7ZH-03

Linux初学者に向けた試行錯誤を可能とするセキュリティ演習システムにおける試行錯誤機構の開発

竹原一駿 †1　石塚美伶 †1　亀井仁志 †1　喜田弘司 †1　最所圭三 †1
香川大学 †1

1. はじめに

近年のセキュリティ人材の不足を受けて，大学などの教育機関には，攻撃に対し事前の予防や的確な対処ができるセキュリティ人材の育成が求められている．このような人材を育成する演習の 1 つに，ハードニング演習がある．

ハードニング演習を大学で行う場合の受講者は，Linux コマンドの使い方を学習したばかりの初学者であることが多い．そのため受講者は演習中の攻撃に対し，複数ある防御手法のうち，どの防御手法が攻撃に適しているか選ぶことができず，場当たり的な防御手法に終始し，最適な防御手法を学ぶことができない．

そこで我々は，サイバー攻撃に対し複数ある防御手法をそれぞれ検討でき，何度でも試行錯誤できる (やり直せる) システム “ぷろてっくん” を開発している [1]．本稿では，開発したぷろてっくんの試行錯誤機構について述べる．

2. 試行錯誤機構を用いた演習の想定

一般に，攻撃への防御手法は複数ある．初学者が，攻撃に合わせて最も適切な手法を選択できるように学習することが課題である．試行錯誤機構を用いてハードニング演習を行うことで，受講者は 1 つの攻撃に対し様々な手法を検討できる．失敗しても反復することで，受講者は攻撃に対し最適な手法を発見できる．これにより，実際にサービスを運営する際に，様々な手法に迷うこと無く，最適な手法を展開できる人材を育成できる．

図 1 に試行錯誤のイメージを示す．実際の演習では，試行錯誤機構を以下の流れで用いる．受講者は，攻撃を受ける前に，後ほどリストアするために演習状態をセーブする (セーブポイント SP: セーブ A)．受講者は攻撃に気づいた後に，攻撃への事前対策を施すために攻撃を受ける前のセーブ A にリストアする．

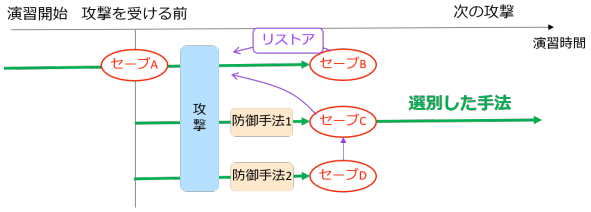

図 1: 試行錯誤のイメージ

防御手法 1 を実践し，攻撃を受け防御できているかを確認し，セーブする (セーブ B)．その後，再度セーブ A にリストアし，異なる防御手法を検証できる (セーブ C，セーブ D)．また，敢えて更に脆弱な状態にし，攻撃を受けた際にどのような影響がでるか，検証することもできる．これらの SP は，攻撃と防御を繰り返す上で，木構造で構成される．

3. 試行錯誤機構の要件

2節の演習を実現するには，以下に示す要件を満たす必要がある．

① **セーブ・リストアの任意性:** 受講者は，演習中に攻撃を受ける直前や直後セーブ・リストアを行う．ハードニングでの演習をしながら (サービスを守りながら)，行えることが望ましい．

② **SP 管理の容易性:** 受講者毎にセーブするタイミングや数は異なる．受講者毎に SP を管理できる機構が必要である．また，複数の防御手法を実践し，木構造で構成される SP を管理する必要がある．

③ **攻撃タイミングの同一性:** 受講者が同じ攻撃に対し様々な手法を検証できるようにするために，セーブ・リストアした後にも，同じタイミングで再度攻撃する必要がある．

4. 試行錯誤機構の実装

試行錯誤機構は試行錯誤を実現するために，VM(仮想マシン) の SnapShot 機能を用いる．受講者には，次に示す VM を提供する．**防御用 VM:** 受講者が操作し，防御手法を実践することで，攻撃からサービスを守る．**攻撃用 VM:**防御用 VM に対し，シナリオファイルを基に攻撃を仕掛ける．

3節に示した要件を満たすために，試行錯誤機構を含むぷろてっくんを図 2 に示す構成で実装した．

Development of a Trial and Error Mechanism in a Security Exercise System Enabling Trial and Error for Linux Beginners
†1 Ichitoshi TAKEHARA, Mirei ISHIZUKA, Hitoshi KAMEI, Koji KIDA, Keizo SAISHO, Kagawa University

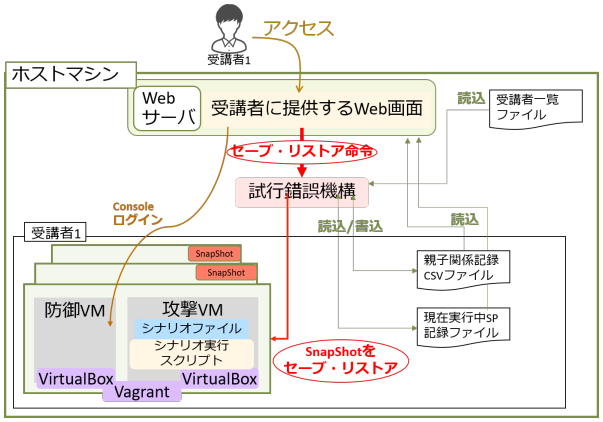

図 2: システム構成

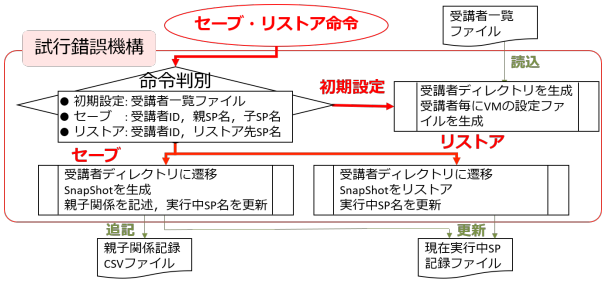

図 3: 試行錯誤機構の処理フロー

演習する際には，教授者が試行錯誤機構 (図 3) により，受講者 ID を記した受講者一覧ファイルを基に初期設定を行う．初期設定では，受講者 ID を基に受講者毎のユーザディレクトリを生成する．ディレクトリを分割し，その中で VM を管理することで，受講者間の影響を無関係にできる (要件②)．

受講者は，図 4 に示す Web 画面にて演習する．図 4-A にて防御用 VM に対して Console 操作が可能であり，防御手法を実践できる．

セーブ・リストア処理: 図 4-B，図 4-C にて，演習中の任意のタイミングでセーブ・リストアを実行できる (要件①)．

図 4-B にて，SP 名を入力し “save” ボタンを押下することで，必要なパラメータを併せて，試行錯誤機構にセーブ命令を送る．本機構のセーブ処理では，セーブする受講者 ID と親 SP 名と生成する子 SP 名を用いる．子 SP を基に VM の SnapShot 機能を用いてセーブし，親 SP を基に SP の親子関係を CSV ファイルに記録する (要件②)．セーブ時のファイルの変化を図 5 に示す．新たに子 SP を生成すると，子 SP が次回セーブするときの親 SP となる．このとき，子 SP 名を実行中 SP 記録ファイルに書き込むことにより，次回のセーブ処理の際に，試行錯誤機構がファイルを読み込むことで，受講者がセーブ毎に親 SP を指定せずとも，親子関係を維持して記録できる．

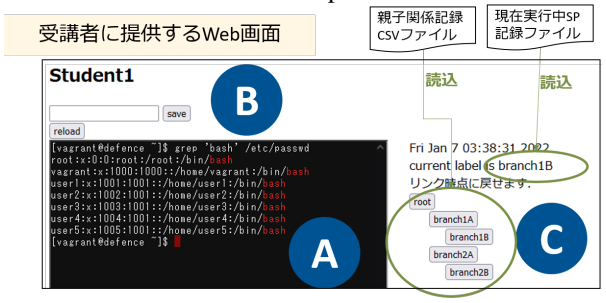

図 4: 受講者画面

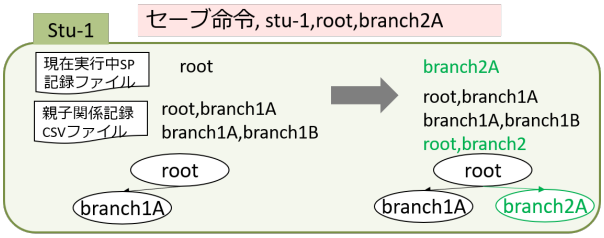

図 5: セーブ時のファイル変更

```
attack:
  10: 攻撃開始時間
    explain: rootアカウントのパスワードを奪取する
    command: ~/bin/attack.bash 攻撃コマンド
```

図 6: シナリオファイル

攻撃処理: 攻撃用 VM では，図 6 に示すシナリオファイルに従って攻撃するシナリオ実行スクリプトが動作する．1 分毎にシナリオファイルを読み込み，キーである攻撃開始時間を基にコマンドを実行し，攻撃する．攻撃開始時間は，防御用 VM の起動時からの稼働時間を基準とする．試行錯誤機構では，稼働時間も含めて SP に記録する．そのために，稼働時間もリストアし，同じタイミングで攻撃する (要件③)．

VM の構築や SnapShot の生成には，VM 構築ソフトウェア “Vagrant” と 仮想化ソフトウェア “VirtualBox” を用いる．Vagrant により，防御用 VM と攻撃用 VM を 1 セットで管理する．これにより，攻撃用 VM と防御用 VM を同時にセーブ・リストアできる．

今後は，実装した試行錯誤機構を用いたハードニング演習を行う予定である．

参考文献

1) 竹原一駿, 石塚美伶, 喜田弘司, 最所圭三, ”試行錯誤を可能とするセキュリティ演習システムの提案”, マルチメディア，分散協調とモバイルシンポジウム 2021 論文集, Vol.2021, No.1, pp.1473–1478(2021)