自律サイバー推論システムにおけるBERTモデル導入による状態推定の強化

米田 智紀[1,a)]　大塚 玲[1]

概要：ペネトレーションテストは機器やシステムに対して様々な技術を駆使して侵入を試みることで，対象のセキュリティ上の脆弱性を検査する手法であり，特に機械学習ベースの自律的ペネトレーションテスト技術は，offensive security を実現する重要な手法として，ますます増加，巧妙化するサイバー攻撃への対応策になると目されている．既に Deepexploit[10] 等，様々な機械学習ベースの自律的ペネトレーションテストツールが生み出されている．中でも，訓練データを予め準備しなくても自律的に攻撃手法を獲得できる強化学習によるペネトレーションテスタが注目を集めている．本研究では，従来から多くの提案があるマルコフ決定過程 (MDP) に基づく強化学習モデル [5] ではなく，ペネトレーションテストの過程で得られるシステムレスポンスをニューラル自然言語処理技術により解釈して状態を推定しながら次の最適攻撃行動を推定する部分観測マルコフ決定過程に基づく強化学習モデルに注目している．2020 年に発表された LeDeepChef[3] は，テキストベースのダンジョンゲームである Textworld[4] を部分観測マルコフ決定過程 (POMDP) に基づく強化学習で効率的にゴールを見いだすニューラルエージェントを提案している．
本論文では，ニューラルエージェントの状態推定を行うモデルに SecBERT[11] を導入したシステムを提案する．本システムを Windows/Linux 等の OS 環境に作用させ，exploit コマンドを行動集合に持たせ実験を行い，従来の GRU モデルに対する優位性を示す．

キーワード：深層強化学習，ペネトレーションテスト，部分観測マルコフ決定過程，自律サイバー推論システム，BERT

Enhanced State Estimation by Introducing BERT Models in Autonomous Cyber Reasoning System

Tomonori Yoneda[1,a)]　Akira Otsuka[1]

Abstract: Penetration testing is a method of testing for security vulnerabilities by attempting to penetrate devices and systems using various techniques. In particular, autonomous penetration testing technology based on machine learning is important for achieving offensive security. The autonomous penetration testing technology based on machine learning is expected to become an important method to realize offensive security and to cope with the increasing and sophisticated cyber attacks. In particular, autonomous penetration testing based on machine learning is an important method to realize offensive security and cope with cyber attacks' growing number and sophistication. Various machine learning-based autonomous penetration testing tools have already been developed, such as Deepexploit[10]. In particular, penetration testers based on reinforcement learning[5], which can autonomously acquire attack methods without preparing training data in advance, are attracting attention. This paper focuses on a reinforcement learning model based on a partially observed Markov decision process. The system response obtained in the penetration testing process is interpreted by neural natural language processing techniques to estimate the state of the system and the subsequent optimal attack behavior. LeDeepChef[3], published in 2020, proposes a neural agent that can efficiently find the goal of Textworld[4], a text-based dungeon game, by reinforcement learning based on the Partial Observation Markov Decision Process (POMDP). In this paper, we propose a system that introduces SecBERT[11] into the model for neural agent state estimation. The system is tested under Windows/Linux operating systems, with exploit commands in the action set, and its superiority over the conventional GRU model is demonstrated.

Keywords: deep reinforcement learning, penetration test, POMDP, autonomous cyber reasoning system, BERT

1. はじめに

近年において，人工知能を用いたセキュリティが注目され，マルウェア検知等，様々な応用がなされている．防御だけでなく，攻撃においても盛んに研究されており，人工知能を用いたハッキングコンテストも開催されている．そのうちの 1 つがサイバーグランドチャレンジと呼ばれる，DARPA（米国防高等研究計画局）によって開催された世界初の全自動ハッキングコンテストである．予算としては 56 億円が投入され，上位チームには 10 億円の賞金が授与されている [2].

大塚研究室においても，人工知能を用いた攻撃の研究がなされており，深層強化学習を用いた自律サイバー推論システム [1] が提案されている．本研究の目的は，自律サイバー推論システムがネットワーク環境上において攻撃を行える機能を付与することである．

以降本稿では，2 章で代表的な強化学習手法について説明し，3 章で自律サイバー推論システムの紹介，4，5 章で関連論文を述べ，6 章で提案手法を述べ，7 章で今後の展望について述べる．

2. 強化学習

2.1 強化学習とは

強化学習は，エージェント (行動主体) がある環境の状態に応じて，どのように行動すれば報酬が多くもらえるかを求める手法である．教師なし学習や教師あり学習とは違い，学習データなしに自身の試行錯誤のみで学習するのが特徴である．強化学習の学習サイクルは下記のとおりである．

①エージェントは最初に何を判断すべきか分からないので，選択できる行動の中から，ランダムに選択を行う．

②行動に伴う報酬を受け取ったとき，どのような状態で，どのような行動をしたら，どの程度の報酬が貰えたかという経験を記憶する．

③それら経験に応じて方策 (エージェントが行動を決定する戦略) を求める．

④ランダムな動きは残しつつ，方策を手掛かりに行動を決定する．
⑤②~④を繰り返して，将来的に多くの報酬を得られる方策を求める．

2.2 Q 学習

本項では強化学習の一般的な手法である Q 学習を紹介する．ほとんどすべての強化学習アルゴリズムは価値関数 (value function) に基づく評価を行っている．価値関数は 2 種類存在しており，状態価値関数と行動価値関数が存在している．状態価値関数は，エージェントがある状態にいることがどれだけ良いのかを評価し，行動価値関数はある状態において，ある行動を行うことがどれだけいいのかを評価する．本項では Q 学習で使用する行動価値関数を取り上げ，下記にその行動価値関数を表した式を示す．

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty}\gamma^{k} r_{t+k+1} \middle| s_t = s, a_t = a\right]$$

s が状態を表し，a が行動を表している．r がある行動をした場合に受け取ることのできる報酬値を示しており，γ は将来の報酬値をどれだけ考慮するのかを数値で示したものであり，$0 \leqq \gamma \leqq 1$ の範囲で示される. 行動価値関数である $Q^{\pi}(s,a)$ はある状態 s で行動 a をとり，その後に方策 π に従った期待報酬として定義される．この行動価値関数の更新に用いるのが，Q 学習であり，下記の式で定義される

$$Q(s_{t+1}, a_{t+1}) \leftarrow$$

$$Q(s_t, a_t) + \alpha[r_{t+1} + \gamma\, max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

$r_{t+1} + \gamma \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$ の部分は TD 誤差と呼ばれており，行動前と行動後の評価値の誤差のことである．最適化によって TD 誤差を 0 に近づけることで，行動前と行動後の評価値が一致するようになり，行動に対する報酬が正確に予測できるようになる．ハイパーパラメータである α は 1 回の学習で更新される大きさを表している．この学習によって得られる行動価値関数は使われている方策とは独立に，最適な行動価値関数を直接近似する．このため，Q 学習は方策オフ型と呼ばれている．Q 学習では，行動価値関数を表形式で表現しており，その表を Q テーブルと呼んでいる．表にはある状態とある行動に対する Q 値が格納されている．任意の状態と行動が決定することで Q 値も求めることができるため，Q 値が最大となるような行動を選択することができる．

2.3 深層強化学習

深層強化学習とは強化学習と深層学習を組み合わせたもので，本稿では Q 学習の欠点を克服した DQN を紹介する．Q 学習の欠点として行動価値関数を表形式で表現しているため，状態の種類が増加すると，表の行数も膨大となる．そのため，非常に多くの学習が必要となってしまい，計算量的に現実的ではなくなる．そこで，その欠点を克服すべく，行動価値関数を表形式ではなく，ニューラルネットワークで表現する DQN が提案された．DQN は CNN を用いて行動価値関数の近似を行う．DQN の入力としては状態が入力され，出力としては行動価値が出力される．通常ニューラルネットワークは誤差関数を指標として，パラメータ (重み) を更新していくことで，精度を上昇させていく．DQN の誤差関数では前項で紹介した TD 誤差の考

[1] 情報セキュリティ大学院大学
Institute of Information Security
[a)] mgs211101@iisec.ac.jp

えを使用し，式を下記に示す．

$$\frac{1}{2}\left[r_{t+1}(s_t,a_t)+\gamma \max_a Q(s_{t+1},a_{t+1})-Q(s_t,a_t)\right]^2$$

損失関数では TD 誤差の部分に 2 乗をかけて出力している．この損失関数によって本来目指すべき行動価値と現在の行動価値とのズレを表すことができるようになり，このズレを解消すべく，パラメータの更新を行っていく．

2.4 部分観測マルコフ決定過程（POMDP）

マルコフ決定過程 (MDP) ではエージェントはマルコフ性のある状態を常に観測できるとしているが，部分観測マルコフ決定過程では，状態を部分的にしか観測できず，観測状態がマルコフ性を満たすとは限らない状況を扱う確率過程である．POMDP において最も重要な点は MDP とは異なり，エージェントは状態を観測できず，代わりに観測と呼ばれる値を環境から観測する．下記に POMDP を定義した式を示す．

$\mathcal{S}=\{s^1,\ldots,s^N\}$： 状態 (state) の有限集合

$\mathcal{A}=\{a^1,\ldots,a^K\}$： 行動 (action) の有限集合

$T:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to[0,1]$： 条件付き状態遷移確率

$R:\mathcal{S}\times\mathcal{A}\to\mathbb{R}$： 報酬関数 (reward function)

Ω： 観測の集合

$O:\mathcal{S}\times\mathcal{A}\times\Omega\to[0,1]$： 条件付き観測確率の集合

部分観測マルコフ決定過程は, 状態更新関数 $u:\mathcal{S}\times\mathcal{A}\times\Omega\to\mathcal{S}$ を用いて, 次状態を以下のように推定するプロセスとして定義される．

$$s_{t+1}=u(s_t,a_t,o_{t+1}),\qquad \text{for all } t\geq 0$$

ここで s_0 は初期状態を表す．

3. 自律サイバー推論システム

3.1 自律サイバー推論システムとは

自律サイバー推論システム [1] は深層強化学習を用いて，サイバー空間における未知の攻撃に対し，迅速かつ的確に対応することを目標に研究されたシステムである．このシステムは，Microsoft 社が提供しているテキストベースのアドベンチャーゲームである TextWorld[4] を CTF 問題に当てはめ，その TextWorld で高い成果を出したニューラルエージェントを搭載し，CTF を攻略できるようにしたシステムである．このシステムのエージェントの目標としては，Unix 環境上に設置されたファイルに対し，string コマンドを実行し flag を取得することである．このシステムに搭載されているニューラルエージェントについては次項で解説を行うが，全体的な流れとしては，まず，コマンドの実行によって得られた部分観測情報 (出力結果等) とアクションとして使用できる全コマンド情報をニューラルエージェントの入力として与える．それらの情報をニューラルエージェントで処理を行い，コマンドごとのスコアの確率分布を基にコマンドのサンプリングを行う．

3.2 ニューラルエージェント

ニューラルエージェント [3] とはニューラルネットワークにより構築される強化学習エージェントのことである．図 1 にてその概要を示す．ニューラルエージェントは観測された環境を元に，Unix に使用されるコマンドのリストから最も有望なコマンドを選択するようにトーレーニングを行っていく．状態を完全に観測できないため，観測された部分観測情報を元に，学習を行う．ニューラルエージェントは主に 4 つの段階に分かれている．

context encoding：入力としてコマンドによる出力情報やディレクトリ情報等の部分観測情報を与える．次に，入力情報に embedding を行い，分散表現を獲得し，それを GRU レイヤーに渡す．次に GRU レイヤーで語順情報を含んだベクトルに変換し，連結を行う．最後に次の GRU レイヤーに渡され，前時刻の隠れ状態を受け取り，現時刻における隠れ状態を出力する．

value：現時刻の隠れ状態を全結合層に渡し，状態価値の算出を行う．

commands encoding：こちらの部分では使用する全コマンドを入力として与える．それら入力に embedding を行い，ベクトル化し分散表現を獲得する．次に GRU レイヤーにおいて語順情報を含んだベクトルに変換し，score & action に渡す．

score & action：現時刻の隠れ状態及び，各コマンドの行列情報を組み合わせて全結合層に渡し，コマンドごとのスコアを算出を行い，ソフトマックスで確率分布に変換し，コマンドのサンプリングを行う．

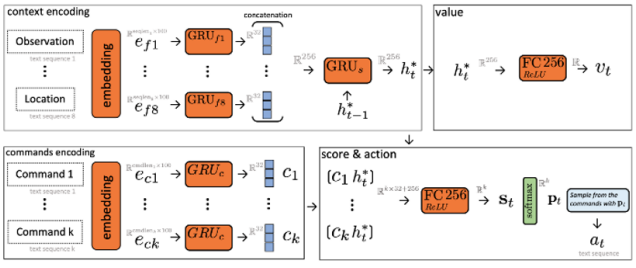

図 1 ニューラルエージェント [3]

4. BERT

これまでの Deep Learning は大量のデータと大量の計算資源が必要であったが，BERT[7] は Fine-tuning を行い，特定のタスクのために微調整するだけで，少データ，少計算資

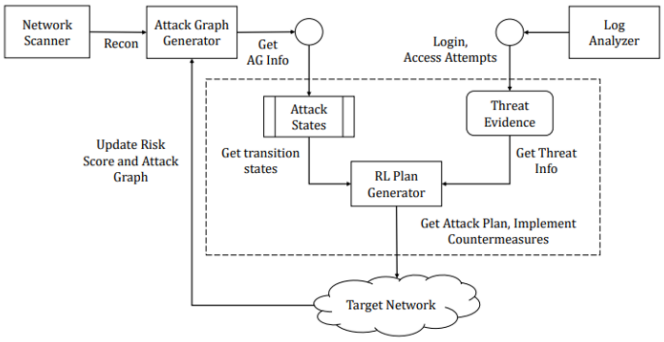

図 2 ASAP のシステム概要図 [9]

源でも自然言語処理モデルを作ることができる．2019 年には Google 検索に BERT モデルが適用されている．BERT のモデル構造は Transformer のエンコーダ部分のみで構成されている．以下にその詳細を示す．

①BASE：12 層，隠れ層 768，12 ヘッド，1 億 1000 万 パラメータ

②LARGE：24 層，隠れ層 1024，16 ヘッド，3 億 4000 万 パラメータ

BERT の入力表現は，トークン embedding とセグメント embedding と位置 embedding の合計で表される．入力表現図を図 4 で示す．BERT の特徴としては，Pre-training と Fine-tuning に学習フェーズを分けていることである．学習概要図を図 4 に示す．Pre-training(事前学習) では Masked Language Modeling(MLM) タスクにより言語の文法と単語の意味などの理解を行い，Next Sentence Prediction(NSP) タスクにより文意，文脈の理解を行う．Masked Language Modeling タスクでは，BERT でテキストの一部を [MASK] という別の単語で置き換えたテキストを入力する．BERT は，ある単語を周りの単語から予測するというタスクを用いて，単語の入出力関係を学習を行っている．具体的には，入力トークンの 15 %を [MASK] という特殊トークンに置き換える．そして，置き換えられた文章を BERT に入力し，[MASK] の位置に元々あったトークンを予測するというタスクを用いて学習を行う．さらに，Fine-tuning との差異を埋めるために，トークンを常に [Mask] トークンに置き換えるのではなく，80 %を [MASK] トークンで置き換え，残りの 20 %のうち 10 %をそのままの単語にし，残りの 10 %をランダム選ばれたトークンで置き換えている．次に Next Sentence Prediction(NSP) タスクでは，2 つの文章の関係性について予測するタスクを行う．入力される 2 つの文のうち，後の文が 50 %の確率でランダムに置き換えられる．これらの文章は [SEP] という特殊トークンで分けられ，入力された 2 つの文が連続したものであるか，そうでないかを判定するタスクを繰り返し，学習を行う．また，Pre-training では，BooksCorpus(8 億単語) と Wikipedia(25 億単語) で学習が行われている．次に Fine-Tuning（ファインチューニング）では，ラベル付きデータを用いて，特定のタスク（ラベル分類，質疑応答）などに特化させるように事前学習モデルに学習させる．

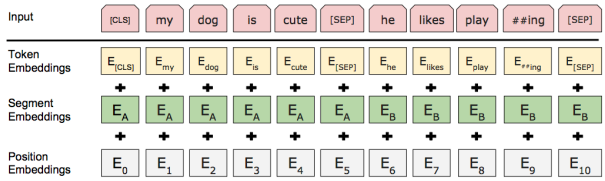

図 3 BERT の入力表現図 [7]

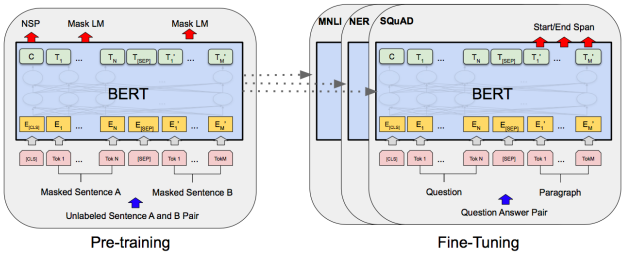

図 4 BERT における学習概要図 [7]

5. 従来研究

本章では，深層強化学習を使用した既存のペネトレーションテスタである ASAP[9] を紹介する．ASAP のシステム概要を図 2 に示す．このシステムでは，MDP をベースに多数のノードを有したネットワークに対して，ペネトレーションテストを行う目的で作成されたシステムである．このシステムの流れを下記に示す．

①Nessus や nmap を使用して，対象ネットワークに対してスキャンをかける．スキャン後，脆弱性情報を抽出する．

②抽出された脆弱性情報を attack graph tool である MulVAL に渡し，attack graph を作成する．

③attack graph を解析し，対象ネットワークに存在する特権ノード，脆弱性の存在するノードを抽出する．また，それら抽出されたノードから，CVSS スコアやそれらノードに対するアクセスの複雑さなどを基に，強化学習に必要なパラメータの抽出を行う．

④それらのパラメータを RL Plan Generator に渡し，DQN をもとに，agent がこのネットワークを攻略する上で最適な経路を導くために，学習を行う．学習の結果，導かれた経路の有効性を確かめるために，metasploit を使用して実際に攻撃を行い，その攻撃結果を基に，attack graph を更新する．

⑤更新，抽出，学習，攻撃を繰り返して，将来的に最も多

くの報酬を得られる最適な攻撃経路を求める．

このシステムの有効性を確かめるべく，実験として NASim と呼ばれるネットワークアタックシュミレータ上に，産業用制御システム（Subnet2）と IoT デバイス（Subnet3）で構成される企業内ネットワークをシミュレーションしている．このネットワークは 16 台のホスト，3 つのネットワーク（Net1-3）で構成されており，実験には Windows と Linux を使用している．また，4 つの主要サービス (SSH, FTP，HTTP，SMTP) を展開している．そのターゲットネットワーク図を図 5 に示す．エージェントには，SMTP サービスに存在する脆弱性と IoT サブシステムに存在する脆弱性を利用しての侵入を行うという目標が与えられている．このネットワークに対して DQN アルゴリズムは，かなり早く収束を達成した．割引率とバッチサイズを変化させた際の実行時間の対応図を図 6 に示す．収束時間は，γ の値を変更することで変動している．γ = 0.6, 0.7 の場合，2.5(s) から 4(s) の間であり，γ =0.8 で約 2(s) で最も高速に収束することが確認された．また，3 つの脆弱性（CVE-2011-0411, CVE-2013-2566, CVE2011-1431）をランダムに配置した 300 のホストを用意し，対象ホストの脆弱性のアクセス複雑度は medium,hard である．そのような大規模ネットワークに対しては約 70 秒以内に最適な攻撃計画を立てることができている．

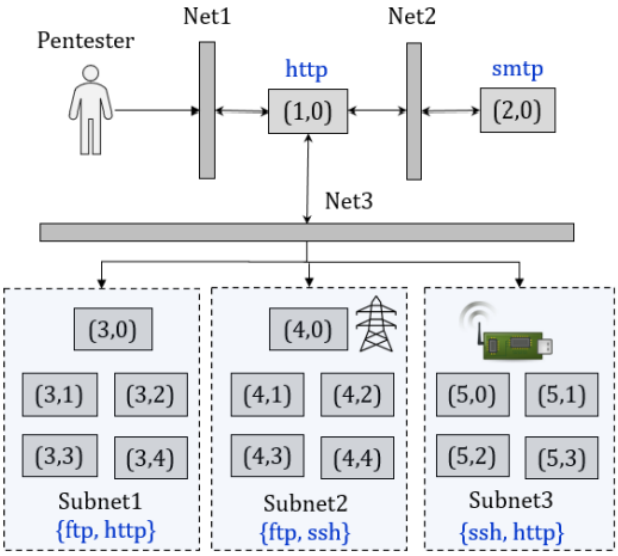

図 5 ターゲットネットワーク [9]

6. 提案手法

次に，今回構築した提案手法について紹介する．従来の ledeepchef では生の部分観測情報を embedding し，複数の GRU レイヤーで encode するという流れになっているが，本手案システムでは GRU レイヤーを SecBERT[11] に変更し，BERT モデルの出力する隠れ状態を連結させ，別の

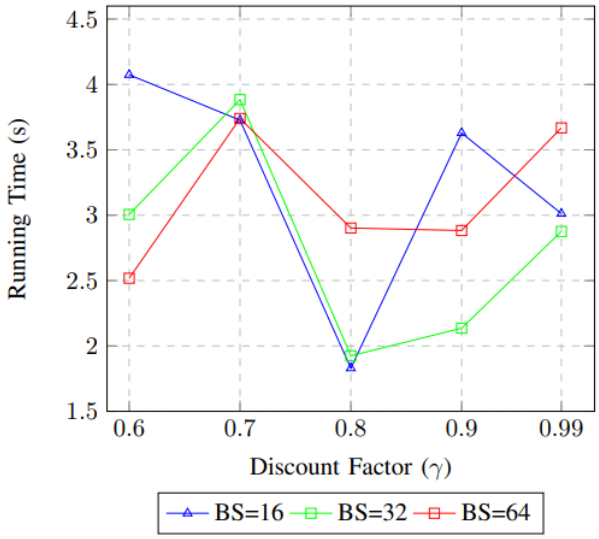

図 6 割引率とバッチサイズを変化させた際の実行時間の対応図 [9]

GRU レイヤーに通し，現在状態である h_t^* を出力する．システム図を図 7 に示す．SecBERT は，BERT をマルウェアに関する情報や metasploit で使用するモジュールの詳細等の各種セキュリティコーパスを用いてトレーニングされている．本論文では，我々が提案したシステムが従来の自律サイバー推論システムに比べて有効であることを 2 つの実験結果をもとに示す．

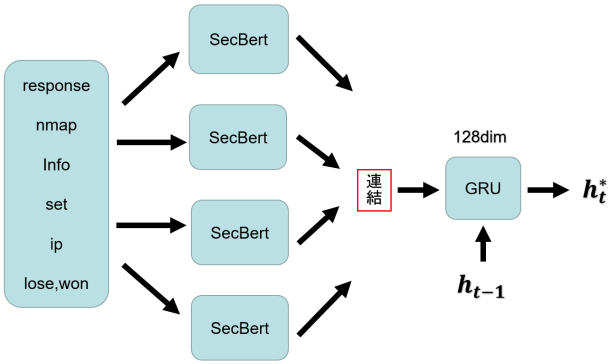

図 7 提案システム（状態推定部）

6.1 実験

実験 1 は ledeepchef と提案システムを metasploit 上で作用させ，1 つの脆弱性を持つターゲットサーバに対して exploit を行うという内容となっている．実験ネットワーク図を図 8 に示す．この問題を解く最短のステップは

①exploit モジュールの指定を行う．

②ターゲットの IP を指定する．

③exploit を実行する．

上記の 3 ステップである．行動空間としては exploit に必

要な msfconsole コマンドや exploit に必要のない unix コマンドを含めた 20 個である．状態推定に必要な部分観測情報としては実行コマンドのレスポンス，nmap によるポートスキャン結果，exploit モジュールの説明である info 情報，exploit を行う上でのターゲット IP やペイロードのセッティング状況を確認することができる set 情報，ターゲット IP 情報やそのゲームの勝利や敗北を表している won，lose 情報をステップごとにニューラルエージェントに与えている．また，報酬設計としては，exploit に成功した場合は勝利として 30，非ターゲットを指定した場合には敗北として-10，最短の手を最終的に打てるようになるためにステップごとに必ず-1 をスコアとして与えている．割引率としては，現在の状態価値と将来の状態価値は非常に相関が高いため，0.95 に設定している．実験 2 としては，ターゲットサーバの脆弱性は変更せずにニューラルエージェントに与える行動空間のみを増加させて，実験を行う．与える行動空間としては実験 1 に比べ倍の 40 個を与えている．

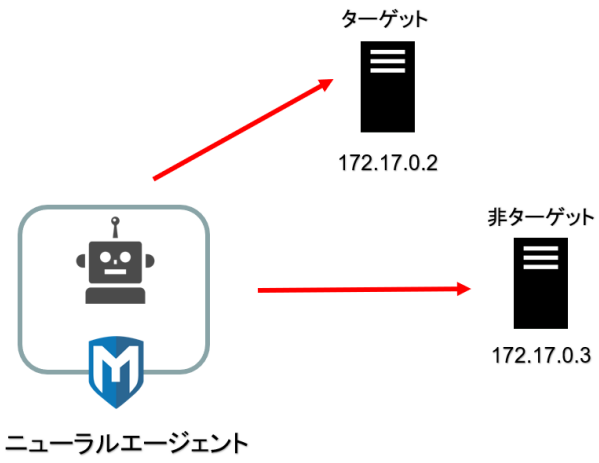

図 8　実験ネットワーク

6.2　結果

実験 1，2 の結果をそれぞれ図 9，図 10 に示す．図としては 100 ステップごとの合計報酬値をプロットしている．実験 1 では ledeepchef + SecBERT が約 2000 ステップ目から合計報酬値が急激に上昇し，約 200〜300 の間を推移しているのに対して，ledeepchef のみでは約 4000 ステップ目から合計報酬値が上昇し始め，約 50〜200 の間を推移している．実験 1 においては ledeepchef + SecBERT の方が早く学習の収束が進み，合計報酬値も高止まりしている．実験 2 では，ledeepchef + SecBERT は約 12500 ステップ目から徐々に合計報酬値が上昇し，約 17000 ステップ目から急激に上昇し，200〜300 の間を推移している．一方，ledeepchef のみでは合計報酬値の推移は約-100 から上昇することはなく，ランダムのような動きとなっている．

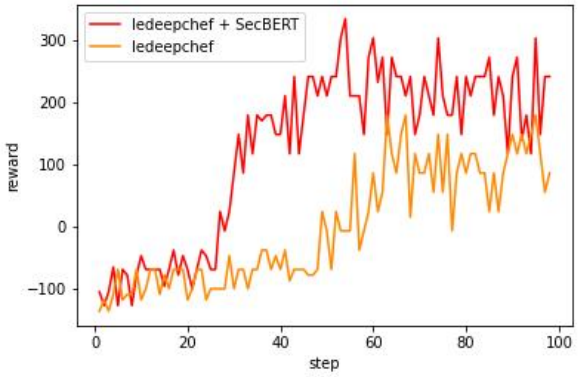

図 9　実験 1 結果 (x100 ステップ)

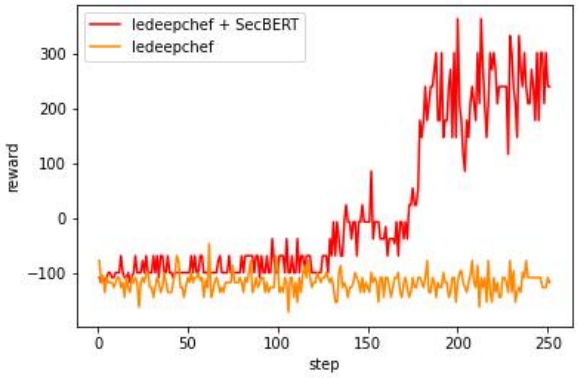

図 10　実験 2 結果 (x100 ステップ)

6.3　考察

これらの実験結果から，従来の自律サイバー推論システムに比べて，提案手法の方が状態推定の精度が上昇しており，より広い行動空間に対しても有効であるということを示した．これらの結果に起因することは，Textworld に比べて，msfconsole 上で得られる部分観測情報の文章量が多く，長文に対して GRU ではうまく状態推定が行えていないことである．また，ステップごとに与える行動空間が増加すればする程，ledeepchef の学習が不安定化していき，実験 2 の 40 コマンド問題では全く適応できていないことが分かる．さらに，実験 1，2 の両方で ledeepchef + SecBERT は最終的に平均で 7〜8 ステップで exploit に成功しており，3 ステップ問題に対して完全ではないが，最適化に向けて学習ができている．

7.　今後の展望

今後は，学習のエピソードごとにターゲットサーバの開放ポート，脆弱性を変更し，非定常な環境に対しても安定して exploit できることを確認する．また，問題を拡張していき，exploit だけでなく，post explotation まで含めた複雑な問題に対応できるエージェントを作成していきたい．最後に，SeqGAN 等を用いて，exploit や post explotation

で使用するコマンドを自動生成できるように学習させ，ペネトレーションテストの完全自動化を目指していきたい．

謝辞 本研究は、防衛装備庁が実施する安全保障技術研究推進制度 JPJ004596 の支援を受けたものである．

参考文献

[1] 藤本大輔 ”深層強化学習を用いた自律サイバー推論システムの研究.” 修士論文 情報セキュリティ大学院大学 (2020).
[2] Cyber Grand Challenge (CGC) : 世界初のマシン同士の全自動ハッキングトーナメント by Tyler Nighswander.
[3] Adolphs, Leonard, and Thomas Hofmann. “LeDeepChef: Deep Reinforcement Learning Agent for Families of Text-Based Games.” In 34th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence (AAAI 2020), 5180, 2020.
[4] Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Ruo Yu Tao, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, Adam Trischler, A. 2018. Textworld: A learning environment for text-based games. CoRR abs/1806.11532.
[5] Zennaro, Fabio Massimo, and Laszlo Erdodi. ”Modeling Penetration Testing with Reinforcement Learning Using Capture-the-Flag Challenges and Tabular Q-Learning.” arXiv preprint arXiv: 2005.12632 (2020).
[6] Zennaro, Fabio Massimo, and Laszlo Erdodi. ”The Agent Web Model – Modelling web hacking for reinforcement learning.” Sep 2020 arXiv preprint arXiv: 2009.11274 (2020).
[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, pages 6000–6010.
[9] Ankur Chowdary, Dijiang Huang, Jayasurya Sevalur Mahendran, Daniel Romo, Yuli Deng, Abdulhakim Sabur. ”Autonomous Security Analysis and Penetration Testing” 2020 16th International Conference on Mobility, Sensing and Networking (MSN).
[10] Isao, T.: Metasploit Meets Machine Learning, https://www.mbsd.jp/blog/20180228.html.
[11] ”GitHub - jackaduma/SecBERT: SecBERT”.
[12] 佐竹達也，大塚玲: 部分観測マルコフ決定過程によるニューラルエージェント強化学習を使用した自律型 SQL インジェクション攻撃手法．In: The 39th Symposium on Cryptography and Information Security．(2022)
[13] 米田智紀，大塚玲：部分観測マルコフ決定過程に基づいたニューラルエージェントを用いたペネトレーションテスト手法の提案．In: The 39th Symposium on Cryptography and Information Security．(2022)