試行錯誤を可能とする
セキュリティ演習システムに関する研究

20G470　竹原 一駿 （最所研究室）

セキュリティ人材の育成方法として，攻撃からの防御を経験するハードニング演習がある．ハードニング演習を授業に導入した際の，育成の課題を解決するシステム「ぷろてっくん」について述べる．

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，サイバー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．ハードニング演習とは，実際のサービスの運営を模した演習システムで，セキュリティ対応チームの一員に成り切ってグループで攻撃からサービスを守る演習である．1 グループは 5 人程度で，OS の操作，Web サービスの操作など役割分担した上で，リーダの指示に従って防御する．

香川大学創造工学部情報システム･セキュリティコース 3 年次開講の授業「情報セキュリティ演習」でも，ハードニング演習を取り入れている．本授業の受講者は，Linux コマンドの使い方やセキュリティに関する座学は習得しているが，実際にセキュリティ対応の経験などはない，セキュリティ初学者である．授業にて受講者を観察した経験から，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ことがわかった．

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能について述べる．

2　試行錯誤機能を用いた演習の想定

初学者がセキュリティ演習を行う上で課題となるのは，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である．本論で提案する試行錯誤機能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，

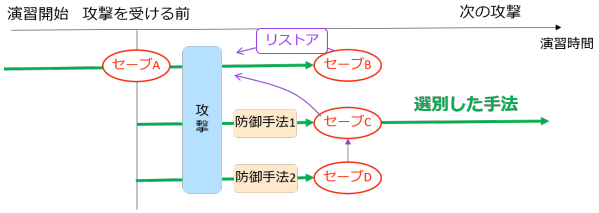

図 1: 試行錯誤のイメージ

試行錯誤機能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き，放置した場合を観察し，セーブする (セーブ B)．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．防御手法 1 を実践し，攻撃を防御できているか確認し，セーブする (セーブ C)．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 D)．また，敢えて更に脆弱な状態にし，攻撃を受けた際の影響を，検証することもできる．これらのセーブポイントは，攻撃と防御を繰り返す上で木構造で構成される．

3　試行錯誤機能の要件

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にセーブ・リストアを行う．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 を管理できる機能が必要である．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　試行錯誤機能の実装

試行錯誤機能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

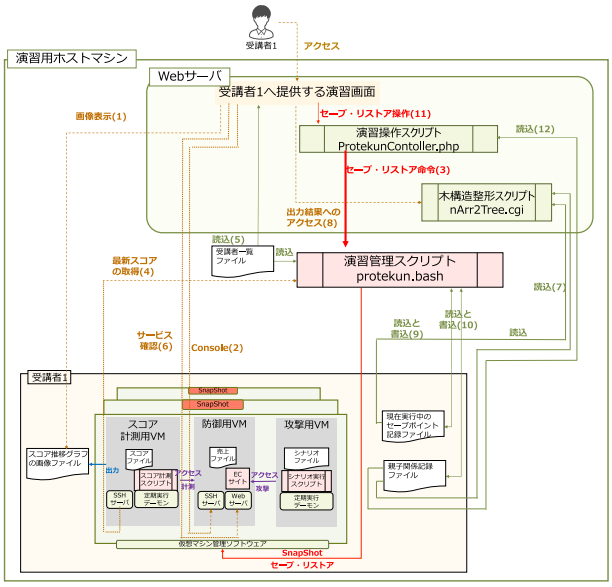

図 2: システム構成

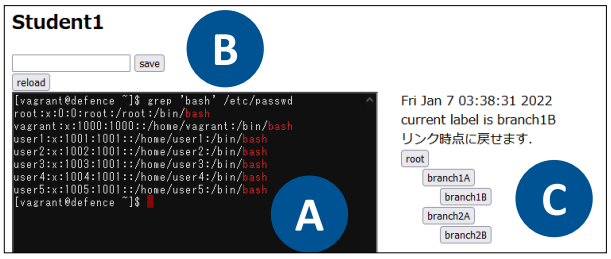

図 3: 受講者画面

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，試行錯誤機能を含むぷろてっくんを図 2 に示す構成で実装した．

演習する際には，教授者が，受講者 ID を記した受講者一覧ファイルを基に作成する．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．生成したディレクトリで 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 3 に示す Web 画面にて演習する．図 3-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 3-B，図 3-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 3-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，試行錯誤機能にセーブ命令を送る．本機能のセーブ処理では，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 名を基に VM の Snapshot 機能を用いてセーブし，親 SP 名を基に 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 4 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，試行錯誤機能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．

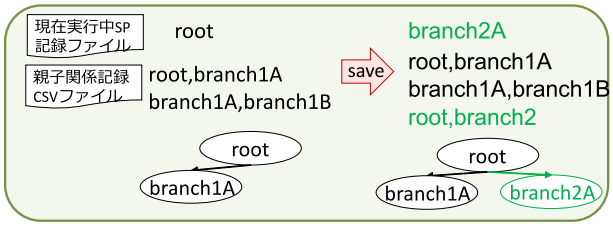

図 4: セーブ時のファイル変更

```
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
    execdir: /home/vagrant/
    command: ~/bin/attack > password.txt 攻撃コマンド
  11:
    explain: 10で得たパスワードを用いてバックドアを送りつける
    execdir: /home/vagrant/
    command: ~/bin/sendfile root password.txt backdoor.sh
  12:
    explain: Webサーバを停止する                    1ステップ
    execdir: /home/vagrant/
    command: ~/bin/sshcmd root password.txt 'systemctl stop httpd'
```

図 5: シナリオファイル

図 3-C の，リストアしたい SP 名のボタンを押下することで，試行錯誤機能にリストア命令を送る．図 3-C は，セーブ時に親子関係を記録した CSV ファイルを読み込むことで，これまでの SP を木構造で表示している．本機能のリストア処理では，VM の Snapshot 機能より，指定された SP へリストアし，実行中 SP 記録ファイルに SP 名を書き込む．

攻撃処理: 攻撃用 VM では，図 5 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．試行錯誤機能では，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．

参考文献

[1] 竹原一駿, 石塚美伶, 亀井仁志, 喜田弘司, 最所圭三. “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能の開発”, 第 84 回情報処理学会全国大会講演論文集, pp.XX-XX, 2021(発表予定)