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

梅内 翼[1,a)]　篠田 陽一[1,b)]

概要：サイバーレンジ上で実施されるサイバー防御演習は，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．そのためには，単に演習を実施するだけではなく，受講者のどのような振る舞いが演習内で実行される攻撃の防御を成功に導いたのかを演習終了後に振り返るプロセスが不可欠である．そこで，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の防御を成功に導いた差分を解析する手法を提案する．本稿では，まず，既存のサイバーレンジやサイバー防御演習に関する課題について関連研究を取り上げつつ検討する．次に，状態遷移モデルの定義や，それを用い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を提示する．最後に，提案手法を実装するとともに，当該実装を用いて行った実験に対する考察を示す．実験の結果，本提案手法による振る舞い解析の高度化が確認された一方で，差分の抽出における課題が明らかになった．

キーワード：サイバーレンジ，サイバー防御演習，状態遷移モデル，振る舞い解析

A Study on Advanced Analysis of Participants' Behavior in Cyber Defense Exercise using State Transition Model

Tsubasa Umeuchi[1,a)]　Yoichi Shinoda[1,b)]

Abstract: 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 the experiments, we confirmed that the proposed method can improve the behavior analysis, but the challenges in extracting the differences were revealed.

Keywords: Cyber Range, Cyber Defense Exercise, State Transition Model, Behavior Analysis

1. はじめに

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

[1] 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology
[a)] tsubasa.umeuchi@jaist.ac.jp
[b)] shinoda@jaist.ac.jp

スに関するスキルを習得させるための手段として，演習のために構築された仮想環境 (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, 9] が存在する．そして，文献 [10] 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 [9] 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．

そこで，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，解析可能になるとともに，解析結果を用いた演習後の振り返りプロセスの活性化が期待される．

本稿の構成を以下に示す．第 2 節では，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，本研究がターゲットとする課題を明らかにする．第 3 節では，提案手法である状態遷移モデルを用いた受講者の振る舞い解析について，状態遷移モデルの定義やそれを用いた受講者の振る舞いの抽出，解析手法について説明する．第 4 節では，本提案手法を実現するためのサイバーレンジや演習シナリオの設計と実装について説明する．第 5 節では，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，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 6 節では，本研究を総括するとともに，今後の展望について論じる．

2. 背景

2.1 サイバーレンジの定義

サイバーレンジとは，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のことである．この仮想空間内には，現実世界のホストやネットワークを模倣したインフラストラクチャが仮想化技術を活用して構築される．さて，サイバーレンジの構築や運用については，次項以降に示す複数の課題と関連研究が存在する．

2.2 構築手法における課題

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 [11]，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 [12] が提案されている．

2.3 演習のリアリティにおける課題

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 [13] が提案されている．

2.4 演習内容における課題

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, 13] 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．

2.5 演習を通したスキルの習得における課題

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 [9] 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．

本稿では，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」を解決するための機構を提案する．

3. 提案手法

3.1 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．

本稿の提案手法は，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差分をすべて抽出する．

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 1 に，提案手法の概要図を示す．

3.2 状態遷移モデル

本項では，状態遷移モデルの定義について述べる．

3.2.1 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 2 に，状態遷移モデルの概要図を示す．

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

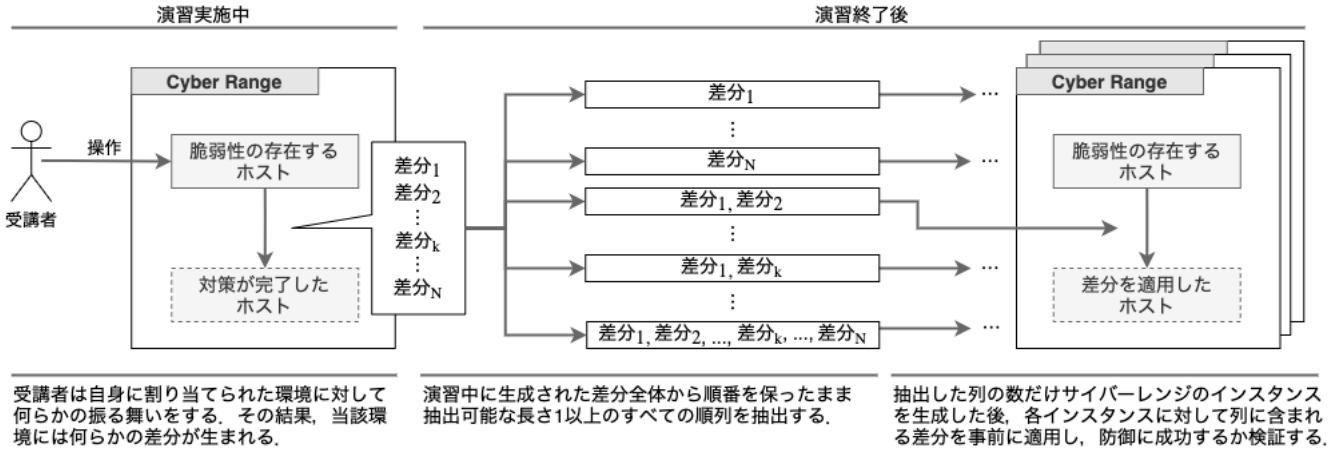

図 1 提案手法の概要図
Fig. 1 Diagram of Proposed Method.

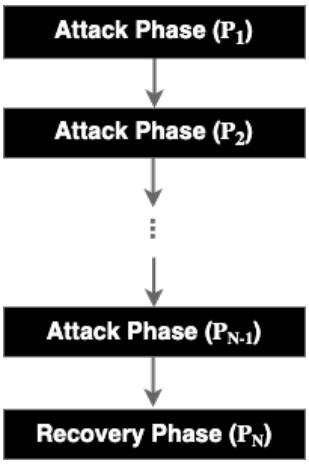

図 2 状態遷移モデルの概要図
Fig. 2 Diagram of State Transition Model.

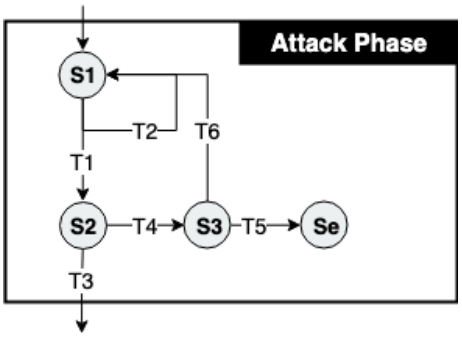

図 3 Attack Phase の概要図
Fig. 3 Diagram of Attack Phase.

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．

3.2.2 Attack Phase

Attack Phase は 4 つの State と 6 つの Transition から構成される．Attack Phase の概要図を図 3 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．

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．

3.2.3 Recovery Phase

Recovery Phase は 2 つの State と 2 つの Transition から構成される．Recovery Phase の概要図を図 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．

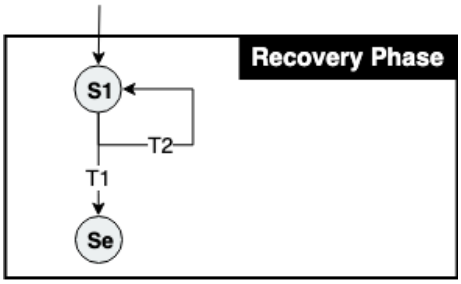

図 4 Recovry Phase の概要図
Fig. 4 Diagram of Recovery Phase.

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.

3.3 差分の抽出と解析

本項では，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示す.

3.3.1 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.

3.3.2 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.

3.3.3 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.

3.4 有効な差分の解析

本項では，前述した解析対象として抽出した差分全体をもとに，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.

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

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，リカバリ検査を実行する場合は，すべての Attack Phase の攻撃が成功した場合の状態を事前に再現しておく．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 を結果として出力する.

4. 設計と実装

4.1 サイバーレンジのアーキテクチャ

本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を図 5 に示す.

図に示した通り，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.

このうち，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

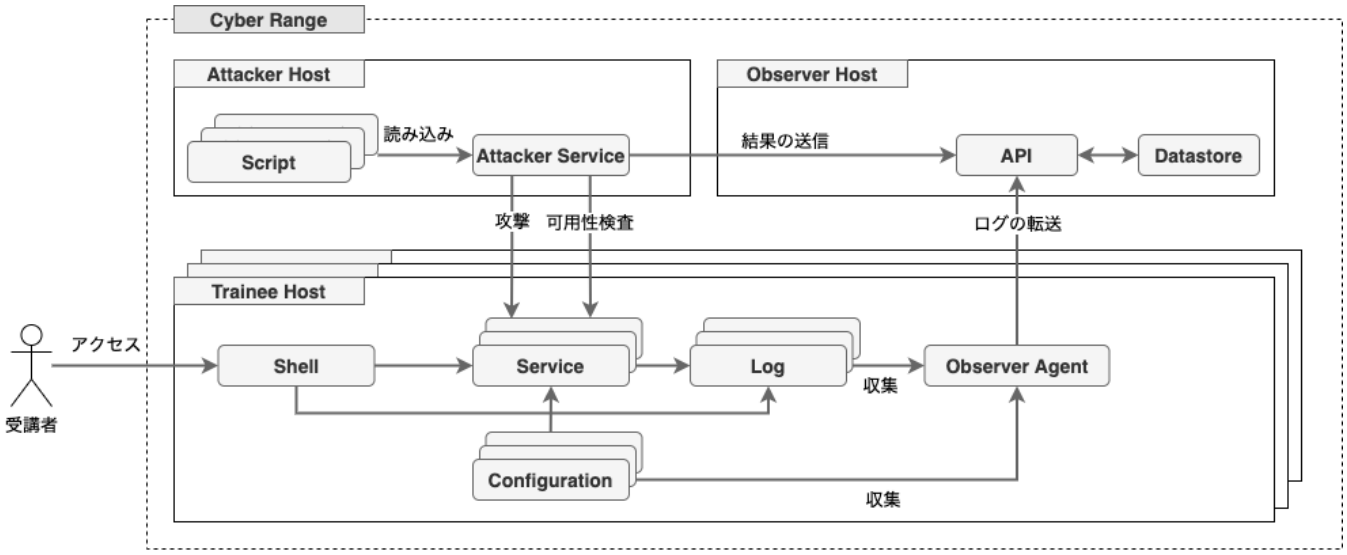

図 5 サイバーレンジのアーキテクチャ
Fig. 5 Architecture of Cyber Range.

存在する．

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．

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．

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．

4.2 演習シナリオの定義とサイバーレンジの実装

本項では後述する実験で用いる演習シナリオの定義と，そのシナリオを実現するためのサイバーレンジの実装について述べる．

さて，2.4 で述べた通り，演習シナリオではリアリティのある攻撃を再現する必要がある．今回，筆者らは演習シナリオとして「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」を取り上げた．演習シナリオの具体的な流れを以下に示す．

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，次のステップに進む．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，次のステップに進む．

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ステップに進む．

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，受講者は自ホスト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．

筆者らはコンテナ型仮想化技術の一種である Docker を用いて上記の演習を実施するためのサイバーレンジを構築した．各ホストは `ubuntu:focal` をベースイメージとするコンテナを用いて構築した．また，ホスト間のネットワークは Docker のネットワーク機能を用いて構築した．

4.3 差分の抽出と解析

本稿では，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ファイルの変更履歴を対象とする．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 (たとえば，`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，差分の適用は，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 1 に示す 3 種類の Dockerfile を用いた Trainee Host のイメージが生成される．

5. 実験と考察

5.1 実験の概要と実験環境

防御やリカバリを実現した差分を解析可能であることに加えて，解析を現実的な時間で完了することが可能であることを確認するために，4 で示したサイバーレンジおよび演習シナリオを用いた実験を表 2 に示す環境で実施した．

表 1 生成される Dockerfile の例
Table 1 Examples of Generated Dockerfile.

差分 d_1 を適用したイメージを生成する Dockerfile	`FROM ubuntu:focal`
	`...`
	`RUN` c_1
	`...`

差分 d_2 を適用したイメージを生成する Dockerfile	`FROM ubuntu:focal`
	`...`
	`RUN` c_2
	`...`

差分 d_1, d_2 を適用したイメージを生成する Dockerfile	`FROM ubuntu:focal`
	`...`
	`RUN` c_1 `&&` c_2
	`...`

表 2 実験環境
Table 2 Experiment Environment.

項目	内容
ホスト 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 20.10.7)

5.2 防御を実現した差分の解析

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 で実行される攻撃を防御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．

(1) SSH サービスを提供するサーバのアップデート．
(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．
(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．

上記の差分のうち，Phase P_2 で実行される攻撃に対する防御の成功に寄与するのは (2) の差分のみである．解析の結果，(2) のみで構成される差分が結果として出力されることを確認した．

5.3 リカバリを実現した差分の解析

リカバリ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，リカバリ検査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．

(1) 不正にログインされたユーザのパスワード更新．
(2) Attacker Service によって確立された SSH コネクションの切断．
(3) Attacker Service によって配置された公開鍵の削除．

(4) SSH サービスの再起動.
(5) 改ざんされたファイルの復元.

上記の差分のうち，リカバリ検査の成功に寄与するのは (1)，(3)，(4) の差分の組み合わせである．解析の結果，(1)，(3)，(4) で構成される差分が結果として出力されることを確認した．

5.4 解析に要する時間に関する検討

解析に要する時間は 3.4 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．

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

T_{all}	解析に要する時間.
$\#D'$	D' の要素数.
$T_{manifest}$	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.

5.5 考察

実験の結果，有効な差分の抽出が可能であることが示された．また，そのパフォーマンスについても，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は困難である．また，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が，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．この課題を解決するためには，既存の仕組みに加えて，システムコールのフック等を用いた，より高度な監視機構を導入する必要がある．

6. おわりに

本稿では，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．今後は，差分抽出の網羅性や汎用性を高めるための高度な監視機構の導入を進める．

参考文献

[1] 総務省: 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)，入手先 ⟨https://www.soumu.go.jp/main_content/000591470.pdf⟩ (参照 2021-07-20).

[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), 入手先 ⟨https://arxiv.org/abs/2001.06681⟩ (参照 2021-07-20).

[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] 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).

[10] 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).

[11] 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).

[12] 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), 入手先 ⟨https://www.ipsj-kyushu.jp/page/ronbun/hinokuni/1009/Papers/C2-4.pdf⟩ (参照 2021-07-20).

[13] 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.