリオを作成するため，および成功した (防御に失敗) サイバー攻撃群を同じ攻撃を繰り返し続けることなく，再び実行するためにシナリオファイルのリストからランダムでシナリオファイルを選択し，実行する。ただし，図 3.13 には，2 つのランダムでシナリオファイルを選択する工程があるが，1 度だけ実行される上部にある工

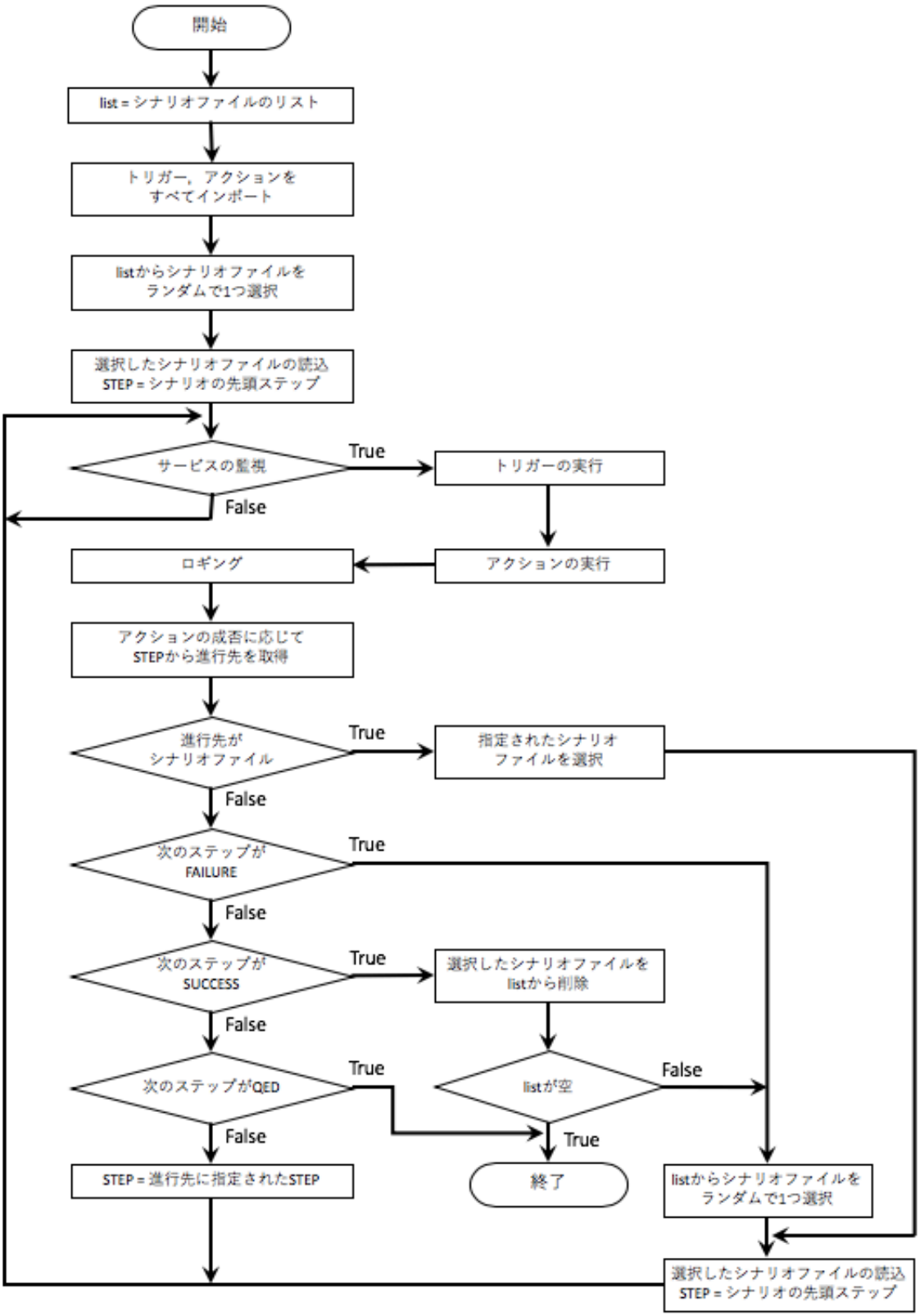

図 3.13: training プロセスの動作

程は僅かに動作が異なる。例えば，受講者はシステム管理者として演習を受講するなどのシチュエーションの説明や，受講者の任意のタイミングで演習を開始するための演習開始ボタンの表示などのように，演習開始時に 1 度だけ実行したいシナリオが存在する場合がある。図 3.13 の上部の工程は，シナリオファイルのリスト内に scenario.yml というファイルが存在した場合，ランダムではなくこのファイルを選択する。

3.5.2 トリガー

DeTMan は，サービスの監視だけでなく，トリガーを用いて，アクションを任意のタイミングで実行させることが可能だ。受講者が特定の動作を行うまで演習の進行を待機させることにより，演習の進行に受講者自身の動作を関与させ，受講者の状況に合わせた進行を実現する。これには以下のようなものが想定される。

- 一定時間待機する
- 受講者がクリックする
- 特定の文言が含まれるメールを受信する

また，トリガーの目的はこれだけではない。トリガーを用いることにより，受講者の動作から始まるサイバー攻撃を再現するが可能となる。これには以下のようなものが想定される。

- 罠サイトへアクセスする
- 受講者がマルウェア (リバースシェル) を実行し，受講者からの通信が発生する

ただし，待ち受けている期間中は演習の進行が停止するため，長い期間待ち受けることは困難だ。そのため，サイバー攻撃の再現に関しては今後の課題となる。

trainig プロセスは指定のディレクトリ内に存在するすべてのファイルをインポートする。そのため，トリガーモジュールを追加する際には，指定の構造で作成し，指定のディレクトリに配置するだけで使用可能となる。トリガーモジュールの構造は，ファイル名と同名のクラスを作成し，クラス内にシナリオファイルの構文チェックに使用する check 関数とトリガーとして実行される trigger 関数を持つ。ただし，モジュール内で pip コマンドなどによりインストールした外部のモジュールを使用する場合は注意が必要である。演習において使用しないモジュールもインポートするため，ディレクトリ内に存在するすべてのモジュールが必要とする外部モジュールをインストールしなければならない。これは，import 文を関数内に記述することにより回避可能である。トリガーの実行では，シナリオファイルの module に記述された関数が実行され，関数の実行終了をトリガーの発生だと判断し，演習が進行する。

3.5.3 アクション

アクションは，ステップ内において中核となる部分である。アクションでは，サイバー攻撃の実行や指導の実施のように，受講者に対して何らかの動作を行う。

アクションも，トリガーと同様に指定の構造で，指定のディレクトリにモジュールを配置することにより使用可能となる。アクションのモジュールは，戻り値を持たないトリガーと異なり，3 つの戻り値を持つ。1 つ目は，アクションの成否を示す Bool 値の result である。True，False は success，failure に変換され，進行の分岐に使用する。2 つ目は，文字列型の comment である。DeTMan にとっては，サイバー攻撃もそれ以外，例えばメールの送信なども同じアクションである。そのため，サイバー攻撃の成功もメールの送信成功も，result は同じ True である。しかし，サイバー攻撃の成功とは防御の失敗であるため，メールの送信とは異なり，アクションが失敗する方が望ましい。演習の進行状況を確認する際に，result が同じでもアクションによって意味が変わっては分かりにくいため，アクションのモジュールは Bool 値による成否だけでなく，人間にとってわかりやすい文章を comment として返す。ロギングの際には，アクションの実行結果として，result ではなく comment が格納される。3 つ目は，data である。図 3.2 のようなシナリオを実現するためには，アクションの結果入手した情報，ここではログイン可能なアカウント情報を次のステップに渡す必要がある。この情報の受け渡しに，この data を使用する。型は特に指定しない。

3.6 学習用の機能の提供

学習用の機能は，アクションの 1 つとして提供される。DeTMan は，受講者の演習状況に応じて進行を分岐させていくため，受講者によって実行されたサイバー攻撃が異なる場合がある。例えばクイズ機能を提供する場合，あるサイバー攻撃に関して出題した際に，受講者によってはその攻撃を受けていない事態が想定される。学習機能をアクションの 1 つとすると，演習の進行の一部として記述可能であるため，この問題を解決可能だ。また，クイズの正解・不正解をアクションの成否とみなすことにより，受講者の理解度に合わせて演習の進行を分岐させることができる。これにより，クイズに正解するまで攻撃を繰り返したり，不正解の場合にはヒントを表示するといった動作も可能である。

3.7 http-server プロセス

http-server プロセスは，HTTP サーバとして動作する。主にダウンローダ型のマルウェアが，別のマルウェアをダウンロードする際に使用することを想定している。ダウンローダ型のマルウェアとは，マルウェアをダウンロードするための

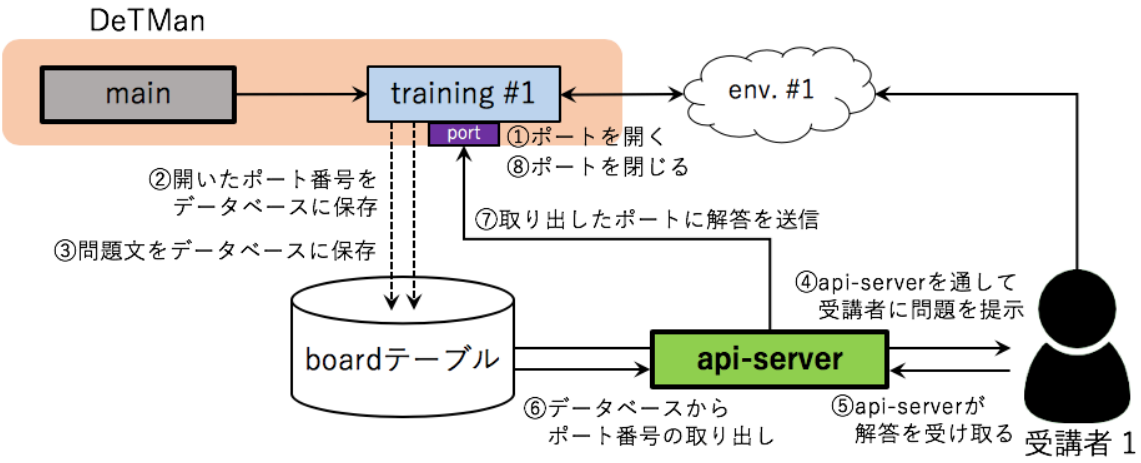

図 3.14: server-api と DeTMan の連携

マルウェアである。他にも，サイバーレンジ内で，演習に使用するファイルの配布などにも用いることができる。

3.8 api-server

api-server は，DeTMan の API として動作し，外部から HTTP アクセスがあった場合，アクセスされたパスに応じてデータベース内の情報を JSON 形式にて送信する。また，学習用の機能を提供するための機能も提供する。api-server は，演習終了後などの DeTMan が動作していない期間であっても，動作可能とするため，DeTMan とは別に動作する。データベースとして使用している SQLite は，MySQL などのように外部と直接通信することはできない。そのため，api-server は，DeTMan と同じマシン上で動作する。api-sever は，DeTMan と受講者のインタラクションを確保するためと，学習用機能の提供のために DeTMan と連携して動作する。例として，図 3.14 に，クイズ機能を提供するための api-server と DeTMan の連携を示す。DeTMan と api-server はデータベースと scoket 通信により連携する。training プロセスごとに異なるポートを使用するため，複数の受講者にも対応可能である。データベースのみで連携した場合と比較して，socket 通信を用いた場合は，受講者の動作に対して素早く動作することが可能だ。例では，クイズの解答を通信したが，他にも利用可能である。また，api-server には，サンプルとしてシンプルな WEB UI も組み込まれている。

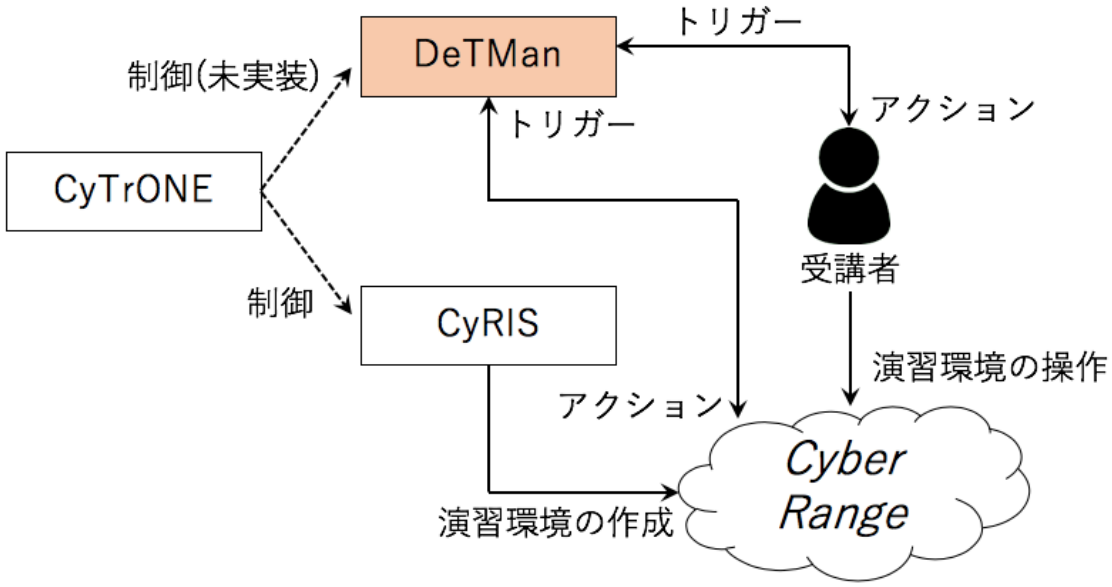

図 3.15: CyTrONE と DeTMan の連携

3.9 CyTrONE との連携

CyTrONE は，クイズ形式によるサイバー演習を提供するが，刻一刻と状況が変化する動的な演習や，実際に演習を行うまで演習がどのように進行するかわからない柔軟な演習には対応できない。そのため，CyTrONE では，柔軟に進行するサイバー防御演習は困難である。この課題は CyTrONE は Moodle を用いた静的な演習を目的としていることが原因であるため，DeTMan は図 3.15 に示すように Moodle の代わりに動作する。ただし，CyTrONE による DeTMan の制御方法については，検討中である。

CyTrONE は演習全体を制御するためのものであり，DeTMan は実際の動作としては CyRIS と連携する。CyRIS と DeTMan の連携を図 3.16 に示す。CyRIS は，設定ファイルに従ってサイバーレンジを作成するが，サイバーレンジの作成後に range_details ファイルと呼ばれるファイル作成する。range_details ファイルには，CyRIS が作成したサイバーレンジ内の仮想マシンの IP アドレスやネットワークに関する情報が記述されている。DeTMan は，このファイルをターゲットファイルの代わりに用いることが可能だ。これにより，DeTMan が使用するシナリオファイルが準備できていれば，サイバーレンジの作成からサイバー防御演習の実施まで行うことができる。また，api-server を用いて受講者とのインタラクションを確保している。そのため，api-server より得られる JSON 形式による情報を可視化することにより，CyTrONE のような演習をどう的なサイバー防御演習の中で提供可能である。

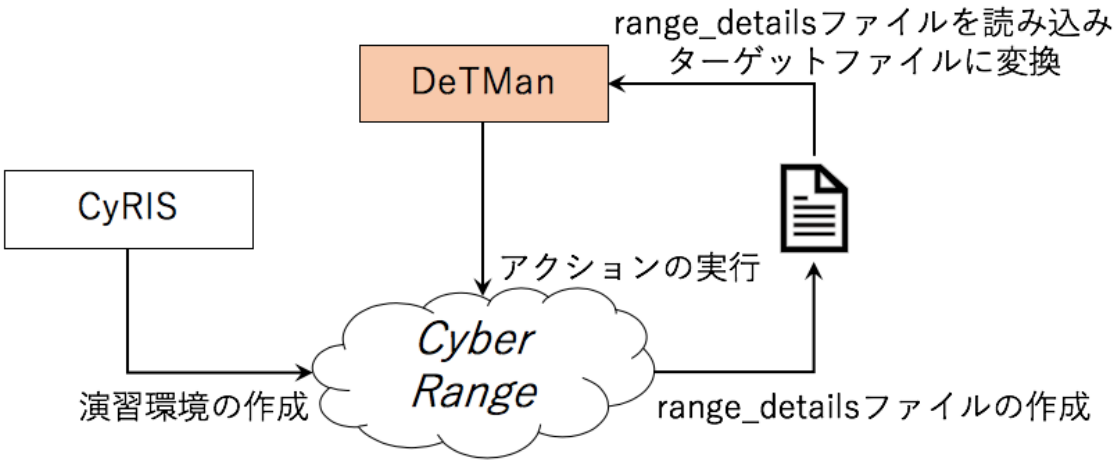

図 3.16: CyRIS と DeTMan の連携

第4章　実験

本章では，DeTMan を用いて，サイバー攻撃の自動化と，既存のサイバー防御演習の再現に関する実験を行う。

4.1　サイバー攻撃の自動化に関する実験

本節では，DeTMan が多様なサイバー攻撃を実行可能であることを確認するため，3 つの攻撃群を試行する。

WordPress4.7.0 における REST API の脆弱性を用いた攻撃

WordPress4.7.0 及び 4.7.1 には，REST API の処理に起因する脆弱性が存在する。本実験では，WordPress4.7.0 を使用する。これにより，サーバ上で第三者によりコンテンツを改ざんされる恐れがある。これは，不正な通信を発生させる単純な攻撃であり，1 ステップで構成されている。

実験の結果，WEB サイトの改ざんが確認できた。また，攻撃の成否の判断も正確に行うことができた。

OS コマンドインジェクションと辞書攻撃の連携

本実験では，サイバー攻撃間の連携について実験する。本実験で用いたシナリオファイルを図 4.1 に示す。まずは，OS コマンドインジェクションにより /etc/paswd ファイルを入手する。次に，passwd ファイルに記述されたユーザ名を元に辞書攻撃を実行し，ログイン可能なアカウントを探索する。ログイン可能なアカウントが発見できた場合はログインし，httpd を停止させる。OS コマンドインジェクション，辞書攻撃および SSH によるコマンドの実行は，すべて別のステップである。

実験の結果，httpd のドキュメントルートに index.html ファイルが作成されていることが確認できた。そのため，step:os-injection から step:ssh-list-attack には /etc/passwd ファイルの中身が，step:ssh-list-attack から step:ssh-login にはログイン可能なアカウントの情報が渡されており，攻撃間の連携を確認することができた。

```
scenario:
 - step: os-injection
   target: HTTP-server
   action:
    module: commandInjection
    pass: /injection.php?cmd=<command>
    command: cat /etc/passwd
   failure: SUCCESS

 - step: ssh-list-attack
   target: HTTP-server
   action:
    module: ssh_list
    passwd: True
   failure: SUCCESS

 - step: ssh-login
   target: HTTP-server
   action:
    module: ssh_login
    shell:
    - touch /var/www/html/wordpress/index.html
    - echo "Hacked by DeTMan" > /var/www/html/wordpress/index.html
   failure: SUCCESS
   success: FAILURE
```

図 4.1: OS コマンドインジェクションを発端とした攻撃群のシナリオファイル

```
scenario:
 - step: httpd-stop
   target: HTTP-server
   action:
    module: metasploit
    use: auxiliary/scanner/ssh/ssh_login
    username: test
    password: ttest
    shell:
    - sudo systemctl stop httpd
   failure: SUCCESS
   success: FAILURE
```

図 4.2: Metasploit を用いたシナリオファイル

外部ツール Metasploit を用いた攻撃

多種多様なサイバー攻撃を実行するためのモジュールを，すべて自分で作成することは困難だ。そのため，外部のツールとの連携が重要になる。本実験では，ペネトレーションテスト用のツールである Metasploit を利用可能か実験した。Metasploit を用いて SSH によりログインし，httpd を停止させる。本実験に使用したシナリオファイルを図 4.2 に示す。

実験の結果，httpd が停止していることが確認できた。そのため，DeTMan は外部ツールと連携してサイバー攻撃を行うことができることがわかった。

4.2 既存のサイバー防御演習の再現に関する実験

本節では，前節で用いた 3 つのサイバー攻撃群を用いて既存のサイバー防御演習の再現が可能かどうか実験を行う。以降では WordPress に関する攻撃群を攻撃 1，OS コマンドインジェクションに関する攻撃群を攻撃 2，外部ツールを用いた攻撃群を攻撃 3 とする。また，本節における実験では trainee1 から trainee4 の 4 人の受講者がいると仮定する。trainee1 はすべての攻撃に対策済みであり，trainee2 は攻撃 1，攻撃 2 が対策済み，trainee3 は攻撃 1 のみ対策済み，trainee4 は対策を行っていない。

4.2.1 Hardening の再現

本章では，Hardening における柔軟な演習の進行が再現可能かどうか調査するため，2 つの実験を行う。1 つ目は受講者の能力に合わせた攻撃の選択，2 つ目は

受講者	終了時の攻撃
traiee1	攻撃 3
traiee2	攻撃 3
traiee3	攻撃 2
traiee4	攻撃 1

表 4.1: 受講者の能力に合わせた攻撃の選択に関する実験

trainee1	trainee2		trainee4
攻撃 1	攻撃 3		攻撃 2
攻撃 3	復旧まで待機		復旧まで待機
攻撃 2	攻撃 2		攻撃 2
終了	攻撃 3		復旧まで待機
	復旧まで待機		攻撃 3
	対策		復旧まで待機
	攻撃 1		攻撃 1
	攻撃 3		復旧まで待機
	終了		攻撃 3
			復旧まで待機

表 4.2: 受講者の能力に合わせた攻撃の選択に関する実験

3 つの攻撃群からの攻撃の選択である。

まずは，受講者の能力に合わせた攻撃の選択に関する実験を行う。本実験では，攻撃 1 が最も難易度の低い攻撃，攻撃 3 が最も難易度の高い攻撃であると仮定する。つまり，攻撃 1 の防御に成功した場合は攻撃 2 を実行し，攻撃 1 の防御に失敗した場合は演習を終了する。表 4.1 に，実験結果を示す。表より，受講者が施した対策に応じて実行された攻撃が異なることがわかる。

次に，3 つの攻撃群を 3 つのシナリオファイルに分割して DeTMan を実行した。なお，本実験ではトリガーは用いない。実験結果を表 4.2 に示す。ただし，trainee3 については省略し，trainee4 については一部抜粋にとどめる。表より，防御に成功した攻撃は再度実行されていないことがわかる。また，防御に失敗した攻撃は再度実行されていることもわかる。同時に，防御に失敗した場合は普及するまで次の攻撃の実行を待機していることも確認できる。

この 2 つの実験により，受講者の能力に合わせた攻撃の選択と，受講者に合わせたタイミングで攻撃を実行することが確認できた。

	trainee1	trainee2	trainee3	trainee4
開始～攻撃 1	05:00	05:00	05:00	05:00
攻撃 1	0:00	00:00	00:00	00:04
攻撃 1～攻撃 2	05:00	05:00	05:00	05:00
攻撃 2	0:00	00:00	00:02	00:02
攻撃 2～攻撃 3	05:00	05:00	05:00	05:00
攻撃 3	0:00	00:10	00:09	00:07
合計	15:00	15:10	15:11	15:13

表 4.3: Micro Hardening の再現に関する実験における計測時間

4.2.2 Micro Hardening の再現

本章では，DeTMan が Micro Hardening を再現可能かどうかについて実験を行う。Micro Hardening は時間経過によって攻撃を実行する。そのため，本実験における演習シナリオでは，5 分ごとに攻撃を攻撃 1 から攻撃 3 まで実行する。実験結果を表 4.3 に示す。表より，シナリオ通りの間隔で攻撃が実行されていることがわかる。

4.2.3 クイズ形式によるサイバー防御演習

本章では，クイズ形式によるサイバー防御演習に関する実験を行う。演習シナリオは，まずメッセージを表示する。次に，WEB UI 上のボタンのクリックをトリガーとして攻撃 1 を実行する。攻撃 1 の防御に成功した場合は，利用された脆弱性に関して出題する。防御に失敗した場合は，ヒントを表示し，最初に戻る。演習中の様子を図 4.3 に示す。図より，攻撃 1 の成否に応じたクイズ形式の演習が提供できていることがわかる。

4.3 評価

本節において行ったすべての実験は，シナリオファイルのみが異なる。つまり，DeTMan はモジュールを用意することにより，シナリオファイルに従って様々なサイバー攻撃を自動で実施可能であることがわかる。同時に，DeTMan はシナリオファイルを変更するだけで，様々な演習が実施可能であることがわかった。

DeTMan と Hardening および Micro Hardening との比較結果を 4.4 に示す。DeTMan は，受講者の能力に合わせた攻撃の選択や，受講者に合わせたタイミングでの攻撃の実行を自動で行うことができる。そのため，Hardening には及ばないが柔軟な進行を可能としている。また，DeTMan は Micro Hardening と同様に演習の進行に人の手を必要とせず，同時に 1 人でも演習を開催できる。そのため，演習の開催

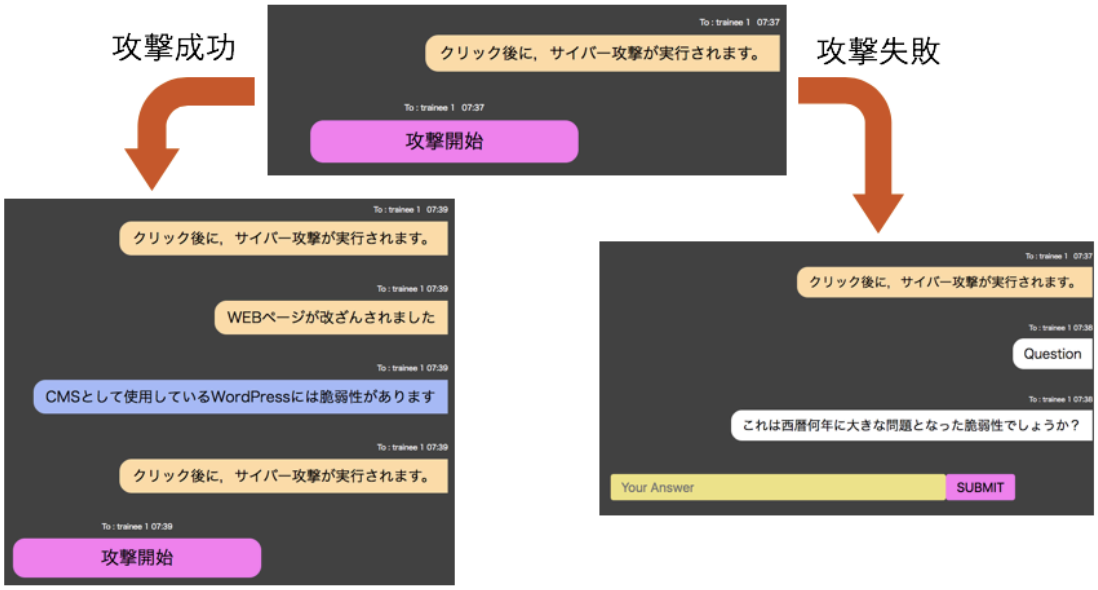

図 4.3: クイズ形式によるサイバー防御演習

	Hardening	Micro Hardening	DeTMan
進行の柔軟さ	◎	×	○
演習進行の負担	×	◎	◎
開催の容易さ	×	◎	◎
学習要素	△	○	◎
演習の用途	訓練	訓練	訓練・学習
演習の自由度	×	×	◎

表 4.4: 既存の演習との比較

や演習の進行に関する負担は Micro Hardening と同様に軽い。他にも，Hardening や Micro Hardening とは異なり，DeTMan は演習の進行に学習要素が組み込まれている。そのため，訓練を目的として開催される Hardening や Micro Hardening に対し，DeTMan は訓練だけでなく，学習に用いることも可能だ。また，学習機能の提供だけでなく，演習の進行に受講者の意思を関与させることが可能だという違いもある。最後に，DeTMan は演習シナリオに従って演習を進行させ，演習シナリオは DeTMan の使用者が自由に作成することが可能だ。そのため，他の演習とは異なり，DeTMan は演習の運営が望む通りの演習を」開催することが可能だ。サイバー防御演習に求められるシナリオは千差万別だが，DeTMan を用いることにより，いつでも，誰でも，意図した通りの演習を実施することが可能になる。これは，セキュリティ教育という点で有用である。

第5章　おわりに

本章では，本論文のまとめと今後の課題と展望について述べる。

5.1　まとめ

本論文では，まず既存のサイバー防御演習を比較し，学習用サイバー防御演習には，柔軟な進行の自動化と学習用機能の追加が必要であることを明らかにした。課題解決のため，学習用サイバー防御演習の進行管理を自動で行うシステムを提案した。本研究では, 柔軟な進行の自動化を，攻撃の成否に応じた進行の分岐と，演習シナリオを複数の攻撃群へと分割し，実行する攻撃群をランダムで選択することにより実現した。また，多種多様な演習に対応するため，拡張性を重視してシステムを設計した。提案システムがサイバー攻撃を自動で実行可能であるか調査した結果，DeTMan は様々なサイバー攻撃を自動で実行することができることが確認できた。また，提案システムの有用性を確かめるため，特定の個人・組織によってのみ運営されている既存のサイバー防御演習を，提案システムによって実施可能か調査した。その結果，提案システムは様々な演習に対応可能であることが明らかとなった。また，様々な実験の際に変更したものは，演習シナリオのみであった点から，提案システムは，演習シナリオを記述するだけで，様々なサイバー防御演習が開催可能であることがわかった。これにより，サイバー防御演習の開催が一層簡単となり，セキュリティ教育という点で有用である。

5.2　今後の課題と展望

5.2.1　トリガーに関する検討

本研究では，主にアクションについて検討した。しかし，より高度なサイバー攻撃の再現と，より柔軟な演習の進行を実現するためにはトリガーが重要になる。ファイアウォールを突破した攻撃を実現するための手法の 1 つとしてリバースシェル型のマルウェアが挙げられる。しかし，現在の DeTMan の構造では，リバースシェル型のマルウェアによる通信をトリガーとして用いることは困難だ。他にもフィッシングサイトに関する演習なども困難である。そのため，DeTMan が対応

可能なサイバー防御演習の幅を広げる必要があると考えている。他にどのようなトリガーが考えられるのかについても検討が必要だ。

5.2.2 CyTrONE との連携に関する検討

現在の DeTMan では，CyTrONE との連携ができていない。そのため，現在は手動で CyRIS を使用してサイバーレンジを作成し，次に DeTMan を動作させるコマンドを入力している。これを CyTrONE が自動で行うようにする必要がある。

謝辞

本研究を行うにあたり、多くの方から多大なご助言やご助力を頂きました。それらの 方々のご協力がなければ、本研究は成り立ちませんでした。心から厚くお礼申し上げま す。

本研究を進めるにあたり、指導教員である Razvan Beuran 特任准教授には様々な助言、適切な御指導を賜りました。心から深く感謝します。また、助言を頂いた副指導教員である篠田陽一教授、副テーマ指導教員である吉高淳夫准教授，西本一志教授に感謝致します。CROND Project の知念賢一特任准教授には、研究に関して活発な議論や多大なご指導を賜りました。心から感謝致します。篠田・知念研究室修了生の村上正樹氏、押川侑樹氏には、研究に関して活発な議論、ご指導を賜りました。また、研究生活を送る上で様々なご助力を頂きました。心から感謝致します。篠田・知念研究室の博士前期課程の橋本光世氏、阿波史和氏、砂川真範氏、淺葉祥吾氏、広瀬太志氏、三島航氏、宮崎駿氏、山口礼央氏、小松源氏には活発な議論や、研究生活を送る上で様々なご助力を頂きました。心から感謝致します。最後に研 究や生活で支えてくれた家族へ心から感謝致します。

参考文献

[1] 警察庁. 平成 29 年中におけるサイバー空間をめぐる脅威の情勢等について. https://www.npa.go.jp/publications/statistics/cybersecurity/data/H29_cyber_jousei.pdf.

[2] 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.

[3] 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.

[4] 一般社団法人 JPCERT コーディネーションセンター. インシデントハンドリングマニュアル. https://www.jpcert.or.jp/csirt_material/files/manual_ver1.0_20151126.pdf.

[5] 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.

[6] Hardening project. http://wasforum.jp/hardening-project/.

[7] 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.

[8] 宮地利幸, 中田潤也, 知念賢一, Razvan Beuran, 三輪信介, 岡田崇, 三角真, 宇多仁, 芳炭将, 丹康雄, 中川晋一, 篠田陽一. Starbed:大規模ネットワーク実証環境. 情報処理, Vol. 49, No. 1, pp. 57–70, January 2008.

[9] 川口 洋. https://microhardening.connpass.com.

[10] Trend Micro. インシデント対応ボードゲーム. https://www.trendmicro.com/ja_jp/about/press-release/2016/pr-20160719-01.html.

[11] Kaspersky. Kaspersky interactive protection simulation. https://www.kaspersky.co.jp/about/press-releases/2018_pro14022018.

[12] JNSA 教育部会. セキュリティ専門家　人狼. https://www.jnsa.org/edu/secgame/secwerewolf/secwerewolf.html.

[13] Moodle 教育管理システム. https://moodle.org.