JAIST Repository
https://dspace.jaist.ac.jp/

Title	状態遷移モデルを活用したサイバー防御演習における振る舞い解析の高度化に関する研究
Author(s)	梅内, 翼
Citation	
Issue Date	2022-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/17626
Rights	
Description	Supervisor: 篠田　陽一, 先端科学技術研究科, 修士（情報科学）

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修士論文

状態遷移モデルを活用したサイバー防御演習における振る舞い解析の高度化に関する研究

梅内 翼

主指導教員 篠田 陽一

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）

令和４年３月

Abstract

In recent years, with the development and spread of information and communication technologies, cyber security measures have become a challenge for many organizations, while a significant shortage of human resources with skills in cyber security is considered to be a problem. In response to this situation, the importance of cyber defense exercises conducted in a virtual space, which is constructed for the purpose of conducting cyber security exercises, has been recognized as a means of acquiring incident response skills. A cyber defense exercise is an exercise aimed at empirically acquiring incident response skills through detection and response to attacks on hosts and systems assigned to the participant, and recovery from failures caused by attacks.

The purpose of the cyber defense exercises conducted on the cyber range is to provide participants with knowledge and skills on incident response through the exercise. To achieve this, it is essential to reflect on what kind of behavior led to a successful defense against the attacks executed in the exercise.

In this study, based on the assumption that the changes brought to the cyber range by the behavior of the participants are considered as differences, we propose a method to systematically extract the differences by integrating the progress of the cyber defense exercise with the state transition model and a method for automatically analyzing the differences that led to the successful defense of the attacks executed in the exercise among the extracted differences.

As a result of our experiments, we verified that it is possible to analyze such differences accurately and in a realistic time. On the other hand, we also found concerns about the coverage of the sources where the differences are extracted, the reproducibility of the behavior, the performance of the analysis algorithm, and the existence of differences that cannot be observed on the cyber range.

目 次

第 1 章　はじめに **1**
1.1　背景 1
1.2　目的 2
1.3　本論文の構成 2

第 2 章　サイバー防御演習の現状と課題 **4**
2.1　サイバー防御演習の概要 4
2.2　サイバーレンジの構築 5
2.3　演習のリアリティ 6
2.4　演習コンテンツ 7
2.5　演習によるスキルの習得 7
2.6　本研究で取り組む課題 8

第 3 章　関連知識・関連事例 **9**
3.1　関連知識 9
3.1.1　コンテナ型仮想化 9
3.1.2　状態遷移モデル 9
3.2　関連事例 10
3.2.1　CyTrONE 10
3.2.2　Hardening 競技会 10
3.2.3　情報危機管理コンテスト 10
3.2.4　CYDER 11

第 4 章　提案手法 **12**
4.1　提案手法の概要 12
4.2　状態遷移モデルの定義 14
4.2.1　状態遷移モデルの概要 14
4.2.2　Attack Phase 14

4.2.3 Recovery Phase 16
4.3 差分の抽出アルゴリズム 16
4.3.1 演習中に防御を実現した差分の抽出 17
4.3.2 潜在的な防御を実現した差分の抽出 17
4.3.3 リカバリを実現した差分の抽出 18
4.4 差分の解析アルゴリズム 18

第 5 章 設計と実装 20
5.1 サイバーレンジのアーキテクチャ 20
5.1.1 サイバーレンジの概要 20
5.1.2 Attacker Host 20
5.1.3 Trainee Host 21
5.1.4 Observer Host 22
5.2 差分の抽出と解析 22
5.3 演習コンテンツの構成 (1) 24
5.4 演習コンテンツの構成 (2) 25

第 6 章 実験・評価 27
6.1 解析環境 27
6.2 実験用データを用いた実験 27
6.2.1 演習コンテンツ A 28
6.2.2 演習コンテンツ B 31
6.3 演習の実施による実験 34
6.3.1 演習コンテンツ A 34
6.3.2 演習コンテンツ B 36
6.4 有効性と解析時間の評価 39

第 7 章 おわりに 41
7.1 本研究の総括 41
7.2 今後の展望 41
7.2.1 差分抽出のために監視する対象の網羅性 41
7.2.2 差分の再現性 42
7.2.3 解析アルゴリズムのパフォーマンス 42
7.2.4 差分として観測できない事項に関する検討 42

図 目 次

2.1 サイバー防御演習の概要図 5

4.1 提案手法の概要図 (差分の抽出) 12
4.2 提案手法の概要図 (差分の解析) 13
4.3 状態遷移モデルの俯瞰図 14
4.4 Attack Phase の状態遷移図 15
4.5 Recovry Phase の状態遷移図 16
4.6 「演習中に防御を実現した差分の抽出」の概要図 17
4.7 「潜在的な防御を実現した差分の抽出」の概要図 17
4.8 「リカバリを実現した差分の抽出」の概要図 18

5.1 サイバーレンジのアーキテクチャ 21

6.1 T_{max} の変化 40

表 目 次

5.1 生成される Dockerfile の例 23

6.1 解析環境 27
6.2 演習コンテンツ A における実験用データを用いた場合の解析結果 (1) 28
6.3 解析結果列の記号の説明 29
6.4 演習コンテンツ A における実験用データを用いた場合の解析結果 (2) 30
6.5 演習コンテンツ B における実験用データを用いた場合の解析結果 (1) 31
6.6 演習コンテンツ B における実験用データを用いた場合の解析結果 (2) 33
6.7 演習コンテンツ A を用いた演習実施時のログ 35
6.8 演習コンテンツ A における演習結果を用いた場合の解析結果 35
6.9 演習コンテンツ B を用いた演習実施時のログ 36
6.10 演習コンテンツ B における演習結果を用いた場合の解析結果 38
6.11 計算式中の変数の説明 39

第1章　はじめに

本章では本研究の背景と目的，本論文の構成について述べる．

1.1　背景

近年，情報通信技術の発展とその普及に伴い，サイバーセキュリティ対策が多くの組織にとって課題となっている一方で，サイバーセキュリティに関するスキルを有する人材が大幅に不足していることが問題視されている．総務省による 2016 年時点の試算では，2020 年の時点においてセキュリティ人材が約 19.3 万人不足することが指摘されている [1]．そのような状況を受け，インシデントレスポンスに関するスキルを習得させるための手段として，サイバーセキュリティに関する演習の実施を目的として構築される仮想空間 (以下，サイバーレンジ) 上で実施されるサイバー防御演習の重要性が認知されている．サイバー防御演習とは，受講者が自身に割り当てられたホストやシステムに対して実行される攻撃の検知や対応，攻撃によって発生した障害に対する復旧を通して，インシデントレスポンスのためのスキルを経験的に習得させることを狙った演習のことである．

サイバーレンジおよびサイバー防御演習については，複数の観点において既存研究が存在する．文献 [2,3] では，サイバーレンジを構築するためのアーキテクチャに着目し，仮想マシンやコンテナ型仮想化技術を用いたサイバーレンジの構築手法について検討が行われている．また，文献 [4–6] では，サイバーレンジ上で実施される演習の進行管理に焦点が当てられており，受講者の進行状況に応じてイベントの発生を制御する機構や，特定のポイントまで演習をロールバックする機構を実装することで，受講者のレベルに応じた演習や反復学習の機会を提供する試みがなされている．これら以外にも，演習コンテンツの自動生成に関する研究 [7] や演習コンテンツと LMS(Learning Management System) を同期させることで学習効果の向上を狙った研究 [8,10] が存在する．そして，文献 [14] では，上記で示したように広範に渡るサイバー防御演習およびサイバーレンジの研究を複数の観

点から包括的にサーベイした結果が示されている．

さて，サイバー防御演習においてインシデントレスポンスのためのスキルを経験的に習得するためには，一般的な学習と同様に，演習後にその内容を振り返り，復習するプロセスが必要不可欠である．そのようなプロセスが用意されていないと，受講者は当該演習の実施を通して何を学んだのか，あるいは，どうすれば攻撃に対して適切に対応することができたのかといった事項について十分理解できないまま演習を終えてしまうことが懸念される．既存研究 [10] では，サイバーレンジ上で得た情報をもとに事前に作成された問題に回答する手法が提案されているが，当該手法には 2.5 に挙げる課題が存在する．

1.2 目的

本研究では，サイバー防御演習を通したスキルの習得を促進させるために，演習後の振り返りにおいて有用な情報を提供することを目的とする．この目的を達成するために，受講者の振る舞いによってサイバーレンジにもたらされた変化を差分として捉えた上で，サイバー防御演習の進行状況を状態遷移モデルと統合することで差分を体系的に抽出する手法，および，抽出した差分のうち，演習内で実行される攻撃への対策を成功に導いた差分を自動的に解析する手法を提案する．提案手法により，サイバーレンジにおける受講者の振る舞いを効果的に抽出，解析可能になり，解析結果を用いた演習後の振り返りの活性化が期待される．

1.3 本論文の構成

本論文の構成を以下に示す．

第 2 章では，サイバーレンジやサイバー防御演習に関する既存研究を示すとともに，本研究にて取り組む課題を明らかにする．

第 3 章では，本論文で述べる提案手法やその実装において利用する関連技術および関連知識について説明する．

第 4 章では，状態遷移モデルを活用したサイバー防御演習における振る舞い解析の高度化を実現するための提案手法について述べる．

第 5 章では，提案手法を実現するためのサイバーレンジのアーキテクチャや差分の抽出および解析の実施手法について述べる．加えて，第 6 章の実験および評

価で利用する演習コンテンツの構成を提示する．

第 6 章では，実験によって本提案手法の効果を検証する．検証においては，実験用データおよび実際の演習の実施を通して収集した受講者の振る舞いデータから有効な振る舞いを解析することが可能か検証する．あわせて，解析に要する時間についても検討する．そして，実験の結果明らかになった課題について議論する．

第 7 章では，本研究を総括するとともに今後の展望について述べる．

第2章　サイバー防御演習の現状と課題

本章では，まずサイバー防御演習の概要について述べる．次に，複数の観点におけるサイバー防御演習の現状と課題について概述する．最後に，本研究が対象とする課題を明らかにする．

2.1　サイバー防御演習の概要

サイバー防御演習とは，受講者が自身に割り当てられたホストやシステムに対して実行される攻撃の検知や対応，攻撃によって発生した障害に対する復旧を通して，インシデントレスポンスのためのスキルを経験的に習得させることを狙った演習のことである．インシデントレスポンスおよびサイバーセキュリティに関するスキルの概念は Workforce Framework for Cybersecurity (NICE Framework) [11] において詳細に定義されているが，本稿におけるスキルとは，発生したインシデントの検知や即時的な対応，根本的な原因の解決，システムの復旧を適切かつ迅速に実施する能力のことを指すものとする．

一般的に，サイバー防御演習は受講者用のネットワークと主催者用のネットワークが分離されたサイバーレンジ上で実施される．演習において，受講者は，関係者役の主催者と連絡を取りながら，受講者用のネットワーク内に構築された各種サーバやシステムにおいて発生するインシデントに適切に対応することが求められる．また，主催者は前述の関係者役と攻撃役に二分されることが多い．このうち，関係者役は主催者の上司やインシデントが発生するシステムのステークホルダーとして振る舞う．一方の攻撃役は主催者用ネットワーク内に構築された C2 (Command & Control) サーバや攻撃用ホストを駆使して受講者用ネットワーク内に構築されたサーバやシステムに対して攻撃を実施する．なお，攻撃の実施度合いは，受講者の対応状況を監視した結果に基づいて攻撃役側の裁量によって調整されることがある．これらは一般論であり，攻撃用の仕組みが自動化されている場合や関係

者役が用意されていない場合も存在する．図 2.1 に，サイバー防御演習の概要図を示す．

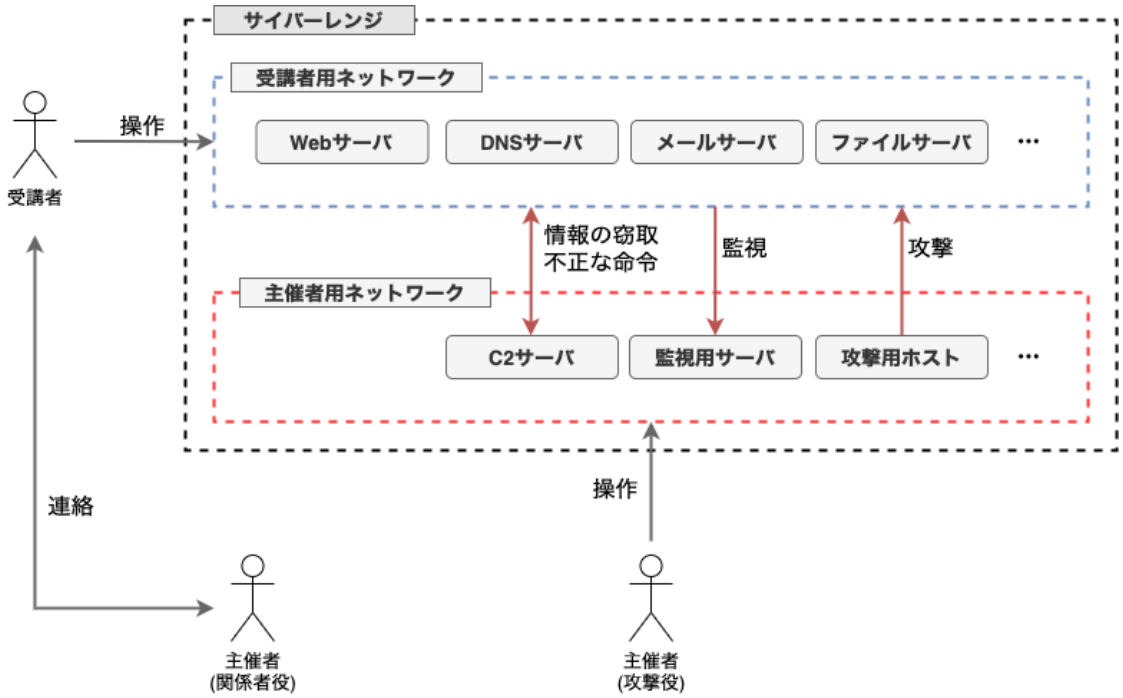

図 2.1: サイバー防御演習の概要図

サイバー防御演習の具体例としては，Hardening Project [12] によって実施されている Hardening 競技会や，国立研究開発法人 情報通信研究機構 (NICT) [13] のナショナルサイバートレーニングセンターによって実施されている CYDER 等が存在する．これらの関連する取り組みについては 3.2 にて詳しく述べる．

2.2 サイバーレンジの構築

サイバーレンジの規模や構成は，そこで実施される演習の内容や難易度，実施形態によって大きく異なる．たとえば,「Web アプリケーションに存在する脆弱性の修正」を題材とする場合，サイバーレンジ内には単一の Web サーバと当該サーバ上で稼働させる Web アプリケーションのプログラムを用意するだけで良いが，他方で「企業を対象とした標的型メール攻撃によるマルウェア感染に起因するインシデントの対応と封じ込め」を題材とする場合，企業を模したネットワーク環境やそこに接続された大量のホスト，標的型メールの送信機構といった複数の構成要素を用意する必要が生じる．そのため，サイバーレンジの構築手法には，様々

なニーズに沿ったサイバーレンジを一元的な仕組みで構築可能であることという要件が求められる．この要件を受け，文献 [2] では，YAML を用いて様々なニーズに沿ったサイバーレンジを構築するためのフレームワークが提案されている．この他にも，コンテナ型仮想化技術を活用する手法 [3] やファイルシステムのクローンを活用する手法 [15]，クラウドサービスおよびソフトウェア VPN を活用する手法 [16] が提案されている．

このうち，本研究では文献 [3] と同様にコンテナ型仮想化技術を活用してサイバーレンジを構築する手法を採用している．ここで，文献 [3] では物理リソースに乏しい環境でも導入が容易であり，かつ，高い移植性を有するというコンテナ型仮想化技術の特長を多様な組織に対するサイバー防御演習の導入に活用しているが，本研究では，宣言的に仮想環境を定義，展開することが可能であるというコンテナ型仮想化技術の特長をサイバー防御演習における受講者の振る舞い解析の自動化に活用している点が大きく異なる．

2.3 演習のリアリティ

可能な限り現実に近い環境を提供するために，サイバーレンジには高いリアリティが求められる．たとえば，受講者に割り当てられたホストやシステムを監視する機構は現実の環境には存在しないため，可能な限り受講者から隠蔽されていることが好ましい．また，リアルタイムなログ解析やトラフィック解析を行うような演習では，不審なログやトラフィックだけでなく，正常なものも生成，転送されるように準備する必要がある．これらの要件を受け，サイバーレンジにおいてサーバに対するアクセスやメールの送信をダイナミックに行う機構 [4]，監視用プロセスの偽装やランダムなトラフィックの生成，転送を行う機構 [17] が提案されている．

このうち，本研究では文献 [4] で言及されていた「参加者が意識しない方法での監視/観測方法」に着目した．具体的には，サイバー防御演習の実施中に演習の進捗状況や各種のログを集約するホストを分離し，可能な限り受講者から演習とは本質的に関係しない機能が隠匿されるようにした．なお，サイバーレンジの具体的な構成については 5.1 で詳細に述べる．

2.4 演習コンテンツ

演習の実施を通して現実的なインシデントレスポンスのスキルを育成するため，演習内で実施される攻撃は現実において発生が確認されている攻撃手法に沿ったものであることが求められる．しかしながら，実施される演習の内容は，当該演習の作成者が有する知識やスキルに制限されるという課題が存在する．この要件を受け，文献 [2,17] では，STIX 等の脅威インテリジェンスをベースとした演習の生成について言及されている．

本研究では，演習コンテンツの自動構成についてはスコープ外としているが，現実的な演習コンテンツの作成を容易にするための仕組みを構築している．具体的には，Cyber Kill Chain [18] 等の攻撃が複数のステップで表されるモデルをベースとした演習コンテンツの作成が可能となるように後述する状態遷移モデルを定義している．なお，Cyber Kill Chain とサイバー防御演習の統合については文献 [2] にて検討している．

2.5 演習によるスキルの習得

サイバー防御演習の実施の本来の目的は，当該演習の実施を通して受講者にインシデントレスポンスに関する知識やスキルを習得させることが目的である．そのため，単に演習を実施するだけではなく，演習で習得することが想定されている知識やスキルが実際に習得されているかを確認する仕組みや，その確認を行いやすくする工夫が必要である．この要件を受け，文献 [10] では，演習終了後に LMS を用いて演習に関する問題に回答させる仕組みが提案されている．また，文献 [6] では，反復学習を可能にするために演習を特定の時点まで巻き戻す機構が提案されている．このような仕組みを用いて演習を複数回行ったり，それに基づいて問題を出題したとしても，以下に挙げる問題点が存在するため，スキルの習得という観点においては不十分である．

- 問題に正答できたからといって，演習中に攻撃に対して適切な対処を行ったとは限らない．
- 問題で確認できる内容は，演習内容における課題と同様に演習および問題の作成者が有する知識やスキルに制限される．

- より厳密に振り返りをするためには問題数や巻き戻しの回数を増やすか無いが，これは受講者と演習作成者の双方にとって負担である．

2.6 本研究で取り組む課題

本研究では，サイバーセキュリティ人材の育成が急務であるという実社会の要請を踏まえ，サイバー防御演習の最大の目的を「演習によるスキルの習得」と位置づける．そこで，上記に挙げた課題のうち 2.5 で提示した「演習を通したスキルの習得における課題」の解決を目的としたサイバー防御演習における受講者の振る舞い解析の手法，および，当該手法を実現するためのサイバー防御演習における状態遷移のモデリング手法やサイバーレンジの構築手法について提案する．

第3章　関連知識・関連事例

本章では，本論文で述べる提案手法やその実装において利用する関連知識およびサイバー防御演習に関連する事例について説明する．

3.1　関連知識

3.1.1　コンテナ型仮想化

コンテナ型仮想化とは，OS レベルでコンピュータシステムのリソースを仮想化することで，ハードウェアを仮想化する VM(Virtual Machine) 型仮想化と比較して軽量なアプリケーションを実行するための環境を提供する仕組みを指す [19]．コンテナ型仮想化には，過去に FreeBSD Jail や OpenVZ 等の多数の実装が存在したが，現在は Linux の `pivot_root` システムコールや Namespace，`cgropus` 等の仕組みを活用してコンテナ型仮想化を実現する Docker [20] が標準的に利用されている．

本研究においては，提案手法を実現するために構築するサイバーレンジ内のホストを構築するために Docker コンテナを利用する．あわせて，サイバーレンジにおいて攻撃を行う際に利用するコンテナ間の通信は Docker のネットワーク機能を利用する．

3.1.2　状態遷移モデル

状態遷移モデルとは，あるシステムの状態およびそのシステムにおいて発生するイベントに応じた状態の変遷をモデル化したものである．代表的な状態遷移モデルとして UML のステートマシン図 (State Machine Diagram) が存在する．ステートマシン図では，あるシステムの状態を角丸矩形を用いて，また，遷移を矢印を用いて表現する．本研究においては，サイバー防御演習の進行状況の把握や進行状況に応じた受講者の振る舞いの抽出を可能にする，独自に定義した状態遷移モデルを利用する．本研究で利用する状態遷移モデルの定義は 4.2 において示す．

3.2 関連事例

3.2.1 CyTrONE

CyTrONE [2] は，北陸先端科学技術大学院大学の Cyber Range Organization and Design (CROND) チームにより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 の特徴として，CyRIS [8] や CyPROM [9] といったツールと連携することで，演習環境の構築や LMS による演習コンテンツによる学習のための仕組みを統合的に管理することができる点が挙げられる．

3.2.2 Hardening 競技会

Hardening 競技会は，Hardening Project [12] により 2012 年より実施されているセキュリティ堅牢化の競技会である．本競技会の特徴として，一般的な CTF(Capture The Flag) 系の競技会で競われる脆弱性のエクスプロイトに関するスキルではなく，脆弱性の修正やインシデントレスポンス，サイトの可用性の維持，関係者との適切なコミュニケーションといったシステムの堅牢化に関する総合的なスキルを競う点が挙げられる．競技において，参加者には脆弱性が存在する EC サイトの Web アプリケーションが割り与えられる．参加者は，当該 EC サイトに存在する脆弱性を修正しながら，随時発生するインシデントに対して適切に対応することが求められる．なお，参加者に割り与えられるシステムは StarBED [21] 内に構築されたサイバーレンジ上で展開されている．

競技は 2 日間に渡って開催される．このうち，初日は堅牢化の競技を行う Hardening Day であり，2 日目は競技に関する振り返りを行う Softning Day である．この Softning Day において他参加者と共同で競技に関する振り返りを行うことで，競技においてより良いパフォーマンスを発揮できなかったチームにおいてもシステムの堅牢化に関するスキルを学ぶことができる点も大きな特徴である．

3.2.3 情報危機管理コンテスト

情報危機管理コンテスト [22] は，サイバー犯罪に関する白浜シンポジウム実行委員会が主催するインシデントレスポンスのスキルを競う競技会である．参加者は顧客企業のサーバの管理者として，競技中に発生するインシデントに対して技術的な対応を行ったり，関係者に対する電話対応を行ったりする点が特徴である．

なお，コンテストの運営者は競技実施中の参加者がインシデントレスポンスの中で実施した対応や発言をサイバーレンジ内に展開されたシステムのログや通話のログを随時収集している．当該コンテストにおいては，競技実施後に参加者の評価を行うためにこれらのログを収集しているが，本研究においては，このようなログをもとにインシデントレスポンスにおける受講者の有効な振る舞いを解析する点が大きく異なる．

3.2.4 CYDER

CYDER(Cyber Defense Exercise with Recurrence) [23] は，国立研究開発法人情報通信研究機構 (NICT) [13] のナショナルサイバートレーニングセンターが国の機関や地方公共団体，重要社会基盤事業者等を対象に実施している体験型のサイバー防御演習である．演習において，受講者は組織のネットワーク模した大規模仮想 LAN 環境下で実機の操作を伴う一連のインシデントレスポンスの手法を実践する．なお，Hardening 競技会と同様に，参加者に割り与えられる仮想 LAN 環境は StarBED [21] 内に構築されている．

演習は 1 日で実施されるが，演習前にオンライン教材を用いた事前学習を行うことで学習効果を高めている点が演習によるスキルの習得の観点における特徴である．

第4章　提案手法

本章では，状態遷移モデルを活用したサイバー防御演習における受講者の振る舞い解析の高度化を実現するための提案手法について述べる．

4.1　提案手法の概要

前提として，サイバー防御演習において受講者は自身に割り当てられた環境に対して実施される攻撃を遮断するために様々な振る舞いをする．たとえば，既存の脆弱性が確認されているバージョンのソフトウェアがインストールされている場合は当該ソフトウェアをアップデートしたり，脆弱な設定が行われているサービスに対してコンフィグレーションの修正を行ったりする．その結果，受講者に割り当てられた環境には何らかの差異が生じる．

本研究の提案手法は，受講者の振る舞いによってサイバーレンジにもたらされた変化を差分として捉えた上で，サイバー防御演習の進行を状態遷移モデルと統合することで差分を体系的に抽出する手法，および，抽出した差分のうち，演習内で実行される攻撃の防御を成功に導いた差分を自動的に解析する手法の 2 つである．

まず，差分の抽出について述べる．ある受講者が特定の攻撃の防御に成功した場合，受講者は当該攻撃が実施される前に自身のホストに対して何らかの差分を生じさせており，その差分のいずれかの組み合わせが防御を成功に導いていると考えられる．そこで，ある攻撃の防御を成功に導いた差分を解析するために，当該攻撃が実施されるより前に生成された

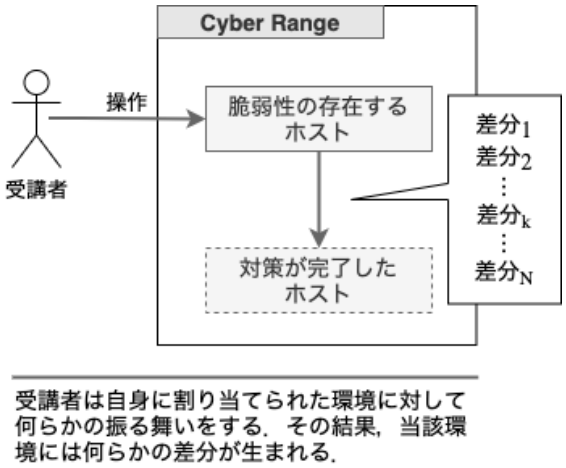

図 4.1: 提案手法の概要図 (差分の抽出)

差分をすべて抽出する．図 4.1 にその
概要図を示す.

次に，差分の解析について述べる．
上記の手順において抽出した差分の中には，攻撃の防御に関連しない差分も含まれている．そこで，本手順では抽出した差分全体から，防御を成功に導くために必要十分な差分のみを抽出することを目的とする．この目的を実現するために，まず，差分全体から順番を保ったまま抽出可能な長さ 1 以上のすべての列を抽出する．次に，抽出した列の数に対応するサイバーレンジのインスタンスを生成した後，各インスタンスに対して列に含まれる差分を事前に適用する．そして，抽出した差分に対応する攻撃を実行し，当該攻撃の防御が成功するか検証する．最後に，防御に成功したもののうち，最も長さの短いものを結果として出力する．図 4.2 にその概要図を示す.

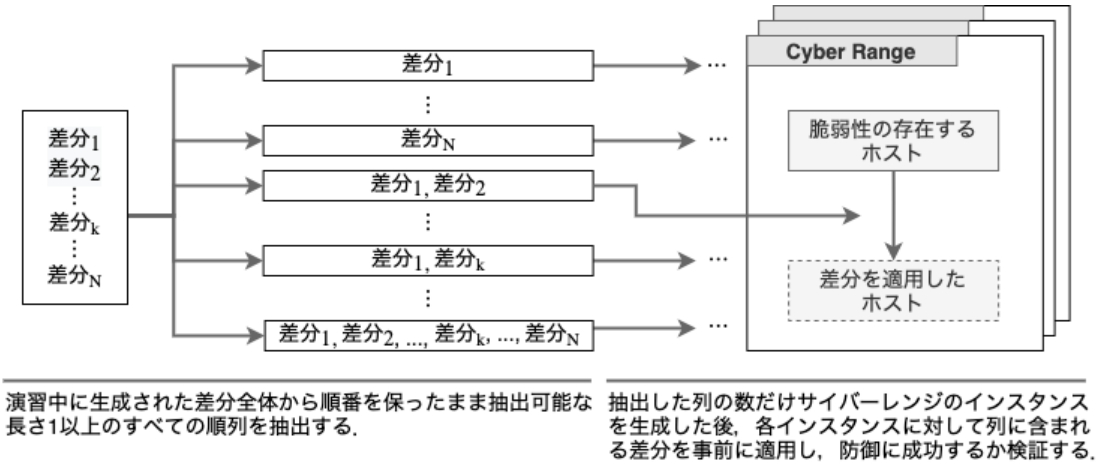

図 4.2: 提案手法の概要図 (差分の解析)

4.2 状態遷移モデルの定義

本章では，状態遷移モデルの定義について述べる．

4.2.1 状態遷移モデルの概要

本稿において提案する状態遷移モデルは，複数の Attack Phase と単一の Recovery Phase のリストから構成される．図 4.3 に，状態遷移モデルの俯瞰図を示す．

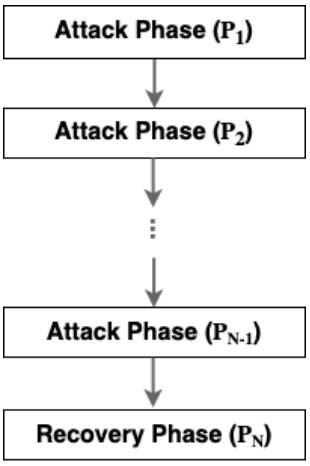

図 4.3: 状態遷移モデルの俯瞰図

Attack Phase は，演習において実行される攻撃を構成する特定のステップに対応する Phase である．また，Recovery Phase は，演習において実行される攻撃の防御に失敗し，その攻撃の最終目的 (たとえば，ファイルの改ざんやサービスの停止等) が達成された後に，攻撃の封じ込めや攻撃によって発生した障害の対処を行うプロセスに対応する Phase である．ここで，各 Phase は実行される順に P_1，P_2，...，P_N と表現する．

また，各 Phase には複数の State および Transition が含まれる．ここで，各 State は Phase と同様に S_1，S_2，...，S_e と表現する．また，各 Transition も同様に T_1，T_2，... と表現する．さらに，Phase と State の組み合わせを Condition と呼び，$(P_x,\ S_y)$ と表現する．たとえば，ある時点における状態が Phase P_2 内の State S_3 である場合は $(P_2,\ S_3)$ と表現する．なお，演習はある Phase の S_e に到達した時点で終了するものとする．

4.2.2 Attack Phase

Attack Phase は 4 つの State と 6 つの Transition から構成される．Attack Phase の状態遷移図を図 4.4 に示す．また，それぞれの State および Transition の意味を以下に示す．なお，下記の説明における可用性検査とは，受講者に割り当てられ

たホスト上で稼働しているサービスが正常に動作していることを検証するプロセスを指す.

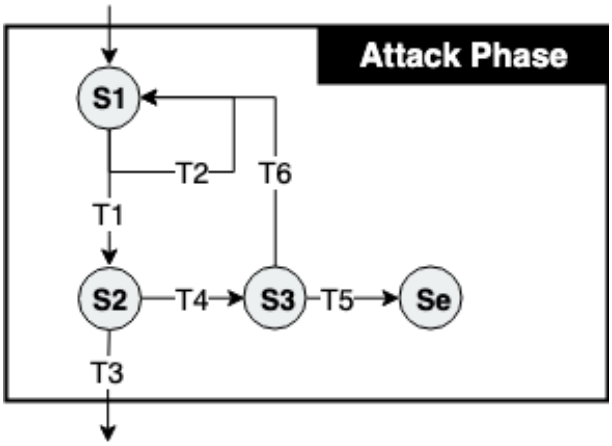

図 4.4: Attack Phase の状態遷移図

State

S_1 攻撃実行前の可用性検査に成功していない状態.

S_2 可用性検査に成功し，攻撃の実行を待っている状態.

S_3 攻撃の防御に成功し，攻撃終了後の可用性検査の実行を待っている状態.

S_e 攻撃の防御に成功し，かつ，攻撃終了後の可用性検査の実行に成功した状態.

Transition

T_1 攻撃実行前の可用性検査に成功した.

T_2 攻撃実行前の可用性検査に失敗した.

T_3 攻撃の防御に成功した.

T_4 攻撃の防御に失敗した.

T_5 攻撃実行後の可用性検査に成功した.

T_6 攻撃実行後の可用性検査に失敗した.

4.2.3 Recovery Phase

Recovery Phase は 2 つの State と 2 つの Transition から構成される．Recovery Phase の状態遷移図を図 4.5 に示す．また，それぞれの State および Transition の意味を以下に示す．なお，下記の説明におけるリカバリ検査とは，攻撃の結果として受講者に割り当てられた環境内に作成されたバックドアの削除や停止されたサービスの復旧が適切に実施されていることを検証するプロセスを指す．

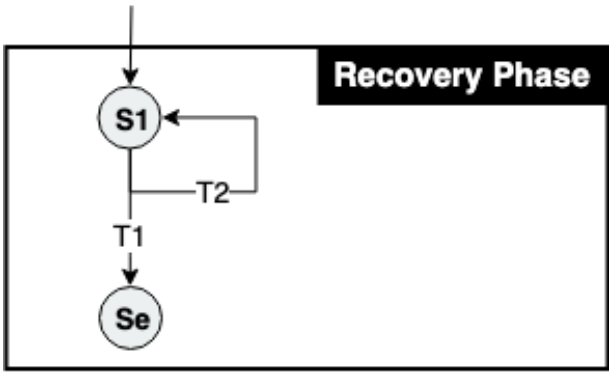

図 4.5: Recovry Phase の状態遷移図

State

S_1 リカバリ検査に成功していない状態.

S_e リカバリ検査に成功した状態.

Transition

T_1 リカバリ検査に成功した.

T_2 リカバリ検査に失敗した.

4.3 差分の抽出アルゴリズム

本節では，演習終了後の Condition ごとに，その結果から導出可能な事実を提示する．あわせて，解析対象とする差分を抽出するためのアルゴリズムを示す．

4.3.1 演習中に防御を実現した差分の抽出

演習終了後の Condition が $(P_x,\ S_e)_{x \neq N}$ だった場合，受講者が Phase P_x で実行される攻撃の防御に成功していると言える．この場合，$(P_1,\ S_1)$ から $(P_x,\ S_2)$ の間に受講者が生成した何らかの差分の列が当該攻撃の防御を成功に導いている．そこで，上記の期間に生成された差分を抽出し，解析対象とする．本項の抽出対象を図示したものを図 4.6 に示す．

図 4.6: 「演習中に防御を実現した差分の抽出」の概要図

4.3.2 潜在的な防御を実現した差分の抽出

演習終了後の Condition が $(P_x,\ S_e)$ だった場合，$1 \leq y < x$ の制約を満たすすべての y について，Condition を $(P_y,\ S_1)$ に設定した上で演習終了直後の状態のホストに対して Phase P_y の攻撃を実行し，当該攻撃の防御に成功する Phase が存在するかどうかを検証する．もし，そのような Phase P_z が存在した場合，$(P_{z+1},\ S_1)$ から $(P_x,\ S_e)$ の間に受講者が生成した何らかの差分の列が当該攻撃の防御を潜在的に成功させている．そこで，上記の期間に生成された差分を抽出し，解析対象とする．本項の抽出対象を図示したものを図 4.7 に示す．

図 4.7: 「潜在的な防御を実現した差分の抽出」の概要図

4.3.3 リカバリを実現した差分の抽出

演習終了後の Condition が $(P_N,\ S_e)$ だった場合，すべての攻撃の防御に失敗しているが，最終的なリカバリには成功していると言える．この場合，$(P_N,\ S_1)$ から $(P_N,\ S_e)$ の間に受講者が生成した何らかの差分の列がリカバリを実現させている．そこで，上記の期間に生成された差分を抽出し，解析対象とする．本項の抽出対象を図示したものを図 4.8 に示す.

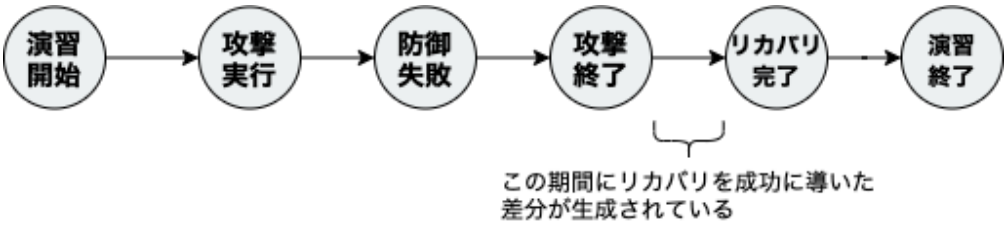

図 4.8: 「リカバリを実現した差分の抽出」の概要図

4.4 差分の解析アルゴリズム

本節では，前述の解析対象として抽出した差分をもとに，ある攻撃の防御を成功に導く，あるいは，リカバリを実現するために必要十分な差分 (以下，有効な差分) を解析する手法を示す.

まず，受講者によって特定の期間に生成された差分全体のリストを $D = [d_1,\ d_2,\ ...,\ d_M]$ とする．ここで，D に含まれる差分は，当該差分が生成されたタイムスタンプ順にソートされているものとする.

次に，D から順番を保ったまま抽出可能な長さ 1 以上の列を要素とする集合を D' とする．すなわち，D' は以下に示すような集合である.

$$
\begin{aligned}
D' = \{ & \\
& [d_1],\ [d_2],\ ...,\ [d_M], \\
& [d_1,\ d_2],\ ...,\ [d_x,\ d_y],\ ...,\ [d_{M-1},\ d_M], \\
& ..., \\
& [d_1,\ d_2,\ ...,\ d_{M-1},\ d_M] \\
\} &
\end{aligned}
$$

次に，任意の D' の要素 d について，d ごとにサイバーレンジのインスタンスを生成した後，各インスタンスに対して d に含まれる差分を順序を保存したまま適用する，そして，抽出した差分に対応する攻撃，あるいは，リカバリ検査を実行する．なお，攻撃やリカバリ検査の実行にあたっては，それまでの Phase で実行された攻撃が成功した場合の状態を事前に再現しておくものとする．また，効率良く必要十分な差分を探索するため，上記の手続きは要素数の少ない d から順に実施する．このとき，攻撃の防御，あるいは，リカバリ検査を成功に導く要素数が最も少ない d を有効な差分として出力する．

第5章　設計と実装

本章では，前章の提案手法を実現するためのサイバーレンジのアーキテクチャや差分の抽出および解析の実施手法について述べる．加えて，次章の実験で利用する演習コンテンツの構成を複数提示する．

5.1　サイバーレンジのアーキテクチャ

5.1.1　サイバーレンジの概要

本研究においては，本提案手法を実現するためのサイバーレンジを独自に実装した．そのアーキテクチャを図 5.1 に示す．

図 5.1 に示す通り，サイバーレンジは Attacker Host，Trainee Host，Observer Host の 3 種のホストとそれらに含まれる各種のサービス，エージェント，コンフィグレーション等の要素から構成される．

5.1.2　Attacker Host

Attacker Host は，Trainee Host において稼働しているサービスに対して攻撃や可用性検査を実行するための単一のホストである．Attacker Host に含まれる要素の概要を以下に示す．

Script Trainee Host において稼働しているサービスに対して実行する攻撃が実装されたスクリプト群．あるスクリプトが特定の Attack Phase において実行される攻撃を実装している．すなわち，演習に含まれる Phase の数が N 個であるとき，Script の数は全体の Phase 数から Recovery Phase の分を除いた $N-1$ 個存在する．

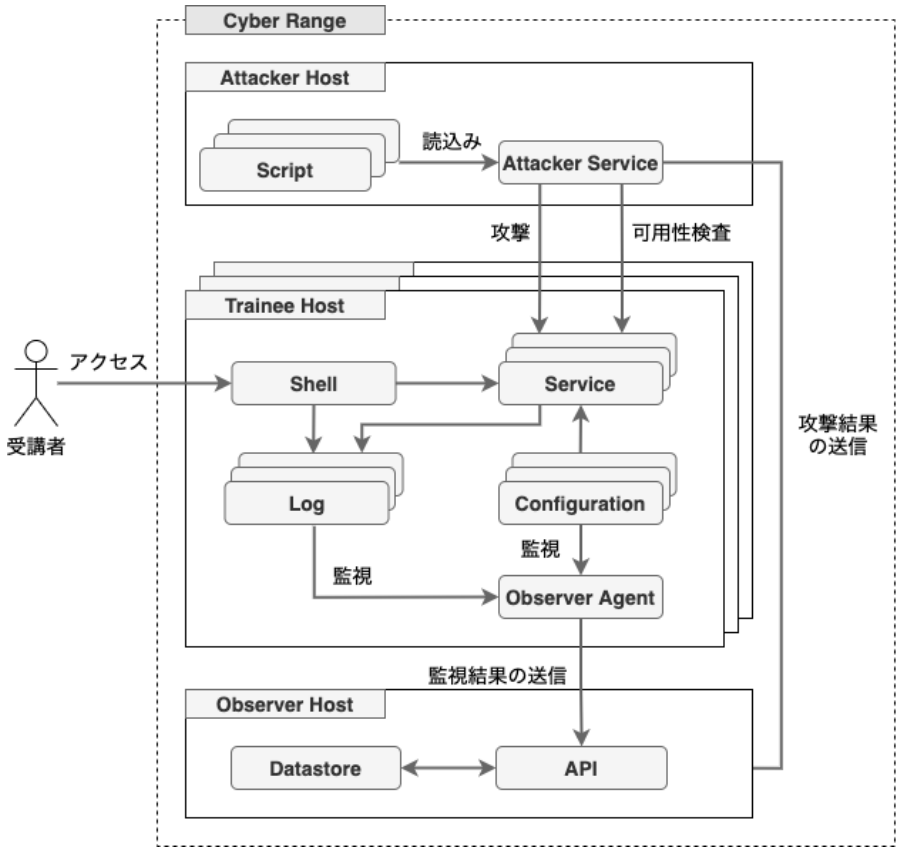

図 5.1: サイバーレンジのアーキテクチャ

Attacker Service Trainee Host において稼働しているサービスに対して攻撃や可用性検査を実行するサービス．前述の機能に加えて，実行した攻撃や可用性検査の結果，発火した Transition を Observer Host に送信する機能を持つ．

5.1.3 Trainee Host

Trainee Host は，受講者に割り当てられる単一あるいは複数のホストである．当該ホスト上では，1 つ以上のサービスが稼働している．Trainee Host に含まれる要素の概要を以下に示す．

Service Trainee Host において提供されているサービス．Attacker Agent によって実行される攻撃は，当該サービスの実装に存在する脆弱性やサービスの不適切なコンフィグレーションに起因する不備を悪用する．

Shell 受講生が Trainee Host にアクセスするためのシェル．

Log Service や Shell によって生成されたログデータ群．ログデータは生成元となるサービスによってフォーマットが異なるため，後述する Observer Agent において適宜整形する．

Configuration Service のコンフィグレーションを示したファイル．

Observer Agent Service や Shell から生成されたログデータを収集，整形し，Observer Host に送信するエージェント．

5.1.4 Observer Host

Observer Host は，演習実施中に収集されたログを集約するための単一のホストである．Observer Host に含まれる要素の概要を以下に示す．

API Attacker Service や Observer Agent から送信されるデータを受信し，Datastore への永続化を行うモジュール．

Datastore API 経由で受信したデータの永続化を行うデータストア．振る舞いの解析は，演習終了後にこのデータストアから抽出したデータをもとに実施する．

5.2 差分の抽出と解析

本研究では，受講者による振る舞いの結果生成される差分として，コマンドの実行履歴と特定のファイルの変更履歴を対象とした．このうち，コマンドの実行履歴は，Trainee Host 内の Shell のヒストリーを記録したファイルの内容を定期的に観測することで収集する．ただし，ホストに影響を及ぼさないようなコマンド (`ls` や `cwd` 等) は観測対象から除外する．また，ファイルの変更履歴については，Trainee Host 内の Configuration や攻撃の過程において参照されるファイルを監視対象とする．

抽出した差分の解析は，前述の通りサイバーレンジのインスタンスを複数生成し，事前に Trainee Host に各差分を適用した上で攻撃やリカバリ検査を実行することで実現する．

なお，差分の適用手法は，解析対象の差分が Attack Phase および Recovery Phase のどちらにおいて生成されたかによって異なる．

まず，解析対象の差分が Attack Phase において生成された場合，Trainee Host 用のコンテナイメージをビルドするための Dockerfile において `RUN` コマンドを用いて差分が適用された状態を作り出すことで実現する．たとえば，抽出した差分が d_1 および d_2 の 2 つであり，それぞれの差分を適用するためのコマンドが c_1 および c_2 の場合，表 5.1 に示す 3 種類の Dockerfile を用いた Trainee Host のコンテナイメージが生成される．

表 5.1: 生成される Dockerfile の例

差分 d_1 を適用した
コンテナイメージを生成する
Dockerfile
	`FROM ubuntu:focal`
	`...`
	`RUN apt install`
	`...`
	`RUN` c_1
	`CMD ["..."]`

差分 d_2 を適用した
コンテナイメージを生成する
Dockerfile
	`FROM ubuntu:focal`
	`...`
	`RUN apt install`
	`...`
	`RUN` c_2
	`CMD ["..."]`

差分 d_1, d_2 を適用した
コンテナイメージを生成する
Dockerfile
	`FROM ubuntu:focal`
	`...`
	`RUN apt install`
	`...`
	`RUN` c_1 `&&` c_2
	`CMD ["..."]`

一方で，解析対象の差分が Recovery Phase において生成された場合は，事前に Attack Phase で実行されるすべての攻撃を実行した上で，対象の差分を適用するためのコマンドを含んだシェルスクリプトを実行することで実現する．

5.3 演習コンテンツの構成 (1)

1 つ目の演習コンテンツ (以下，演習コンテンツ A) として「OS Command Injection 脆弱性が存在するロジックが実装された Web アプリケーションに対するリバースシェルの設置とサービス拒否攻撃の実施」を取り上げ，作成した．演習コンテンツ A の具体的なシナリオの流れを以下に示す.

P_1 Attacker Service は Trainee Host に対してポートスキャナを用いたポートスキャンを行う．ここで，ポート番号 5000 において HTTP サービスが提供されていることを確認できた場合，次の Phase に進む.

P_2 Attacker Service は OS Command Injection 脆弱性の存在有無を検証するため，Trainee Host に対してシェルのコマンドとして解釈される文字列をクエリに含んだ HTTP リクエストを送信する．ここで，当該リクエストに対応するレスポンスにコマンドの実行結果である文字列が含まれていた場合，次の Phase に進む.

P_3 Attacker Service は OS Command Injection 脆弱性を用いて Trainee Host 上にバックドアプロセスを生成するためのコマンドを含んだ HTTP リクエストを送信する．ここで，生成したバックドアプロセスを介して Trainee Host において root 権限でのコマンドの実行が可能であることを確認できた場合，次の Phase に進む.

P_4 Attacker Service は P_3 で生成したバックドアプロセスを介して以下の情報を取得するためのコマンドを実行する．すべてのコマンドの実行が完了し，目的とする情報が取得できた時点で次の Phase に進む.

- Trainee Host のハードウェア情報.
- Trainee Host 上で動作している OS のバージョン.
- Trainee Host のファイルシステム上に存在するアカウント情報.

P_5 Attacker Service は P_3 で生成したバックドアプロセスを介して Trainee Host 上で動作していた Web アプリケーションのプロセスを kill することで，当該アプリケーションをサービス不能状態にする．なお，当該アプリケーションの停止が確認された時点で攻撃を終了する.

P_6 ここまでの一連の攻撃を防御できなかった場合，受講者は Trainee Host のリカバリを実施する．リカバリは，以下の条件が全て満たされた時点で完了した

ものとする．

- Web アプリケーションに存在した OS Command Injection 脆弱性を無効化するような修正が行われていること．
- Attacker Service によって生成されたバックドアプロセスが削除されていること．
- Web アプリケーションが再起動され，本来のサービスを提供できる状態へと復旧していること．

なお，本演習コンテンツでは Web アプリケーションが実装されたスクリプトファイルを監視対象として設定している．

5.4 演習コンテンツの構成 (2)

2 つ目の演習コンテンツ (以下，演習コンテンツ B) として「脆弱なパスワードが設定されたユーザに対する辞書攻撃を契機として発生する不正な SSH 経由のアクセスおよび当該ユーザのホームディレクトリ内に存在するファイルの改ざん」を取り上げ，作成した．演習コンテンツ B の具体的なシナリオの流れを以下に示す．

P_1 Attacker Service は Trainee Host に対してポートスキャナを用いたポートスキャンを行う．ここで，ポート番号 22 において SSH サービスが提供されていることを確認できた場合，次の Phase に進む．なお，当該サービスはパスワードによる認証と公開鍵による認証の双方を提供している．

P_2 Attacker Service は SSH によってログイン可能なすべてのユーザに対して辞書攻撃を行う．ここで，特定のユーザ名とパスワードの組を用いてログインに成功した場合，次の Phase に進む．

P_3 Attacker Service は自身の公開鍵を不正にログインしたユーザのホームディレクトリ内に配置し，当該公開鍵を用いたログインに成功した時点で次の Phase に進む．

P_4 Attacker Service は不正にログインしたユーザのホームディレクトリ内に配置されたファイルを改ざんする．改ざんに成功した時点で攻撃を終了する．

P_5 ここまでの一連の攻撃を防御できなかった場合，受講者は Trainee Host のリカバリを実施する．リカバリは，以下の条件が全て満たされた時点で完了した

ものとする．

- 不正にログインされたユーザのパスワードが変更されていること．
- Attacker Service によって配置された公開鍵が削除されていること．
- 改ざんされたファイルが本来の状態に戻されていること．

なお，本演習コンテンツにおいて監視対象のファイルは存在しない．

第6章　実験・評価

本章では，実験によって本提案手法の効果を検証する．検証においては，実験用データおよび実際の演習の実施を通して収集した受講者の振る舞いデータから有効な振る舞いを解析することが可能か，また，解析に要する時間について検討する．また，実験の結果明らかになった課題について議論する．

6.1　解析環境

本章で実施する実験の結果は表 6.1 に示す環境において実施した．

表 6.1: 解析環境

項目	内容
ハードウェア	MacBook Air (M1, 2020)
OS	macOS Big Sur (Version 11.5)
CPU	Apple M1 3.20GHz
RAM	16GB
コンテナランタイム	Docker (Version 20.10.12)

6.2　実験用データを用いた実験

実験用データを用いた実験を演習コンテンツ A と演習コンテンツ B のそれぞれに対して行った．以下では，その内容と結果について述べる．

6.2.1 演習コンテンツ A

6.2.1.1 Attack Phase 中に生成された差分の解析

演習コンテンツ A における Attack Phase で実行される攻撃の防御を実現した差分を解析可能であることを検証するために，Phase P_2 で実行される攻撃の防御に成功するように擬似的に作成した差分を対象に解析を実施した．解析の対象となる差分は以下の通りである．なお，下記の差分のうち，差分 d_3 を適用した場合，Phase P_2 で実行される攻撃に対する防御に成功する．

d_1 Web アプリケーションが依存しているライブラリのアップデート．

d_2 Web アプリケーションプログラムの修正 (リクエストのクエリにセミコロンが含まれている場合に当該リクエストを破棄する処理の追加)．

d_3 Web アプリケーションプログラムの修正 (リクエストのクエリがコマンドとして認識されないようにするためのエスケープ処理の追加)．

表 6.2 に解析結果を示す．解析結果において，出力された有効な差分は事前の想定と一致していた．また，差分の組み合わせの総数が少なく，かつ各差分列に対する解析が短時間で完了したため，解析に要した時間の合計も低く抑えられている．

表 6.2: 演習コンテンツ A における実験用データを用いた場合の解析結果 (1)

解析結果	適用した差分	解析に要した時間 [秒]	累積和 [秒]
×	$[d_1]$	7.41	7.41
×	$[d_2]$	7.81	15.22
○	$[d_3]$	7.72	22.94
–	$[d_1, d_2]$	–	22.94
–	$[d_1, d_3]$	–	22.94
–	$[d_2, d_3]$	–	22.94
–	$[d_1, d_2, d_3]$	–	22.94
解析に要した時間の合計 [秒]			22.94
出力された有効な差分			$[d_3]$

なお，解析結果列の記号の意味は表 6.3 に示す通りであり，これ以降の解析結果の表においても同様の意味である．

表 6.3: 解析結果列の記号の説明

記号	意味
×	差分の適用による攻撃の防御，あるいは，リカバリ検査に失敗した．
○	差分の適用によって攻撃の防御，あるいは，リカバリ検査に成功した．
–	解析実施前に有効な差分が出力されたため，解析をスキップした．

6.2.1.2 Recovery Phase 中に生成された差分の解析

演習コンテンツ A における Recovery Phase で実行されるリカバリ検査を成功させるような差分を解析可能であることを検証するために，擬似的に作成した差分を対象に解析を実施した．解析の対象となる差分は以下の通りである．なお，下記の差分のうち，差分 d_3, d_4, d_5 を適用した場合，Recovery Phase で実行されるリカバリ検査に成功する．

d_1 Web アプリケーションが依存しているライブラリのアップデート．

d_2 Web アプリケーションプログラムの修正 (リクエストのクエリにセミコロンが含まれている場合に当該リクエストを破棄する処理の追加)．

d_3 Web アプリケーションプログラムの修正 (リクエストのクエリがコマンドとして認識されないようにするためのエスケープ処理の追加)．

d_4 Attacker Service によって生成されたバックドアプロセスの削除．

d_5 Web アプリケーションの再起動．

表 6.4 に解析結果を示す．解析結果において，出力された有効な差分は事前の想定と一致していた．ただし，差分の組み合わせの総数が多かったため，6.2.1.1 のケースと比較して解析に要した時間の合計が長くなった．

表 6.4: 演習コンテンツ A における実験用データを用いた場合の解析結果 (2)

解析結果	適用した差分	解析に要した時間	累積和
×	$[d_1]$	13.95	13.95
×	$[d_2]$	13.02	26.97
×	$[d_3]$	12.74	39.71
×	$[d_4]$	12.61	52.32
×	$[d_5]$	15.05	67.37
×	$[d_1, d_2]$	14.17	81.54
×	$[d_1, d_3]$	13.72	95.26
×	$[d_1, d_4]$	15.39	110.65
×	$[d_1, d_5]$	13.88	124.53
×	$[d_2, d_3]$	12.92	137.45
×	$[d_2, d_4]$	13.15	150.60
×	$[d_2, d_5]$	13.72	164.32
×	$[d_3, d_4]$	12.83	177.15
×	$[d_3, d_5]$	13.21	190.36
×	$[d_4, d_5]$	13.89	204.25
×	$[d_1, d_2, d_3]$	12.20	216.45
×	$[d_1, d_2, d_4]$	14.14	230.59
×	$[d_1, d_2, d_5]$	15.68	246.27
×	$[d_1, d_3, d_4]$	12.28	258.55
×	$[d_1, d_3, d_5]$	15.81	274.36
×	$[d_1, d_4, d_5]$	13.97	288.33
×	$[d_2, d_3, d_4]$	12.30	300.63
×	$[d_2, d_3, d_5]$	13.75	314.38
×	$[d_2, d_4, d_5]$	14.37	328.75
○	$[d_3, d_4, d_5]$	14.08	342.83
–	$[d_1, d_2, d_3, d_4]$	–	342.83
–	$[d_1, d_2, d_3, d_5]$	–	342.83
–	$[d_1, d_2, d_4, d_5]$	–	342.83
–	$[d_1, d_3, d_4, d_5]$	–	342.83
–	$[d_2, d_3, d_4, d_5]$	–	342.83
–	$[d_1, d_2, d_3, d_4, d_5]$	–	342.83
解析に要した時間の合計 [秒]			342.83
出力された有効な差分			$[d_3, d_4, d_5]$

6.2.2 演習コンテンツ B

6.2.2.1 Attack Phase 中に生成された差分の解析

演習コンテンツ B における Attack Phase で実行される攻撃の防御を実現した差分を解析可能であることを検証するために，Phase P_2 で実行される攻撃の防御に成功するように擬似的に作成した差分を対象に解析を実施した．解析の対象となる差分は以下の通りである．なお，下記の差分のうち，差分 d_2 を適用した場合，Phase P_2 で実行される攻撃に対する防御に成功する．

d_1 SSH サービスを提供するサーバ用ライブラリのアップデート.

d_2 脆弱なパスワードが設定されているユーザのパスワード更新.

d_3 SSH サービスが提供されているポート以外に対するアクセスのブロック.

表 6.5 に解析結果を示す．解析結果において，出力された有効な差分は事前の想定と一致していた．解析に要した時間の合計は，差分の組み合わせの総数自体は少ないが，各差分列に対する解析に時間を要した[1]ため，6.2.1.1 のケースよりも長くなった．

表 6.5: 演習コンテンツ B における実験用データを用いた場合の解析結果 (1)

解析結果	適用した差分	解析に要した時間 [秒]	累積和 [秒]
×	$[d_1]$	54.56	54.56
○	$[d_2]$	52.14	106.70
–	$[d_3]$	–	106.70
–	$[d_1, d_2]$	–	106.70
–	$[d_1, d_3]$	–	106.70
–	$[d_2, d_3]$	–	106.70
–	$[d_1, d_2, d_3]$	–	106.70
解析に要した時間の合計 [秒]			106.70
出力された有効な差分			$[d_2]$

6.2.2.2 Recovery Phase 中に生成された差分の解析

演習コンテンツ B における Recovery Phase で実行されるリカバリ検査を成功させるような差分を解析可能であることを検証するために，擬似的に作成した差分

[1] 一部の Phase においてユーザのパスワードを総当りで特定する攻撃を行っていたため.

を対象に解析を実施した．解析の対象となる差分は以下の通りである．なお，下記の差分のうち，差分 d_1, d_3, d_5 を適用した場合，Recovery Phase で実行されるリカバリ検査に成功する．

d_1 不正にログインされたユーザのパスワード更新.

d_2 Attacker Service によって確立された SSH コネクションの切断.

d_3 Attacker Service によって配置された公開鍵の削除.

d_4 SSH サービスを提供するサーバ用ライブラリのアップデート.

d_5 改ざんされたファイルの復元.

表 6.6 に解析結果を示す．解析結果において，出力された有効な差分は事前の想定と一致していた．ただし，6.2.1.2 のケースと同様に，差分の組み合わせが多く，かつ各差分列に対する解析に時間を要しため，解析に要した時間の合計はこれまでのケースで最も長くなった．

表 6.6: 演習コンテンツ B における実験用データを用いた場合の解析結果 (2)

解析結果	適用した差分	解析に要した時間 [秒]	累積和 [秒]
×	$[d_1]$	51.57	51.57
×	$[d_2]$	52.37	103.94
×	$[d_3]$	49.49	153.43
×	$[d_4]$	52.12	205.55
×	$[d_5]$	52.73	258.28
×	$[d_1, d_2]$	51.45	309.73
×	$[d_1, d_3]$	48.90	358.63
×	$[d_1, d_4]$	48.62	407.25
×	$[d_1, d_5]$	54.22	461.47
×	$[d_2, d_3]$	49.03	510.50
×	$[d_2, d_4]$	51.63	562.13
×	$[d_2, d_5]$	44.39	606.52
×	$[d_3, d_4]$	53.49	660.01
×	$[d_3, d_5]$	45.78	705.79
×	$[d_4, d_5]$	51.90	757.69
×	$[d_1, d_2, d_3]$	50.28	807.97
×	$[d_1, d_2, d_4]$	50.54	858.51
×	$[d_1, d_2, d_5]$	47.91	906.42
×	$[d_1, d_3, d_4]$	51.84	958.26
○	$[d_1, d_3, d_5]$	52.45	1010.71
–	$[d_1, d_4, d_5]$	–	1010.71
–	$[d_2, d_3, d_4]$	–	1010.71
–	$[d_2, d_3, d_5]$	–	1010.71
–	$[d_2, d_4, d_5]$	–	1010.71
–	$[d_3, d_4, d_5]$	–	1010.71
–	$[d_1, d_2, d_3, d_4]$	–	1010.71
–	$[d_1, d_2, d_3, d_5]$	–	1010.71
–	$[d_1, d_2, d_4, d_5]$	–	1010.71
–	$[d_1, d_3, d_4, d_5]$	–	1010.71
–	$[d_2, d_3, d_4, d_5]$	–	1010.71
–	$[d_1, d_2, d_3, d_4, d_5]$	–	1010.71
解析に要した時間の合計 [秒]			1010.71
出力された有効な差分			$[d_1, d_3, d_5]$

6.3 演習の実施による実験

過去にサイバー防御演習に参加した経験を有する大学生 2 名 (以下では，被験者 A および被験者 B とする) を対象に，演習コンテンツ A および演習コンテンツ B を用いた実験を行った．なお，被験者 A は演習コンテンツ A を，被験者 B は演習コンテンツ B を用いた演習をそれぞれ実施した．以下では，その演習の結果とそれを対象とした解析の結果について述べる．

6.3.1 演習コンテンツ A

6.3.1.1 演習時の流れとログ

被験者 A は，演習開始直後から Web アプリケーションが実装されたスクリプトファイルをシェル経由で読み取り，脆弱性が存在する処理を探索していたが，Attack Phase で実行される一連の攻撃が完了するよりも前に対策を行うことができず，Recovery Phase へと移行した．その後，脆弱性の修正や Web アプリケーションの再起動，バックドアプロセスの削除等を実施し，演習が完了した．本実験における Condition の変遷とその過程で生成された差分の一覧を表 6.7 に示す．

6.3.1.2 演習結果を用いた解析

前述の演習結果を対象に解析を実施した．当該演習では攻撃の防御に失敗し Recovery Phase に移行しているため，Recovery Phase で実行されるリカバリ検査を成功に導いた差分を解析する．解析対象の差分は以下の通りである．

d_1 Attacker Service によって生成されたバックドアプロセスの削除.

d_2 Web アプリケーションプログラムの修正 (リクエストのクエリがシェル経由で実行されないようにするための処理の実装).

d_3 Web アプリケーションの再起動.

表 6.8 に解析結果を示す．解析結果において，出力された有効な差分は 6.2.1.2 で想定していたものと順番は異なるが一致していた．

表 6.7: 演習コンテンツ A を用いた演習実施時のログ

Condition	**Transition**	**被験者が生成した差分**
$(P_1,\ S_1)$	T_1	• テキストエディタのインストール
$(P_1,\ S_2)$	T_3	–
$(P_2,\ S_1)$	T_1	–
$(P_2,\ S_2)$	T_3	–
$(P_3,\ S_1)$	T_1	–
$(P_3,\ S_2)$	T_3	• Web アプリケーションプログラムの不完全な修正 (リクエストのクエリがシェル経由で実行されないようにするための処理の実装)
$(P_4,\ S_1)$	T_1	–
$(P_4,\ S_2)$	T_3	–
$(P_5,\ S_1)$	T_1	• Web アプリケーションの再起動
$(P_5,\ S_2)$	T_3	–
$(P_6,\ S_1)$	T_2	–
$(P_6,\ S_1)$	T_2	• Attacker Service によって生成されたバックドアプロセスの削除
$(P_6,\ S_1)$	T_1	• Web アプリケーションプログラムの修正 (リクエストのクエリがシェル経由で実行されないようにするための処理の実装) • Web アプリケーションの再起動
$(P_6,\ S_e)$	–	–

表 6.8: 演習コンテンツ A における演習結果を用いた場合の解析結果

解析結果	**適用した差分**	**解析に要した時間 [秒]**	**累積和 [秒]**
×	$[d_1]$	13.93	13.93
×	$[d_2]$	13.35	27.28
×	$[d_3]$	14.42	41.70
×	$[d_1, d_2]$	13.23	54.93
×	$[d_1, d_3]$	14.60	69.53
×	$[d_2, d_3]$	14.52	84.05
○	$[d_1, d_2, d_3]$	15.68	99.73
解析に要した時間の合計 [秒]			99.73
出力された有効な差分			$[d_1, d_2, d_3]$

6.3.2 演習コンテンツ B

6.3.2.1 演習時の流れとログ

被験者 B は，演習開始直後に SSH 関連のシステムログを定期的に確認することで SSH 経由の攻撃を監視していたが，Attack Phase で実行される一連の攻撃が完了するよりも前に対策を行うことができず，Recovery Phase へと移行した．その後，改ざんされたファイルの復元や Attacker Service によって配置された公開鍵の削除，不正にログインされたユーザのパスワード更新等を実施し，演習が完了した．

本実験における Condition の変遷とその過程で生成された差分の一覧を表 6.9 に示す．

表 6.9: 演習コンテンツ B を用いた演習実施時のログ

Condition	Transition	被験者が生成した差分
$(P_1,\ S_1)$	T_1	• SSH サービスの起動 • テキストエディタのインストール
$(P_1,\ S_2)$	T_3	–
$(P_2,\ S_1)$	T_1	• ログファイルが配置されているディレクトリへの移動
$(P_2,\ S_2)$	T_3	–
$(P_3,\ S_1)$	T_1	–
$(P_3,\ S_2)$	T_3	–
$(P_4,\ S_1)$	T_1	–
$(P_4,\ S_2)$	T_3	–
$(P_5,\ S_1)$	T_2	• 不正にログインされたユーザのホームディレクトリへの移動 • 改ざんされたファイルの復元 (相対パスの指定によるファイルの上書き) • Attacker Service によって確立された SSH コネクションの切断
$(P_5,\ S_1)$	T_2	• Attacker Service によって配置された公開鍵の削除
$(P_5,\ S_1)$	T_1	• 不正にログインされたユーザのパスワード更新
$(P_5,\ S_e)$	–	–

6.3.2.2 演習結果を用いた解析

演習コンテンツ A の場合と同様に，前述の演習結果を対象に解析を実施した．当該演習では攻撃の防御に失敗し Recovery Phase に移行しているため，Recovery Phase で実行されるリカバリ検査を成功に導いた差分を解析する．解析対象の差分は以下の通りである．

d_1 不正にログインされたユーザのホームディレクトリへの移動.

d_2 改ざんされたファイルの復元 (相対パスの指定によるファイルの上書き).

d_3 Attacker Service によって確立された SSH コネクションの切断.

d_4 Attacker Service によって配置された公開鍵の削除.

d_5 不正にログインされたユーザのパスワード更新.

表 6.10 に解析結果を示す．解析結果において，出力された有効な差分は 6.2.2.2 で想定していたものとは異なっていた．これは，6.2.2.2 における差分「改ざんされたファイルの復元」を，本ケースでは「不正にログインされたユーザのホームディレクトリへの移動」と「改ざんされたファイルの復元 (相対パスの指定によるファイルの上書き)」によって実現していたためである．これに伴い，解析対象となる差分が 6.2.2.2 のケースと比較して増加したため，解析に要した時間の合計も同様に増加した．

表 6.10: 演習コンテンツ B における演習結果を用いた場合の解析結果

解析結果	適用した差分	解析に要した時間 [秒]	累積和 [秒]
×	$[d_1]$	50.53	50.53
×	$[d_2]$	47.64	98.17
×	$[d_3]$	47.68	145.85
×	$[d_4]$	46.21	192.06
×	$[d_5]$	54.07	246.13
×	$[d_1, d_2]$	48.82	294.95
×	$[d_1, d_3]$	55.93	350.88
×	$[d_1, d_4]$	53.61	404.49
×	$[d_1, d_5]$	52.40	456.89
×	$[d_2, d_3]$	47.37	504.26
×	$[d_2, d_4]$	52.88	557.14
×	$[d_2, d_5]$	51.97	609.11
×	$[d_3, d_4]$	46.03	655.14
×	$[d_3, d_5]$	48.62	703.76
×	$[d_4, d_5]$	52.72	756.48
×	$[d_1, d_2, d_3]$	49.39	805.87
×	$[d_1, d_2, d_4]$	48.30	854.17
×	$[d_1, d_2, d_5]$	46.17	900.34
×	$[d_1, d_3, d_4]$	47.86	948.20
×	$[d_1, d_3, d_5]$	50.97	999.17
×	$[d_1, d_4, d_5]$	53.12	1052.29
×	$[d_2, d_3, d_4]$	49.28	1101.57
×	$[d_2, d_3, d_5]$	46.01	1147.58
×	$[d_2, d_4, d_5]$	50.86	1198.44
×	$[d_3, d_4, d_5]$	45.23	1243.67
×	$[d_1, d_2, d_3, d_4]$	44.93	1288.60
×	$[d_1, d_2, d_3, d_5]$	53.96	1342.56
○	$[d_1, d_2, d_4, d_5]$	52.83	1395.39
–	$[d_1, d_3, d_4, d_5]$	–	1395.39
–	$[d_2, d_3, d_4, d_5]$	–	1395.39
–	$[d_1, d_2, d_3, d_4, d_5]$	–	1395.39
解析に要した時間の合計 [秒]			1395.39
出力された有効な差分			$[d_1, d_2, d_4, d_5]$

6.4 有効性と解析時間の評価

まず、提案手法の有効性について議論する。実験の結果，すべての場合において有効な差分の抽出が可能であることが示された．一方で，今回の実験においては観測対象とする差分をコマンドの実行履歴とファイルの変更履歴を対象に抽出していた．この手法には演習の実施ごとに異なる値 (たとえば，プロセス ID や Attacker Service が利用する動的なポート番号等) の再現や監視対象外のファイルに対する操作を監視することが困難であるという課題が存在する．

次に，解析時間について議論する．前述の実験結果において，D' の要素数や実行の並列度によっては現実的な時間のもとで解析可能であることが示された．しかしながら，D' の要素数が多くなった場合や解析の並列化のための十分なリソースを用意できない場合は，解析を現実的な時間で完了させることが困難になるケースが発生し得る．

解析の実施に要する時間は 4.4 で示した手順全体を実行するために要する時間である．ここで，D' の各要素に応じたサイバーレンジのインスタンス生成や，生成されたサイバーレンジにおける攻撃等の実施は，D' の要素ごとに並列化することが可能である．そのため，解析に要する時間はおおよそ以下の計算式によって見積もることができる．

$$\#D' = \sum_{r=1}^{\#D} \binom{\#D}{r}$$

$$T_{max} = \#D' \cdot (T_{manifest} + T_{build} + T_{action})/P$$

表 6.11: 計算式中の変数の説明

変数名	説明
$\#D$	D の要素数.
T_{max}	解析における最悪実行時間.
$\#D'$	D' の要素数.
$T_{manifest}$	コンテナイメージ用のマニフェスト生成に要する時間.
T_{build}	コンテナイメージのビルドに要する時間.
T_{action}	攻撃やリカバリ検査の実行に要する時間.
P	実行の並列度.

図 6.1 に，$T_{manifest}$ および T_{build} を無視した上で，$T_{action}, \#D, P$ のそれぞれについて値を変化させた場合の T_{max} の値を図示する．図に示す通り，実行の並列度が低い場合 ($P = 1$ や $P = 2$) は $\#D$ の増加に対する T_{max} の増加が顕著だが，並列度が十分に高い場合 ($P = 5$ や $P = 10$) は $\#D$ がある程度増加しても T_{max} を低く抑えることが可能である．

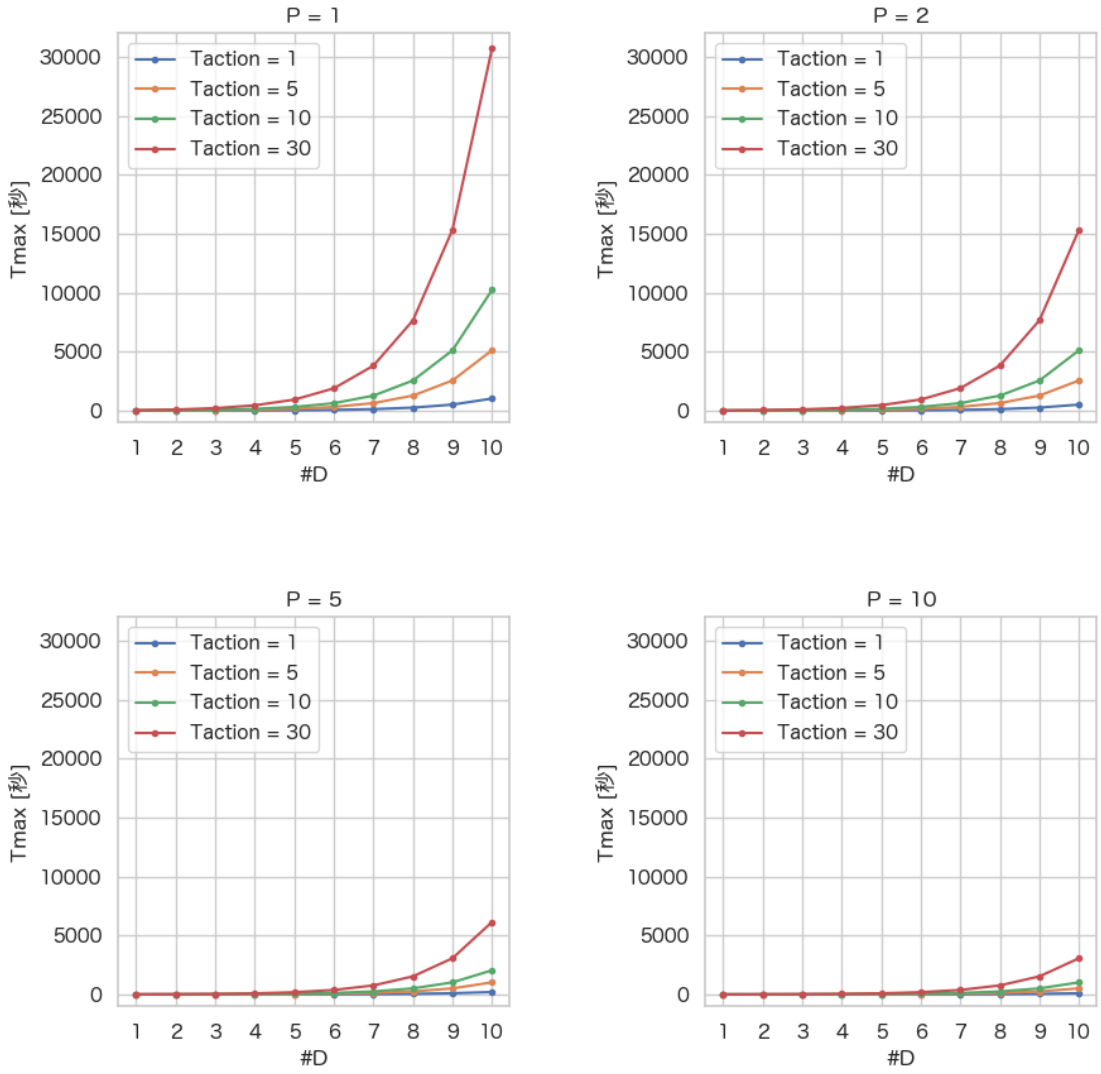

図 6.1: T_{max} の変化

第7章　おわりに

本章では，本研究を総括するとともに，実験を通して明らかとなった課題や今後の展望について述べる．

7.1　本研究の総括

本研究では，サイバーレンジ上で実施されるサイバー防御演習を通したスキルの習得を促進させるための仕組みについて検討した．具体的には，受講者の振る舞いによってサイバーレンジにもたらされた変化を差分として捉えた上で，サイバー防御演習の進行状況を状態遷移モデルと統合することで差分を体系的に抽出する手法，および，抽出した差分のうち，演習内で実行される攻撃への対策を成功に導いた差分を自動的に解析する手法を提案した．また，本提案手法を実施可能な環境において演習を行うことで，演習後の振り返りに活用可能な受講者の振る舞いに関する情報を体系的に抽出，解析可能であることを示した．

7.2　今後の展望

7.2.1　差分抽出のために監視する対象の網羅性

5.2 にて述べた通り，本研究では，受講者による振る舞いの結果生成される差分として，コマンドの実行履歴と特定のファイルの変更履歴を対象とした．これらの情報はサイバー防御演習における受講者の振る舞いを観測する上で有用だが，網羅性の観点においては十分ではない．たとえば，現状の対象のみでは，GUI を介して行われたシステムに対する変更や観測対象としていないファイルに対する変更などを観測することができない．この課題に対処するためには，OS に対して発行されたシステムコールのフックなどによる，より高度な差分の監視機構を検討する必要がある．

7.2.2 差分の再現性

本研究では差分として捉える対象のひとつとしてコマンドの実行履歴を採用していたが，6.4 で述べた通り，そのコマンド内に演習の実施ごとに異なる値が含まれていた場合，たとえ解析において観測されたものと同様のコマンドを実行したとしても，解析環境において演習時の状況を再現できるとは限らない．たとえば，演習時にあるバックドアプロセスを停止する目的で受講者が"kill -9 <PID>"というコマンドを実行した場合，解析環境においてバックドアプロセスのプロセス ID が<PID>と同一であるという保証はない．この課題に対処するためには，7.2.1 と同様に，実行された差分をシステムコールのレベルで監視するといった機構の導入を検討する必要がある．

7.2.3 解析アルゴリズムのパフォーマンス

本研究の提案手法が出力する情報はサイバー防御演習の直後に受講者に対して提供するものであるため，可能な限り短い時間で解析を実行しなければならない．しかしながら，現状のアルゴリズムでは解析対象の差分の数に従って解析に要する時間が指数関数的に増大していくため，解析対象の差分の数が多くなった場合は本来の要件が達成できないおそれがある．この課題に対処するためには，依存関係のある差分同士を結合することで差分全体の数を少なく抑える仕組みや，ホストに影響を及ぼさない差分をより厳密に取捨選択する仕組みを導入する必要がある．

7.2.4 差分として観測できない事項に関する検討

本研究では受講者の振る舞いによってサイバーレンジにもたらされた変化のみを差分として捉えていたが，実際には差分として観測できないながらも検討を要する事項が存在する．たとえば，受講者は自身に割り当てられたホストのシステム情報や当該ホスト上に存在するファイル等を参照することで，当該ホストに存在する脆弱性や攻撃への対処法を考察する．本研究において，これらの振る舞いは観測対象としていないが，受講者の振る舞いをより多面的に捉えるという観点においては，サイバーレンジには差分として現れない受講者のメンタルモデルの変化等も考慮すべきだと考えられる．

謝辞

本研究を進めるにあたり，主指導教員の篠田陽一教授には研究の進め方や対外発表の方法など，様々な観点から多大なご指導を賜りました．心から感謝いたします．

本研究室の宇多仁助教，博士後期課程の三浦良介氏には，研究に関する議論等におきまして多大なご指導を賜りました．深く感謝いたします．また，丹康雄教授には，インターンシップ指導教員をお引き受け頂けただけではなく，研究に関しても様々な助言を賜りました．深く感謝いたします．

本研究室修了生の渡邊司揮氏，油布翔平氏，吉原昴司氏，知念研究室修了生の門脇真之佑氏，古寺雄馬氏には研究に関する様々な議論や研究生活を送る上での多大なご助力を頂きました．深く感謝いたします．

本研究室の博士前期課程の馬越紘氏，本間可楠氏，岡田真一氏，片岡拓海氏，瀧島和則氏には研究に関する活発な議論に加えて研究生活を送る上でも多大なご支援を頂きました．深く感謝いたします．

最後に，これまでの学生生活を支えて頂いた家族に深く感謝いたします．

参考文献

[1] 我が国のサイバーセキュリティ人材の現状について，総務省 (オンライン), https://www.soumu.go.jp/main_content/000591470.pdf (参照 2022/01/13).

[2] R. Beuran, D. Tang, C. Pham, K. Chinen, Y. Tan, Y. Shinoda: *Integrated Framework for Hands-on Cybersecurity Training: CyTrONE*, Elsevier Computers & Security, Vol.78C, pp.43-59 (2018).

[3] 豊田 真一，中田 亮太郎，長谷川 久美，慎 祥揆，瀬戸 洋一: エコシステムで構成するサイバー攻撃と防御演習システム CyExec の提案，コンピュータセキュリティシンポジウム 2018 論文集，Vol.2018, No.2 (2018).

[4] 太田 悟史，安田 真悟，湯村 翼，高野 祐輝: 次世代サイバー演習環境に向けて, マルチメディア，分散協調とモバイルシンポジウム 2016 論文集, Vol.2016, pp. 1776-1782 (2016).

[5] 井上 拓哉, Razvan Beuran: 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, Internet Conference (2018).

[6] 古寺 雄馬, 知念 賢一: サイバーセキュリティ演習巻き戻し機構の設計と実装, 研究報告セキュリティ心理学とトラスト (SPT), Vol.2021-SPT-41, No.14, pp.1-6 (2021).

[7] C. Gabriele, R. Enrico, A. Alessandro: *Automating the Generation of Cyber Range Virtual Scenarios with VSDL*, arXiv(オンライン), https://arxiv.org/abs/2001.06681 (参照 2022/01/14).

[8] R. Beuran, C. Pham, D. Tang, K. Chinen, Y. Tan, Y. Shinoda: *Cybersecurity Education and Training Support System: CyRIS*, IEICE Transactions on Information and Systems, Volume.E101.D, No.3, pp.740-749 (2018).

[9] crond-jaist/cyprom, GitHub(オンライン),
https://github.com/crond-jaist/cyprom (参照 2022/01/13).

[10] R. Beuran, D. Tang, Z. Tan, S. Hasegawa, Y. Tan, Y. Shinoda: *Supporting Cybersecurity Education and Training via LMS Integration: CyLMS*, Springer Education and Information Technologies, Volume.24, No.6, pp.3619-3643 (2019).

[11] NIST: SP 800-181 Rev.1 Workforce Framework for Cybersecurity (NICE Framework).

[12] Hardening Project 2021, WASForum(オンライン),
https://wasforum.jp/hardening-project/ (参照 2022/01/13).

[13] NICT-情報通信研究機構, NICT(オンライン), https://www.nict.go.jp/ (参照 2022/01/13).

[14] C. Nestoras, K. George, K. Ioanna, M. Leandros, P. Grammati, F. M. Amine: *Cyber Ranges and TestBeds for Education, Training, and Research*, Applied Sciences, Vol.11, No.1809 (2021).

[15] 村木 優太, 上原 哲太郎: サイバーレンジ演習環境展開の高速化手法, 研究報告コンピュータセキュリティ(CSEC), Vol.2018-CSEC-83, No.1, pp.1-7 (2018).

[16] 寺嶋 友哉, 小出 洋: 分散環境における拡張性を持つサイバーレンジ構築手法の提案と評価, 火の国情報シンポジウム 2020, 一般社団法人 情報処理学会 九州支部 (オンライン), https://www.ipsj-kyushu.jp/page/ronbun/hinokuni/1009/Papers/C2-4.pdf (参照 2022/01/14).

[17] 砂川 真範, 知念 賢一, 篠田 陽一: インシデントの再現を目的としたサイバーレンジ構築支援システムの提案, マルチメディア，分散協調とモバイルシンポジウム 2018 論文集, Vol.2018, pp.1823-1827.

[18] Eric M. Hutchins, Michael J. Cloppert, Rohan M. Amin: *Intelligence-driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains*, Leading Issues in Information War-fare & Security Research, Vol.1, pp.80–106 (2011).

[19] 川口 直也: コンテナ型仮想化概論, pp.3-5, カットシステム (2020).

[20] Docker, Docker(オンライン), https://www.docker.com/ (参照 2022/01/19).

[21] StarBED4 プロジェクト, NICT(オンライン), https://starbed.nict.go.jp/ (参照 2022/01/13).

[22] 情報危機管理コンテスト, サイバー犯罪に関する白浜シンポジウム実行委員会, https://sccs-jp.org/symposium26/cmc17/whats-cmc/ (参照 2022/01/13).

[23] CYDER, NICT(オンライン), https://cyder.nict.go.jp/ (参照 2022/01/13).

本研究に関する対外発表

[1] 梅内 翼, 篠田 陽一: 状態遷移モデルを活用したサイバー防御演習における受講者の振る舞い解析の高度化に関する研究, コンピュータセキュリティシンポジウム 2021 論文集, Vol.2021, pp.462-469 (2021).

[2] T. Umeuchi: Integrating Cyber Kill Chain and State Transition Models for Enhancement of Incident Response Exercises, WIDE 2021 May Meeting (2021).