Vol.2020-DPS-182 No.50
Vol.2020-CSEC-88 No.50
2020/3/13

# 深層強化学習によるPost-Exploitationの自動化

前田 龍星[1]　三村 守[1]

**概要**：情報システムのリスクを評価するためには，Exploit 成功後の行動（Post-Exploitation）を想定した調査が重要である．しかしながら，その調査には専門家が必要であり，知り得る限りではこれを自動化するツールも存在しない．本研究では，Post-Exploitation フレームワークとして著名な PowerShell Empire と強化学習を組み合わせることによって，Post-Exploitation を自動化する手法を提案する．PowerShell Empire に実装されているモジュールを学習エージェントの行動，エージェントが資格情報を窃取したか，侵害したアカウントは特権ユーザか，などの 10 個の情報をエージェントの状態として定義を行った．学習段階では，A2C，Q-Learning，SARSA の 3 つの強化学習エージェントの学習進行状況を比較し，A2C アルゴリズムの学習効率の優位性を確認した．テスト用ドメインネットワークで各エージェントを実行し，A2C アルゴリズムを用いた強化学習エージェントが，ドメインコントローラの管理者権限を取得できることを確認した．

# Automating Post-Exploitation with Deep Reinforcement Learning

***Abstract:*** In order to assess the risk of information systems, it is important to investigate the behavior of the attacker after successful exploitation (post-exploitation). However, the audit requires the experts, and to the best of our knowledge, there are no solutions to automate this process. This paper proposes a method of automating post-exploitation by combining deep reinforcement learning and the PowerShell Empire, which is famous as a post-exploitation framework. Our reinforcement learning agents select one of the PowerShell Empire modules as an action. The state of the agents is defined by 10 parameters such as type of account that was compromised by the agents. In the learning phase, we compared the learning progress of the 3 reinforcement learning models: A2C, Q-Learning, and SARSA. The result shows that the A2C could gain reward most efficiently. Moreover, the behavior of the trained agents are evaluated in a test domain network. The results show that the trained agent using A2C could obtain the administrative privileges to the domain controller.

## 1. はじめに

情報通信技術の発達に伴い，情報セキュリティ技術が注目されている．セキュリティ評価の手法に，*Penetration Tests* と *Red Team Tests* がある．*Penetration Tests* では，ネットワーク，アプリケーション，ハードウェア等の系統的な脆弱性スキャンを実行する．しかしながら，実際の攻撃者は，Exploit 成功後の行動（Post-Exploitation）では，*Penetration Tests* で行われるような脆弱性スキャンは実施しない．そのため，総合的なセキュリティ評価には *Red Team Tests* が必要になる．*Red Team Tests* では，攻撃者役となる *Red Team* が，侵入から，キャンペーンに応じた Post-Exploitation の実行までを行う．Post-Exploitation は，初期潜入後のコンピュータ間の移動，権限昇格，情報収集，バックドアの作成等の行動で構成されている．*Penetration Tests* をサポート，自動化するツールや脆弱性管理プログラム等は多く存在するが，Post-Exploitation の自動化を行うものは，われわれの知りうる限りでは存在しない．セキュリティ分野のソリューションを自動化する主要な手法として，機械学習を活用するものがある．機械学習の内，教師あり学習，教師なし学習は侵入検知やマルウェア検知，プライバシー保護システム等に活用されている [1, 2]．これらの学習手法は，学習に必要なデータセットを準備が，自動化の前提となる．しかしながら，Post-Exploitation に対応したデータセットは我々の知り得る限り存在しない．機械学習の内，強化学習は未知の環境の探索，経験の蓄積から学習を行う．学習後のエージェ

[1] 防衛大学校 情報工学科
Dept. Computer Science, National Defense Academy, Yokosuka, Kanagawa 239–0811, Japan

ントは連続した最適な行動を行うことができるようにモデル化され，リアルタイムで複雑な環境に対しても適用できる．いくつかの先行研究では，サイバーセキュリティシミュレーションゲームに，マルチエージェント強化学習を適用し，学習アルゴリズムの提案や有用性の評価を行っている [3, 4]．これらの研究成果を現実のサイバーセキュリティ環境へ適用するためには，学習環境や条件に具体性を与える必要がある．しかしながら，実環境で具体性を付与した多量の学習環境の構築は現実的とは言い難い．

本研究は，Data Augmentation [5,6] の発想を応用して，学習に用いる環境の内部ネットワーク構成にある程度の多様性を与えることで学習環境の不足に対処する．学習後の強化学習エージェントは，実環境において効率的な Post-Exploitation を自動で行うことができる状態を目標として設計を行う．本論文の貢献は以下のとおりである．

- Post-Exploitation への深層強化学習の適用の可否と有用性の検証
- Post-Exploitation の自動化

本稿は次のように構成されている．第 2 章では強化学習を用いたセキュリティ技術の概要について示す．第 3 章では本研究で用いた強化学習のモデルの概要，第 4 章では提案する手法の詳細を示す．第 5 章では，エージェントの学習と学習後に実環境での検証を行い，その結果を示す．第 6 章では結果を基に本研究の考察を行い，最後に，第 7 章では結論を示す．

## 2. 関連研究

### 2.1 Lateral Movement

本稿では，Post-Exploitation を構成する攻撃者の動作の内，Lateral Movement の自動化に焦点を当てる（第 4.4 章，第 5 章参照）．Advanced Persistence Threat (APT) の攻撃者は，初期潜入に成功した端末を足掛けとして，内部ネットワークの深部に存在する重要なシステムへのアクセスを試みる．こうした内部ネットワークにおける漸進的な移動を Lateral Movement と呼ぶ．多くのネットワーク型，ホスト型ソリューション（Windows Defender，McAfee，Norton，Snort，OS-SEC など）が Lateral Movement を検知，除去するように設計されている．Tian らは，エッジクラウドコンピューティング環境において効率的に，リアルタイムに Lateral Movement を検知できる手法を提案した [7]．また，Lah らは，パターンリスクスコアリングに基づいた Lateral Movement の検知改善フレームワークを提案している [8]．その一方で，Lateral Movement の手法も多様化している．例えば，SMB や RDP サービスを悪用するもの，資格情報を悪用するもの（例えば，credential dumping，pass-the-ticket [9]），既存のクライアントの通信を再利用するもの（例えば，SSH ハイジャック）がある．その他にも，Niakanlahiji らが権限昇格や移動先への新しいコネクション確立を必要としない，ステルス性の高い Lateral Movement を提案している [10]．Lateral Movement を行うには，権限昇格や認証の成功など予備動作や調査を行う必要があるため，高いスキルが必要である．本稿では，この高いスキルを要する Post-Exploitation の一つである Lateral Movement を深層強化学習によって再現し，自動化することを研究の目的とする．Lateral Movement の単純な自動化（総当たり試行など）は，検知を回避するための実際の攻撃者の行動と対応しない．エージェントが経験を基に学習することで，効率的で現実的な Lateral Movement の自動化手法を提案する．

### 2.2 Penetratinon Tests への強化学習の適用

Ghanem らは，強化学習を *Penetration Tests* のタスクに適用することで効率化を図る手法を提案し，効率化に有効であることを示した [11]．しかしながら，Ghanem らの研究はシミュレーションにおける評価の段階にとどまっており，実環境における実装には至っていない．

DeepExploit [12] は，Metasploit と強化学習エージェントと連携させることで，対象ホストに対する探索行為から侵入テストまで，実環境での自動化を実現している．これらのフレームワークは，目的を，サーバやソフトウェアの脆弱性診断の効率化や自動化に限定しており，Post-Exploitation の自動化や効率化を想定していない．一方で本研究の目的は，Post-Exploitation の効率化および自動化である．初期潜入タスクの自動化においては，強化学習エージェントに状態遷移が生じない．Post-Exploitation タスクの自動化は，状態遷移を定義する必要がある．これは，強化学習を双方のタスクに適用する際にも大きな相違点となる．DeepExploit を例にとってみると，その強化学習エージェントの状態はターゲットサーバのポート開放状況によって定義される．したがって，exploit の実行後にも状態遷移は発生しえない．一方で，Post-Exploitation タスクは連続した状態におけるプロセスとなるため，エージェントの状態遷移が定義される．本研究では，強化学習エージェントを連続した状態に適用する．

### 2.3 サイバーセキュリティシミュレーション

Elderman らの研究 [4] は，ネットワークを想定したサイバーシミュレーションゲームに焦点を当てている．このゲームは攻撃者と防御者の 2 つのエージェントでプレイされる，敵対的，連続的な意思決定問題となっている．2 つのエージェントは，たがいに敵対的学習を行い，強化学習モデルごとの有用性を評価した．

しかしながら，現実の攻撃者と防御者の対戦では，攻撃者と防御者のリアルタイムな攻防が行われることは少ない．統計によるとサイバー攻撃の内，62%が被害を及ぼした後，つまり，攻撃者が目的を達成した後に発覚する [13]．

本研究では，攻撃者と防御者のリアルタイムな攻防が行われることが少ない現状を考慮して，防御エージェントを明示的に設定しない環境で攻撃エージェントの学習を行う．Richard らの研究ではネットワークは単純化され，攻撃者と防御者が移動できるノードによって構成されている．攻撃者の選択できる行動は，攻撃値を持つ選択肢として抽象化され，その成否は防御側の持つ防御値によって決定される．本研究ではネットワーク構造と攻撃者の選択できる行動，状態の定義をより具体化することで，エージェントは現実の環境でも動作可能な設計を実施する．

## 3. 強化学習

本研究では Post-Exploitation の自動化のために強化学習を用いる．本章では，本研究で使用する深層強化学習アルゴリズム（A2C）の概要について説明する．

### 3.1 強化学習の種類

Q-Learning に代表される Value Base のアルゴリズムは，最適な行動価値関数を推定することに主眼を置き，その学習エージェントは環境から得られる経験を，価値評価の更新に利用する．一方で，State-Action-Reward-State-Action(SARSA) に代表される Policy Base のアルゴリズムは，現在の方策を改善することに主眼を置き，その学習エージェントは経験を戦略の更新に利用する．また，Value Base と Policy Base を組み合わせた手法として，Actor-Critic 法がある．Actor-Critic 法では，戦略と価値評価を相互に更新して学習を進める．一般的に，Actor-Critic 法は，学習にかかる時間は長くなるが，安定した学習が期待できる．

### 3.2 A2C

A2C（Advantage Actor Critic）は，A3C（Asynchronous Advantage Actor Critic）の，分散学習 (Asynchronous) の部分を除外した手法である．A3C は，Mnih らによって提案された非同期型分散学習アルゴリズムである [14]．どちらも Actor-Critic の枠組みで，Advantage をもとに学習を行うアルゴリズムである．Advantage は，状態に関係ない，純粋な行動の価値を表す．これを用いることによって，状態の価値を差し引いたうえで，行動を評価することが可能であり，学習の安定化を図ることができる．A2C は，A3C の分散学習部分を取り除いているが，経験の分散収集はそのまま行う．経験を分散収集し，一元的に学習を行うことで，A3C と同等，またはそれ以上の性能を示している [15]．本研究では，A2C の他，Value Base を代表する Q-Learning，Policy Base を代表する SARSA を実装し，学習効率，学習結果の比較評価を行う．

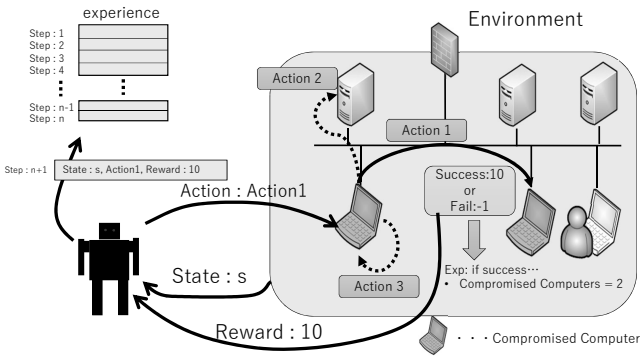


**図 1** 提案手法の概要

## 4. 提案手法

### 4.1 概要

この章では，提案手法の詳細を強化学習エージェントの学習に必要な要素ごとに解説する．まず，強化学習アルゴリズムを Post-Exploitation に適用するために，エージェントが選択できる行動群 $A$ と状態 $s$ を定義する．エージェントが選択できる行動群 $A$ として，PowerShell Empire に登録されているモジュールを設定する．$s$ は，エージェントが観測可能かつ状況判断に有効な 10 項目により定義する．学習環境は，エージェントの状態 $s$ を定義できる環境として実装を行う．

**図 1** に，提案手法の概要を示す．学習エージェントは，学習環境から観測される状態 $s$，エージェントが選択する $a$ とその実行結果に応じた報酬 $r$ を経験として蓄積して学習する．提案手法は，強化学習のモデルとして A2C を用いるため，図 1 の Environment は，分散環境として構築される．各分散環境においてエージェントが経験を分散収集することによって，学習効率を高めることができる．4.2 では，図 1 の State の詳細を，4.3 では，Action 部分の詳細を解説する．4.4 では，図 1 の Reward 部分である，学習エージェントに与える報酬設定の詳細を示し，最後に，4.5 では，これら提案手法の実装について解説する．

### 4.2 状態の定義

学習環境，実環境においてエージェントの状態 $s$ は，10 個のエントリで定義する．10 個のエントリ名とその概要を**表 1** に示す．これらの情報は，実環境において攻撃者が観測可能な情報である．攻撃者の持つ情報を再現し，現実的な攻撃の再現を可能にする．

Computers Found と Compromised Computer の値は表表 1 に示されている数値で表現され，Previous Module の値は，選択される行動ごとの固有の値，その他のエントリはそれぞれ 0 か 1 で表現される．Admin Access は，エージェントが侵害している現在のユーザ権限でローカルアドミンユーザとしてアクセスできるコンピュータの有無

表 1　状態の定義

| エントリ名 | 概要 |
|---|---|
| Computers Found | 発見したコンピュータの数 |
| Admin Access | Local Admin Access の有無 |
| Compromised Computers | 侵害済みコンピュータの数 |
| Previous Module | 前ステップで選択した行動 |
| Admin | 一般ユーザか，その他か |
| User Name | ユーザ名の取得有無 |
| Password | 平文の資格情報を取得有無 |
| Hash | ハッシュ化された資格情報の取得有無 |
| Rhost | 脆弱なホストの有無 |
| DCOM | 動作中の DCOM Application の有無 |

を表している．アクセス可能なコンピュータが存在する場合，容易に任意のシェルやプログラムの実行が行えるため，積極的にこの情報は活用されるべきである．User Name，Password，Hash といった資格情報がネットワークトラフィック，侵害済みの端末から発見された場合，新たにコンピュータや情報を侵害するために有効である．Rhost は，脆弱性を持ったシステムの有無を表す．DCOM（Distributed Component Object Model）は，異なるリモートコンピュータ上のソフトウェアコンポーネント間で通信するための Windows の機能である．侵害済みのアカウントを利用したアクセスができない，もしくは資格情報を利用したトークンでシェルが実行できない場合，DCOM を利用することで様々なシェルを作成，実行することができる．このような理由から，DCOM アプリケーションの動作の有無はエージェントの状況判断に大きく影響を与えるため，状態を定義するエントリの一つとして加えた．

### 4.3　行動の定義

実環境における動作のために，強化学習エージェントの行動として，Post-Exploitation で実際に行われる行動を定義する必要がある．近年，Post-Exploitation において，PowerShell が用いられることが多い [16]．この理由から，エージェントが実環境でも選択可能な行動群 $A$ として，PowerShell Empire [17] に実装されているモジュール群（204 個）を設定する．PowerShell Empire は，PowerShell を用いた Post-Exploitation フレームワークとして実装されたツールである．$A$ は，モジュールの目的ごとに 12 個のグループに分類が可能である．**表 2** にグループ名と所属するモジュールの数を示す．同じグループに属すモジュールは，手段や仕組みは異なるが，同じ目的を達成するために用いられる．例えば，画面のキャプチャをとるモジュールとキーロガーの設置モジュールは動作は異なるが，どちらも情報収集の目的で用いられるため同じグループに属する．

表 2　PowerShell Empire のモジュールの分類

| グループ名 | 登録数 | グループ名 | 登録数 |
|---|---|---|---|
| Code Execution | 6 | Management | 30 |
| Collection | 24 | Persistence | 18 |
| Credentials | 23 | Privesc | 22 |
| Exfiltration | 2 | Recon | 3 |
| Exploitation | 3 | Situational Awareness | 52 |
| Lateral Movement | 12 | Trollsploit | 9 |

Collection は，先に示したキーロガー等の手法による情報収集を行うモジュール群であり，Credential は，Mimikatz 等を利用した資格情報，トークンの収集ができるモジュール群である．Lateral Movement は，WMI（Windows Management Instruction）や PS Command（PowerShell Session Command），DCOM 等を用いて，別のコンピュータへの移動を行うモジュール群である．ここでは，簡単に 3 つのグループの説明をしたが，より詳しい情報は PowerShell Empire のプロジェクト公式ページで確認できる [17]．

### 4.4　報酬の設定

この項では，報酬設定の詳細を説明する．4.3 で説明したように，モジュールは目的別に分類されているため，エージェントの目標に合わせて報酬設定を行う事ができる．

評価実験では，評価を容易にするため，エージェントの最終目標をドメインコントローラの管理者権限の取得とする．多くの場合，最初に侵害されるコンピュータはドメインコントローラのような価値のあるシステムではない．この場合，攻撃者は，価値のあるシステムを発見，またはアクセスできる権限を取得するために Lateral Movement を行う．したがって，本キャンペーンでは Lateral Movement のグループに分類されるモジュールがキーアクションとなる．学習時のエージェントへの報酬 $r$ は，Lateral Movement の成功に対して与えるよう設定した．また，Lateral Movement 成功時に新たなアカウントの制御を取得できているかどうかで $r$ の大きさに差をつける．例えば，WMI を利用してコンピュータ間での移動が成功したとしても，より高い権限を持つアカウントの制御を取得できていなければ，ネットワーク内で行える行動の種類は同じである．より高い権限を持つアカウントの制御を取得した場合，価値の高い資産へアクセスできる確率が上がるため，成功の価値が高い．このような理由から，Lateral Movement が成功した場合に，$r$ は，その成功の価値の大きさによって，価値が小さければ $r$=10，価値が大きければ $r$=50 とし，その他のモジュールは成功の可否にかかわらず $r$=-1 に設定した．$r$ が負の値の時，$r$ はエージェントへの罰を意味する．つまり，Lateral Movement 以外の行動の報酬を $r$=-1 に設定することで，エージェントは環境内で可能な限り早くゴールを目指す．これは，実際の攻撃者の，侵害の露見を避けるため，目標を可能な限り早く達成する状況に対応する．

$r$=50 の Lateral Movement が成功したとき 1 エピソード終了となる．

### 4.5 実装

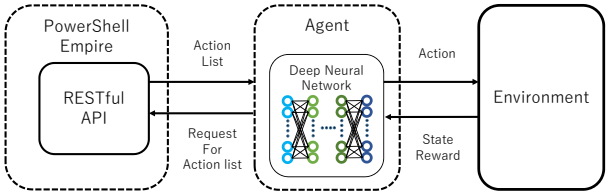


**図 2** 実装の概要

A2C の構成は，中間層 3 層，入力層と出力層を合わせて全体で 5 層のディープニューラルネットワークとした．状態 $s$ を入力値とし，選択できる行動 $a$ のそれぞれの選択確率 $p(a)$，つまり行動群 $A$ の確率分布と状態価値 $v(s)$ が出力される．エージェントの学習時における行動選択は，$\varepsilon$ グリーディ法に基づき，状態 $s$ において，$p(a)$ が最大となる $a$，またはランダムに決定される $a$ を実行する．$\varepsilon$ の値は，学習の開始時 0.5 に設定し，学習の進行に比例して 0 に近づくように設定した．図 2 に実装の概要を示す．エージェントと PowerShell Empire の通信には，PowerShell Empire に実装されている RESTful API を利用した．状態の定義，RESTful API との通信，報酬の設定を含めた強化学習モデルは，Python3.6 を用いて実装した．

## 5. 評価

### 5.1 学習段階

エージェントの学習を行うための環境を実環境で大量に構築することは，現実的には難しい．これらの問題に対処するために，ネットワークのネットワーク構成や攻撃に対する弱点が偏らず，一般性を持つように学習環境を設計した．具体的には Data Augmentation [5,6] の発想を応用し，ベースとなるネットワーク設定にノイズを加えることで，環境に多様性を持たせた．

学習段階で習得すべきタスクは複数考えられるが、ここではドメインコントローラの管理者権限を取得とした。キーアクションとなる Lateral Movement に成功確率を定めることで，ネットワーク設定や脆弱性が一意に定まらない環境となる．成功確率の設定がノイズの役割を果たす．適切な学習が行える Lateral Movement の成功確率を設定するために，確率の設定を 5 パターン試行した．成功確率が全て 50 %である場合（A），全て 20 %である場合（B），全て 80 %である場合（C），全てランダムに設定される場合（D），モジュールの特性に応じて成功確率をそれぞれ一定範囲に定めた場合（E）の 5 パターンである．**表 3** に，モジュールの特性に応じて実行時の成功確率を定めた場合（E）の，各モジュールの成功確率を示す．

**表 3** 手法 E における Lateral Movement を行う各モジュールの成功確率（%）

| モジュール名 | 確率 | モジュール名 | 確率 |
|---|---|---|---|
| invoke wmi debugger | 40-60 | invoke wmi | 70-90 |
| invoke sshcommand | 40-60 | invoke psexec | 40-60 |
| invoke psremoting | 10-30 | invoke smbexec | 10-30 |
| jenkins script console | 10-30 | invoke dcom | 70-90 |
| invoke executemsbuild | 40-60 | inveigh relay | 40-60 |
| new gpo immediate task | 10-30 | invoke sqloscmd | 10-30 |

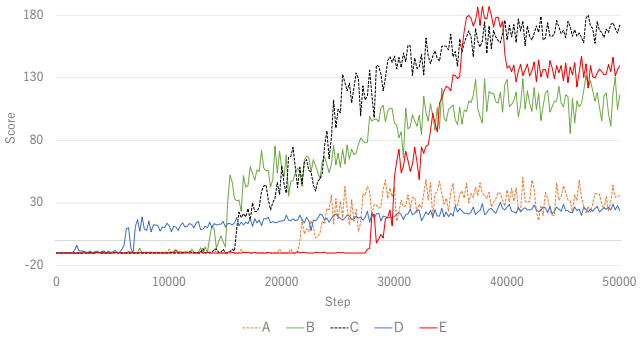


**図 3** Lateral Movement を行う各モジュールの成功確率の定め方による学習の進行状況の比較．直近 200step の獲得報酬から 10step 当たりの平均獲得報酬を計算．

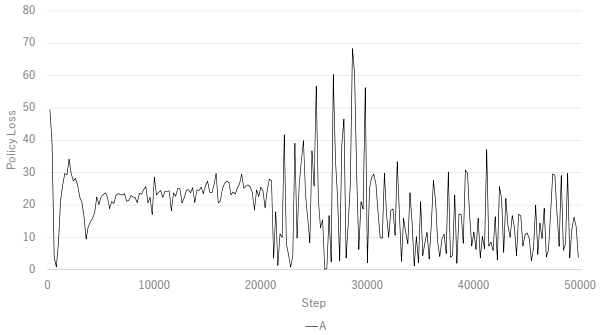


**図 4** 手法 A の場合の学習時の損失関数の推移

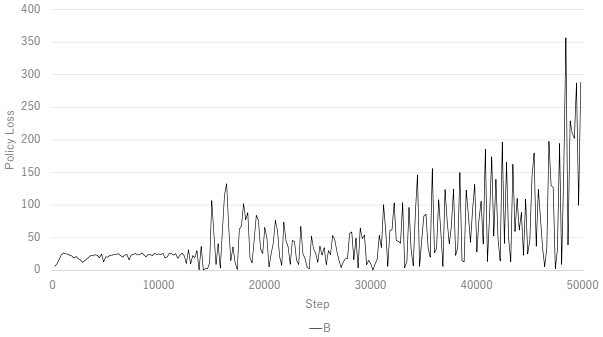


**図 5** 手法 B の場合の学習時の損失関数の推移

**図 3** は，Lateral Movement の成功確率設定ごとの獲得報酬の推移を表すグラフである．強化学習のモデルは全て A2C である．横軸は，エージェントがモジュール（行動 $a$）

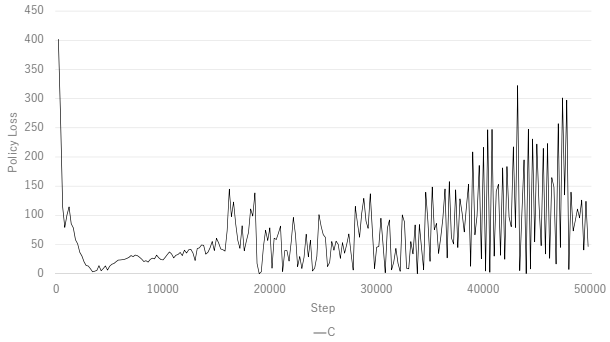


図 6　手法 C の場合の学習時の損失関数の推移

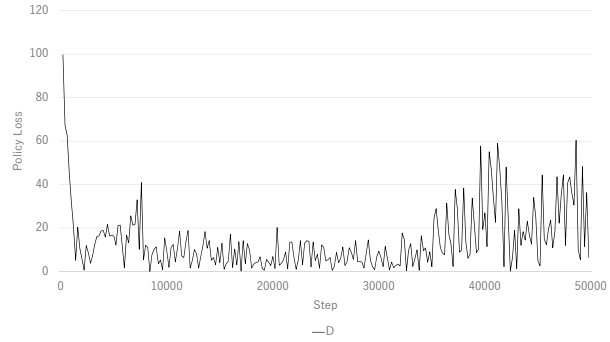


図 7　手法 D の場合の学習時の損失関数の推移

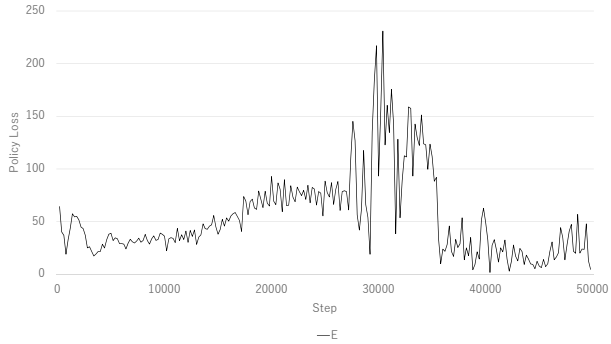


図 8　手法 E の学習時の損失関数の推移

を実行した回数を表す．学習は全ての手法で 50,000Step 行った．獲得報酬の合計の収束値から，A，B，E は，価値の大きな Lateral Movement までの行動を学習できていることが分かる．**図 4** から**図 8** は，各手法の学習時の損失関数の推移を表すグラフである．B，C，D のグラフは変動が激しく，学習終了間際でも 0 への収束の傾向はみられない．対して，A と E は，獲得報酬が顕著に増加した step 数で損失関数の値が大きくなるものの，獲得報酬の安定とともに損失関数のグラフも 0 に収束していく傾向がみられる．

次に，強化学習のモデルごとの学習状況の評価を行う．Q-Learning と SARSA を実装し，A2C の場合と同様の条件で学習を行った．A2C で大きな報酬を獲得した C と E をそれぞれ学習モデルのみ変更した．**図 9**，**図 10** から，A2C モデルは最終的な獲得報酬が他のモデルと比べて大きいことが分かった．SARSA は，Q-Learning に比べて，部分的に獲得報酬が大きくなる箇所があるが，学習が安

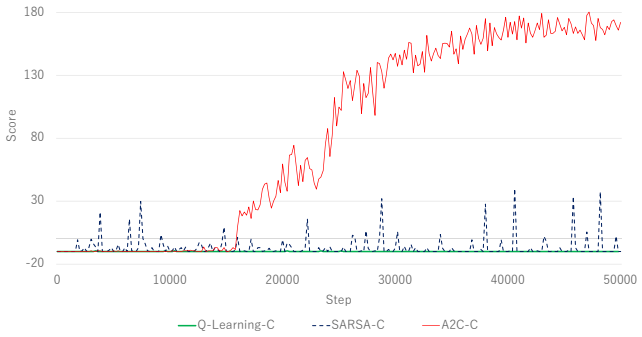


図 9　Lateral Movement の成功確率を全て 80 %に設定した場合の強化学習モデルごとの学習状況比較

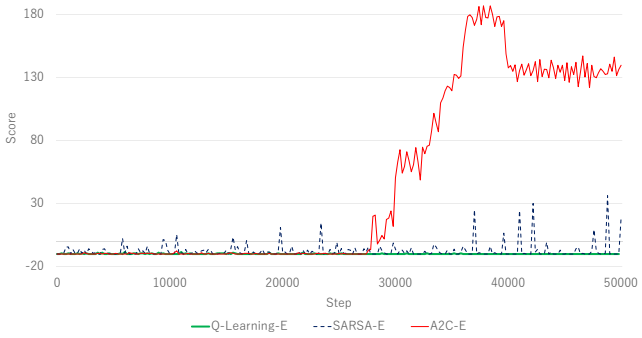


図 10　Lateral Movement の成功確率をモジュールの特性に応じて設定した場合の強化学習モデルごとの学習状況比較

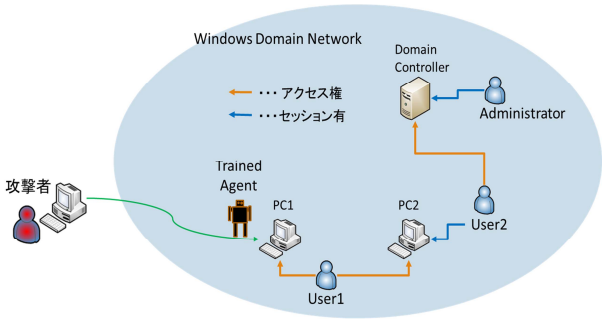


図 11　テスト用のネットワーク構成

定しなかった．Q-Learning は獲得報酬に大きな変化がなかった．

### 5.2　実環境における検証

学習済みのエージェントをテスト用に構築した Windows Domain Network にて実際に実行し，エージェントがドメインコントローラの管理者権限を取得できるか検証する．本実験では説明を容易にするため，**図 11** のように単純化された内部ネットワークでの評価を行う．エージェントは，ネットワーク内において目標から最も遠い位置（図 11 では PC 1）から行動を開始する．図 11 の構成において，エージェントがドメインコントローラーの管理者（Administrator）権限を取得するためには，端末間の移動を 2 回，ユーザ間での移動を 2 回行わなければならない．

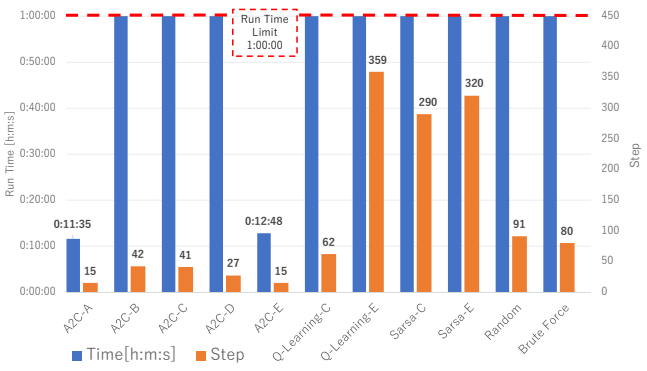


**図 12** 目標達成までの実行時間とステップ数比較．実行の制限時間は 1 時間を設定．

**図 12** に，図 11 のネットワークにおいてテストを行った際の結果を示す．Random は選択できる行動をランダムに選択した場合であり，BruteForce は選択できる行動を制限時間内に総当たりで実行した場合である．この 2 つの手法は，強化学習の適用の効果を評価する際の比較対象となる．A2C-A と A2C-E は，制限時間内に，15step で目標を達成することができている．A2C-A は，A2C-E と Step 数は同じだが，A2C-A の方が実行時間はやや短い．その他の手法は，BruteForce，Random を含め，制限時間内に目標を達成することができなかった．Random と BruteForce は，行動選択に偏りがないため，そのステップ数から 1step 当たりの平均実行時間を求めることができる．平均実行時間は 40〜45 秒間であった．しかしながら，Q-Learning-E と SARSA は，1 時間あたりに平均の 3〜4 倍程度の Step 数を記録している．これらは，適切な行動を学習することができていないため，実行時間の短い同一の行動を繰り返し実行していると考えられる．興味深い実験結果は，A2C-B，C の学習と実環境における検証結果である．学習段階では，A2C-A よりも A2C-B，C は獲得報酬が高いが，実環境において目標を達成したのは A2C-A である．

## 6. 議論

### 6.1 深層強化学習の有効性

本研究は，Post-Exploitation の自動化，Post-Exploitation への深層強化学習の適用の可否およびその有用性の検証を目的としている．5.1 では，Lateral Movement の手法 C および E の場合に，大きな報酬を獲得する学習を行えることを示した．加えて，A2C は Q-Learning，SARSA と比較した結果，Post-Exploitation タスクにおいて学習効率が良いことを示した．5.2 では，A2C-A および E が目標を達成できることを示した．この結果は，深層強化学習を Post-Exploitation に適用可能であること，自動化が可能であることを示唆するものである．また，Random と BruteForce と A2C-A，E を比較したとき，A2C-A と E は 15step で目標を達成しており，深層強化学習を適用した場合の有効性を検証することができた．

A2C-B，C は学習段階で高い報酬を獲得していたが，実環境では目標を達成しなかった．B と C は成功確率が 50 %，80 %であり，実行した Lateral Movement が常に高い確率で成功するため，最初に発見した報酬が獲得できる行動に行動選択が偏り，過学習の状態になったと考えられる．

### 6.2 比較

DeepExploit や Ghanem らの研究は [11, 12]，強化学習を利用して初期潜入タスクの自動化を試みた．初期潜入タスクでは，攻撃者は連続した状態行動空間の中で行動選択をする必要性は低い．一方で，Post-Exploitation タスクでは前の行動の結果を基に次の行動を選択する場面が多い．そのため，連続した状態行動空間において行動選択が重要である．DeepExploit を例に挙げる．DeepExploit で強化学習エージェントの状態はターゲットサーバのポートの開放状況によって定義され，その状態は Exploit の後も変化がない．Post-Exploitation では，エージェントは前の行動の結果に応じて次の行動を選択する必要がある．つまり，状態遷移が生じる．我々の手法は，こうした状態行動空間に深層強化学習を適用することで自動化を達成した．さらに，本研究では深層強化学習の現実的なサイバーセキュリティシナリオへの適用可能性を示した．いくつかの先行研究 [3,4] では，サイバーセキュリティシミュレーションゲームに MARL を適用する研究を行っている．先行研究との違いは，本研究は実環境においてエージェントが動作する前提で実装を行った点である．本研究では，シミュレーションから実環境に深層強化学習の適用先を変更することで，研究の実用性を向上させることに成功した．

### 6.3 研究倫理

攻撃技術の研究の成果は悪用されることを予防する措置が必要である．PowerShell Empire Project は，実際にサイバー攻撃にたびたび悪用されており，サポートを終了している．PowerShell Empire を利用した本研究の実装コードは悪用を防ぐため，公開を見送っている．また，具体的にエージェントが選択した行動への言及にも配慮している．しかしながら，本研究で提案した手法は，脅威への先行的な対処として有効である．攻撃者が実際に用いる可能性のある，現実的な手段を実環境においてシミュレートすることができる．本研究は，高いコストと高いスキルを持った人材が必要である *Red Team Tests* の実行をより容易なものにする可能性を示唆し，防御技術の向上に資するものであると考えられる．

### 6.4 制約

本研究には，いくつかの限界がある．まず，強化学習エージェントの学習環境のバリエーションが寡少であるこ

とである．本研究では，行動 $a$ の成功確率の設定の仕方にバリエーションを持たせることで，学習環境に多様性を持たせつつ，学習が適切に行える設定を模索したが，実際の環境では，成功確率の他に不確定要素が存在することが考えられる．今後，学習環境に多様性を持たせる別の手法を検討する必要がある．

また，エージェントの状態の定義は提案手法で示した 1 パターンのみで評価を行った．エージェントの状態の定義の仕方を変更した場合，学習の進行の仕方，実環境での挙動が変わる可能性が大きいため，別の定義の方法の模索と評価に，更なる研究の余地があると考えられる．

## 7. 結論

本稿では，深層強化学習の適用による，Post-Exploitation の自動化の可否の検証とその有効性を評価を行った．強化学習エージェントは実環境ネットワークにおいて実行可能な実装を行った．実験では，提案した強化学習モデルである A2C と，Q-Learning，SARSA，強化学習を用いないベーシックな手法との比較検証を行い，深層強化学習を用いた A2C の優位性を確認した．本研究は，実環境において，Post-Exploitation を強化学習によって自動化する，我々の知りえる限りで初めての研究である．

今後の研究の発展の方向性として以下 2 つを挙げる．1）学習エージェントの状態の別の定義手法の検討　2）学習環境に多様性を持たせる別の手法の検討

### 参考文献

[1] Jia, J. and Gong, N. Z.: AttriGuard: A Practical Defense Against Attribute Inference Attacks via Adversarial Machine Learning, *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018.*, pp. 513–529 (online), available from ⟨https://www.usenix.org/conference/usenixsecurity18/presentation/jia-jinyuan⟩ (2018).

[2] Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A. and Marchetti, M.: On the effectiveness of machine and deep learning for cyber security, *10th International Conference on Cyber Conflict, CyCon 2018, Tallinn, Estonia, May 29 - June 1, 2018*, pp. 371–390 (online), DOI: 10.23919/CYCON.2018.8405026 (2018).

[3] He, X., Dai, H. and Ning, P.: Faster Learning and Adaptation in Security Games by Exploiting Information Asymmetry, *IEEE Trans. Signal Processing*, Vol. 64, No. 13, pp. 3429–3443 (online), DOI: 10.1109/TSP.2016.2548987 (2016).

[4] Elderman, R., Pater, L. J. J., Thie, A. S., Drugan, M. M. and Wiering, M.: Adversarial Reinforcement Learning in a Cyber Security Simulation, *Proceedings of the 9th International Conference on Agents and Artificial Intelligence, ICAART 2017, Volume 2, Porto, Portugal, February 24-26, 2017.* (van den Herik, H. J., Rocha, A. P. and Filipe, J., eds.), SciTePress, pp. 559–566 (online), DOI: 10.5220/0006197105590566 (2017).

[5] Simard, P. Y., LeCun, Y., Denker, J. S. and Victorri, B.: Transformation invariance in pattern recognition: Tangent distance and propagation, *Int. J. Imaging Systems and Technology*, Vol. 11, No. 3, pp. 181–197 (2000).

[6] Leen, T. K., Dietterich, T. G. and Tresp, V.(eds.): *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, MIT Press (2001).

[7] Tian, Z., Shi, W., Wang, Y., Zhu, C., Du, X., Su, S., Sun, Y. and Guizani, N.: Real-Time Lateral Movement Detection Based on Evidence Reasoning Network for Edge Computing Environment, *IEEE Trans. Industrial Informatics*, Vol. 15, No. 7, pp. 4285–4294 (online), DOI: 10.1109/TII.2019.2907754 (2019).

[8] Lah, A. A. A., Dziyauddin, R. A. and Azmi, M. H.: Proposed Framework for Network Lateral Movement Detection Based On User Risk Scoring in SIEM, *2018 2nd International Conference on Telematics and Future Generation Networks (TAFGEN)*, pp. 149–154 (online), DOI: 10.1109/TAFGEN.2018.8580484 (2018).

[9] Duckwall, S. and Campbell, C.: Hello, My name is MICROSOFT and I have acredential problem In Blackhat USA 2013 White Papers 2013, https://media.blackhat.com/us-13/US-13-Duckwall-Pass-the-Hash-WP.pdf.

[10] : ShadowMove: A Stealthy Lateral Movement Strategy, *29th USENIX Security Symposium (USENIX Security 20)*, Boston, MA, USENIX Association, (online), available from ⟨https://www.usenix.org/conference/usenixsecurity20/presentation/niakanlahiji⟩ (2020).

[11] Ghanem, M. C. and Chen, T. M.: Reinforcement Learning for Intelligent Penetration Testing, *2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pp. 185–192 (online), DOI: 10.1109/WorldS4.2018.8611595 (2018).

[12] Isao, T.: Metasploit Meets Machine Learning, https://www.mbsd.jp/blog/20180228.html.

[13] Sharma, A., Kalbarczyk, Z., Barlow, J. and Iyer, R. K.: Analysis of security data from a large computing organization, *Proceedings of the 2011 IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2011, Hong Kong, China, June 27-30 2011*, pp. 506–517 (online), DOI: 10.1109/DSN.2011.5958263 (2011).

[14] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D. and Kavukcuoglu, K.: Asynchronous Methods for Deep Reinforcement Learning, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pp. 1928–1937 (online), available from ⟨http://proceedings.mlr.press/v48/mniha16.html⟩ (2016).

[15] Wu, Y., Mansimov, E., Liao, S., Radford, A. and Schulman, J.: OpenAI Baselines:ACKTR & A2C, https://openai.com/blog/baselines-acktr-a2c/.

[16] Wueest, C.: THE INCREASED USE OF POWERSHELL IN ATTACKS, https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/increased-use-of-powershell-in-attacks-16-en.pdf.

[17] : PowerShell Empire — Building an Empire with PowerShell, https://www.powershellempire.com/.