サイバー空間における攻撃と防御の共進化シミュレーション

石川　博也†　八槇　博史†

概要：人工知能を利用し Beyond the Attackers を実現し，Proactive な対策を可能にする研究を実現するため LIFT システムを拡張した Super-LIFT システムの開発が進められている．我々はそのアプローチの一つとして共進化モデルに基づく出現の予測を目指している．
これまで進化計算を効率的に行うためのクラウド上のシミュレーションプラットフォームを構築してきた．
今回，Exploit Database や CVE の情報を元に，遺伝的アルゴリズムを用いた攻撃と防御の共進化モデルを示し，進化計算を利用してサイバー空間上の攻撃と防御の共進化シミュレーションを行う上での検討を進めた．

キーワード：人工知能，進化計算，

Simulation of Coevolution on Attack and Defense in Cyberspace

Hiroya Ishikawa†1　Hirofumi Yamaki†2

Abstract: Realized Beyond the Attackers using artificial intelligence, the development of Super-LIFT system that extends LIFT system for implementing a research that enable Proactive measures are underway.
We are aiming to prediction of malware appearance based on the co-evolution model as one of its approach.
So far it has been to build a simulation platform on the cloud in order to executing the evolutionary computing efficiently.
In this paper, We present a co-evolutionary model of attack and defense using a genetic algorithm, based on the information of the Exploit Database and CVE.

Keywords: Artificial Intelligence，evolutionary computing

1. はじめに

近年，特定の企業や組織を攻撃対象とする標的型メール攻撃が問題となっている．標的型攻撃とは，金銭や知的財産等の秘密情報の不正な取得を目的として特定の標的に対して行われるサイバー攻撃である．その中でも，攻撃対象にメールを用いて攻撃を行う標的型メール攻撃[1]が問題となっている．サイバー攻撃がますます，巧妙かつ悪質になることが予想され，マルウェアにも AI 機能が組み込まれ自動的に能力アップする機能が組み込まれてくる可能性が大きい．
また，マルウェアの急速な増加の背景としてマルウェア生成ツールキットによって手軽に多様な組合せのマルウェアが可能になっていることやマルウェアの亜種が大量に発生いる．新たな対策が講じられるとそれに対応して新たな手法が生まれるといったイタチごっこが続いている．
このマルウェアの進化動向は自然界におけるウイルスや細菌の進化と似たような性質を持っている．

これに対抗するため防御側も進化計算を利用して攻撃側の変化に対応するようなモデルをシミュレーションすることを目標としている．

標的型メール攻撃問題について，インシデント発生時にネットーワークログ等のデータを適切に利用し，人工知能を用いて自動的な応急対応を可能とするとともに，運用者が適切な対策を取れるようにするために LIFT(Live and Intelligent Network Forensic Technologies)[2]の開発が行われてきた．これは，既に発生した攻撃と同様な攻撃が起きた場合には対応できるが，新しい攻撃に対応するのは困難であるという問題がある．この問題を解決するために人工知能を利用し Beyond the Attackers を実現し、Proactive な対策を可能にする，LIFT システムを拡張した Super-LIFT システムの開発が進められている．我々はそのアプローチの一つとしてマルウェアの生成を共進化モデルに基づいた予測を目指している．

標的型攻撃が，人工知能技術を搭載したマルウェアを用いて高度化されるという将来像を前提として，攻撃者に先回りする形で攻撃内容を自動合成してシミュレーションを行い，防御方法までを自動的に生成するための技術について研究する．
攻撃内容とそれに対する防御は，人工知能技術の一分野で

† 東京電機大学
Tokyo Denki University

ある進化計算を用いて行い，膨大な数の進化パターンを効率的に探索する．攻撃内容と防御手法がお互いに適応して進化する「共進化」の枠組みを取り入れ，その計算をクラウド上で行うことにより，ネットワーク内で起きうる 攻撃とそれへの適切な防御とを求めることができる．これにより，従来のサイバー攻撃対策では 困難であった「先回り」したプロアクティブな防御を実現することが可能と考えている．

これまで進化計算を効率的に行うためのクラウド上のシミュレーションプラットフォームを構築してきた．

今回，Exploit Database や CVE の情報を元に，遺伝的アルゴリズムを用いた攻撃と防御の共進化モデルを示し，進化計算を利用してサイバー空間上の攻撃と防御の共進化シミュレーションを行う上での検討を進めた．

2. 進化するマルウェアの自動生成

目標とする結果をもたらすプログラムを生成する技法には様々なものがあるが，ここでは典型的なものとして進化計算を考える．

Noreen ら[3]は，実在のマルウェアである Bagle ファミリに対し，そこで使用されるタイムスタンプや偽装するアプリケーションなどを遺伝子として定義して進化的アルゴリズム（以下 GA）を適用することによって，攻撃として有効でありながらもアンチウィルス製品の検知からは逃れるような攻撃コードが自動的に生成されうることを示した．

また，Duchene ら[4]は，クロスサイトスクリプティング(以下 XSS)攻撃に用いられる攻撃用文字列の生成に GA を適用し，有効な XSS 攻撃が実現されうることを示している．

現在のサイバー攻撃においても，多数の攻撃コードが生成され，様々な対象に対する攻撃に使用され，有効性の高かったコードがブラックマーケットを通じて流通するといったことが実際におきている．これはある意味で攻撃コードの進化現象ととらえてよいと考えられる．現在の進化は攻撃者によるコード生成と実在する攻撃対象での検証というループが主であるが，これが仮想システムを用いた進化計算の適用によって加速されるのは時間の問題であろうと筆者は考えている．

攻撃がこのように進化する一方で，防御についても新しい攻撃の出現に対応するように進化が起きている．例えば，コンピュータウィルスの蔓延→アンチウィルスの普及→亜種の大量発生→マルウェア分類技術の発展→・・・といった変化が起きている．また別の事例を挙げれば，不正アクセスの増加→ファイアウォール，IDS の普及→内部侵入手法の高度化→SIEM 等による検知→・・・といったような進化が実際に生じている．

大局的に見れば，これはサイバー攻撃と防御との共進化（Co-evolution）と見ることができる．すなわち，どちらか一方が単独で進化するのではなく，互いへの対応の中で高度化していくと考えるべきである．マルウェアや攻撃手法の複雑化・巧妙化と，防御技術の発達とは各々が単体で起こるわけではない．攻撃手法と防御技術とがお互いに克服しあうように発達するのであるとすれば，未来の攻撃内容はその相克の延長線上にあると考えることができる．そうであるならば，共進化のプロセスを考えることで，将来にありうるサイバー攻撃とその防御も共進化の枠組みでとらえることができるかもしれない．

我々が考える，マルウェア進化予測のための共進化モデルの概要を図 1 に示す．

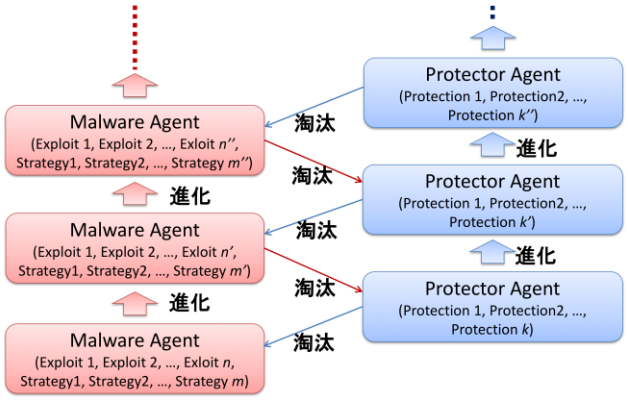

図 1　共進化モデルにおけるマルウェアと防御手法の予測

共進化計算の枠組みに従い，マルウェアやそれに対抗するためのソフトウェアを総称してエージェントと呼ぶ．マルウェアをモデル化したものが Malware Agent である．Malware Agent には，攻撃に用いる脆弱性（Exploit）と，それらを組合せることで実現される攻撃戦略（Strategy）とが，各々複数種組み込まれる．

他方，防御側のモデルが Protector Agent である．ログ取得，ファイアウォール設置，ハードニングなどのセキュリティ保護施策（Protection）が組み込まれる．

Malware Agent の評価は，それが投入された環境に存在する Protector Agent の持つ保護施策によって決定される．ここで進化計算を行うことで，Malware Agent の攻撃内容が進化する．他方，Protector Agent もそれが対する Malware Agent により評価が定まり，同様に進化が起きる．Protector Agent が進化すると，Malware Agent の評価が変わるため，新たな進化が引き起こされる．同様に，Malware Agent の進化は Protector Agent の進化を促す．これを繰り返すことによって，Malware Agent と Protector Agent の間で共進化が起きる．

現在我々はこのビジョンのもとで，Malware Agent や Protector Agent の生成システム，評価のためのシミュレータ，および共進化計算を効率的に行うためのプラットフォームについて設計と実装を進めている．

3. 進化計算

3.1 遺伝的アルゴリズム GA

進化計算のプロセスを計算機上で扱う手法は，Genetic Algorithm(GA)[5]などとして知られており，最適化・探索・学習などに用いられる．GA では自然界における生物の進化過程と同様に，ある世代を形成しているいくつかの個体の集合，すなわち個体群の中で，環境への適合度の高い個体が高い確率で選択される．その個体に対して，遺伝子の交叉や突然変異が，ある確率で発生することによって，次の世代の個体群が形成され，最後に得られた個体群の中で最も適合度の高い個体を最良個体（導いた答え）となる．「変異(mutation)」「交叉(crossover)」「選択(selection)」を個体群に対して繰り返すことで，その個体群を生み出す遺伝子プール内の遺伝子分布の変化がすなわち「進化」である．例えば Genetic Algorithm(GA)では，以下のようにして計算を行う．

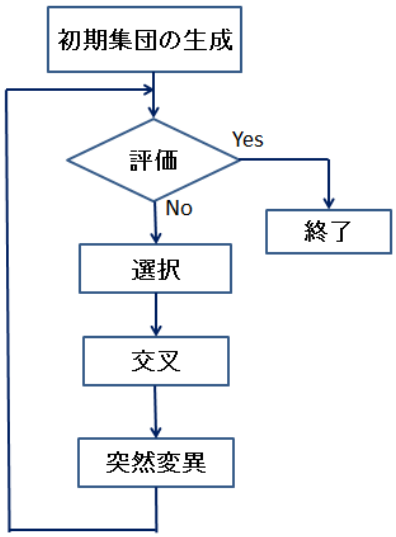

図２　遺伝的アルゴリズムのフローチャート

１．最適化問題におけるパラメータセットをベクトルで表し，これを「遺伝子」と呼ぶ．初期値として複数の遺伝子を用意する．

２．遺伝子それぞれに対する評価関数の値を計算する（評価）．

３．評価関数値の上位の遺伝子を高確率，下位の遺伝子を低確率で残し，残りを捨てる（選択）．

４．残った遺伝子をもとに一部のパラメータを変更（変異）したり，遺伝子同士の部分を交換（交叉）したりして，新しい世代の遺伝子群を生成する．

５．２～４を繰り返す．

通常の進化計算では２で用いられる評価関数は，その個体の環境への適応度の度合いとして表現される．

各個体は，それぞれ染色体によって特徴付けられており，染色体は遺伝子 の集まりで構成されている．また，通常 GA では 1 染色体で 1 個体を表現す る．GA で扱う情報は遺伝子型と表現型の 2 層構造からなる．

・ 遺伝子型・・・GA のオペレータの操作対象．染色体の構造にあたる（例：01110，00110：0,1 のビット列）

・ 表現型・・・問題の中での構造を表す

（例：14，6：具体的な数字）

なお，遺伝子型から表現型に写像することをコード化，逆に表現型から遺 伝子型に写像することデコード化という．遺伝子型によって適合度が決定し，その適合度の大きいものほど子孫を作りやすく，小さいものほど死滅しやすい(淘汰されやすい）ようになっている．このことより，次世代の各個体の 適合度は前世代よりも良いことが期待される．

3.2 共進化アルゴルリズム

共進化とは，本来は進化論における概念であり，「複数の種が相互に影響を 与えながら環境への適応力を高める方向に進化すること」という定義である． 共進化では，自分や相手が進化することによって自分たちの置かれた環境 が動的に変化し続ける特徴があり，相手の行動に対して自分が有利になるような行動をとる．その結果，単独進化と比べて，より優れた行動が得ることができる．

共進化の結果，種の間には以下の 3 点が成立する．

・ 競争（両者が害を与え合う）

・ 協調（両者共に利益を受ける）

・ 寄生（一方だけが利益を受ける） 一般には，進化の過程で寄生から協調へと移行するとされている．

共進化アルゴリズムは，従来の進化的計算のように静的に与えられた適応度関数を用いるのではなく，個体同士が影響を及ぼしあうことによって適応度が決定する．ある個体の適応度はそれを評価するための集団によって違うため，適応度地形は世代に応じて動的に変化する．

また，その相互作用の定義によって競合型，協調型の 2 種類に分けられる．競合型共進化 GA と，協調型共進化 GA があるが，本研究では攻撃と防御は競争型として考える．

3.2.1 競合型共進化 GA

競合型共進化 GA の特長として，以下の 2 点が挙げられる．

・ 両集団が互いに適度な選択圧をかけるので，局所解に陥りにくい．

・ 解候補の完全な評価が困難な問題において，部分的な評価で実現可能． また，競合型共進化 GA の問題点は，以下の 2 点が挙げられる．

- 適応度の評価が相対的であるために，順調に進化しているように 見えても，同じところで堂々巡りしている可能性がある．

- 効率的な進化を実現するために集団間の進化の歩調が合っている ことが保証されていない．適応度の差が開きすぎてしまうと，個体間の適応度の差が有意でなくなってしまい，両者の進化が停滞してしまう．

4. 進化計算機構

4.1 サイバー攻撃シミュレーションにおける進化計算機構

サイバー攻撃シミュレーションにおける進化計算機構の概要を図３に示す.

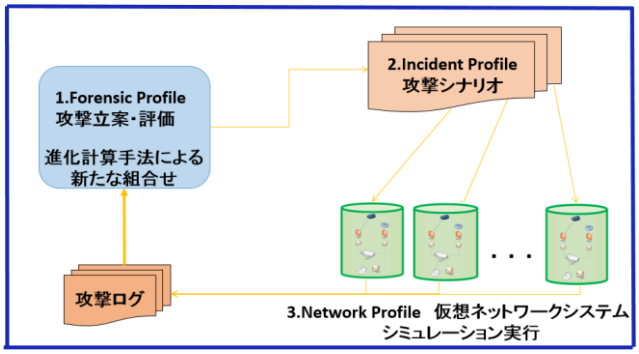

図３　サイバー攻撃シミュレーションにおける
進化計算機構の概要

サイバー攻撃シミュレーションの流れは以下のようになる.

1. Forensic Profile 攻撃立案・評価を行う
2. Incident Profile　攻撃シナリオを生成
3. Network Profile　仮想ネットワークにおいてシミュレーションの実行

攻撃立案は進化計算によって新たな攻撃の組合せを生成する.攻撃評価はログ情報から有効な攻撃であるか評価する.シミュレーション実行は実システムをモデルとした仮想ネットワークを構築し，攻撃シミュレーションをする.

攻撃と防御各々の個体の情報をパラメータとして扱い，２節で述べた攻撃と防御の予測を進化計算機構によって実現される.

攻撃と防御のパラメータとして対象となるエクスプロイトコードや CVE の数は膨大にあり，パラメータ空間が巨大になり研究者は膨大な量の試行錯誤を強いられる傾向にある．このシミュレーション研究を容易するのがパラメータサーベイ機構である.

4.2 クラウドを用いた進化計算機構

シミュレーション研究においては，モデルの性質や特性の分析のために，シミュレーションモデルのパラメータを変更しながら，結果を確認することで研究を進めていく．そのため，各パラメータの組み合わせ（以下，パラメータセット）を大量に試行することが強いられる.

パラメータサーベイ機構とは利用者が求める特徴を示すパラメータセットから優先的に分析・検証が行えるよう実行順序を制御する機構である.

パラメータサーベイ機構では進化計算を用いることにより，利用者がパラメータ空間内のどこに研究上有用なパラメータセットがあるかの見当がつかない場合でも，なるべく有用な結果に近いパラメータセットを見つけ出すことで研究効率の向上を図る.

進化計算を用いることにより，現象を発現するパラメータセットを優先的に実行するよう操作する．しかし，進化計算をでは，最終目標が最適性の保証はないが，限りなく最適解に近いであろう値を見つけ出すことを目的としているが，このパラメータサーベイでは最適値を求めることが目的ではない．対象とする パラメータ空間から，ある現象を発現する，ある特徴を示すパラメータセットを探索し尽くし，研究者の研究効率を向上させることが目的となる．そのため，最適値が見つかり次第操作が終了するというわけではない．進化計算を用いたパラメータサーベイ機構を用いる際に,利用者は望む結果の特徴を指定する必要がある.進化計算では，なるべく最適解に近い方向へ探索する際に,最適解に近いかどうかの指標である関数の最大化，または最小化で行っている．そのため，進化計算を用いたパラメータサーベイ機構でも同様に結果の特徴をスコアとして表わすスコア関数を定義することで,実行順序の制御を行う.スコア関数の定義方法は，シミュレーションの実行結果として出力されるファイルからスコアの定義に必要となるパラメータを抽出し,そ のパラメータを用いて関数を定義することにより行う．以下にパラメータサーベイ機構の流れを述べる.

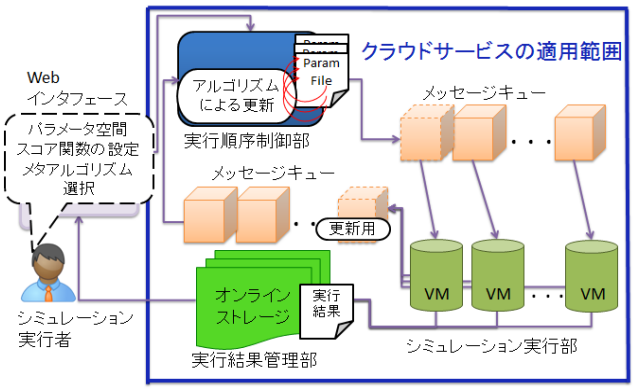

図４: パラメータサーベイ機構

（１） 対象パラメータ空間，スコア関数の決定
（２） ランダム生成での初期パラメータセット群の実行
（３）スコア関数を用いて各パラメータセットのスコアの計算
（４）スコアが良かったパラメータセットの近傍から再度パラメータセット群の決定
（５） 全てのパラメータ空間の実行が完了 or 終了条件を満たすまで（3）へ戻る パラメータサーベイ機構を用いる場合,利用者が行う入力は対象パラメータ空間とスコア 関数の決定である．利用者は望む結果がパラメータ空間のどの範囲に存在するかの検討がついている必要はない．計算機がアルゴリズムとスコア関数を基に，利用者が望む挙動

を示すと思われるパラメータセットから探索を行う．

4.3 GPGCloud

我々はシミュレーションの大量並行実行を実現するクラウド上のシミュレーションプラットフォーム GPGCloud [7]を開発してきた．本システムは組合せを効率良く探索するものであり，実装はクラウドサービスである Amazon Web Services を使用した．また，シミュレータはエージェントシミュレーションで実績のある NetLogo を採用した．

クラウド上のシミュレーション機構の概要を図５に示す．

Managerは最初に実行するパラメータセット群を生成する．初期パラメータセット群を生成したのち，生成した各パラメータセットをクラウド上の仮想計算機（VM）に渡し，対象モデルのシミュレータを用いて実行する．実行結果は次世代の生成に必要になるため，Managerへ返される．Managerでは，実行したパラメータセットが研究上有用であるかをスコア関数により評価し，その評価を基に，次に実行するパラメータセットを生成する．生成された次世代の各パラメータセットは初期集団と同様に，クラウド上のVMに送られたのち，シミュレータを用いて実行される．パラメータ空間全て，もしくは最適解に到達した時点でパラメータ空間探索を終了する．

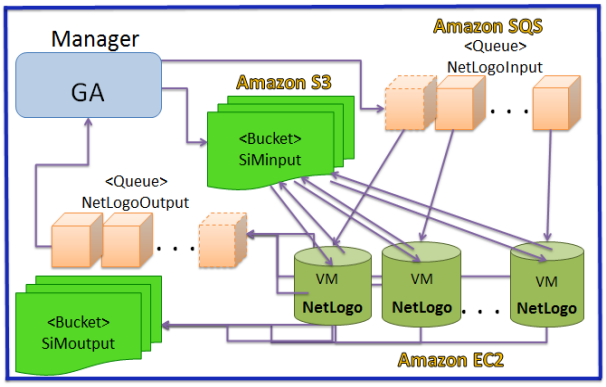

図５：クラウド上のシミュレーション機構

このシミュレーション機構ではパラメータ探索を行うメタアルゴリズムとして遺伝的アルゴリズム(GA)[4]が実装されており，パラメータを大量に生成し，研究上有用なパラメータセットを効率よく探索するという機能がある．また，クラウドサービスによるシミュレーション機構のためスケールアウト容易であり，VMを大量に生成し，並行実行することで効率の良いシミュレーションが可能となっている．

4.4 AWS

本実装において実行順序制御部と VM は AmazonEC2 により作動する．メッセージキューは Amazon SQS により作動する．オンラインストレージは Amazon S3 により作動する．シミュレーション実行部の VM では実行するシミュレーション によってシミュレータ等の設定が異なる事が考えられる．ま た，クラウドサービスは利用する時間に対して課金されるため 使用していない時は VM を停止しておくことが重要であるが，起動するたびに細かな設定をやり直すのは煩雑である．そこ でパブリッククラウドに幅広く対応しているクラウド運用管理 ツールである Right Scale を用いた．

Right Scale は VM の OS 等を設定する Server Templates と VM の起動, 稼働, シャットダ ウン時に実行可能な bash, ruby 等で記述する Right Script を備えている．シミュレーション実行時に指定されるモデル やパラメータサーベイアルゴリズムに応じて起動する VM の Server Templates と実行する． Right Scripts を選択することでそのシミュレーションの必要とする実行環境を自動的に構築する事が可能である．本実装では実行順序制御部と VM の Server Templates に Base Server Template -11H1-EBS-JP を設定した．シミュレーションモデル実行の ための設定を記述した Right Scripts を作成した．

Amazon SQS に基づき構築されたメッセージキューでは 1 つ のメッセージサイズは 64KB 以下に制限されている．そ こで, 今回の実装ではパラメータ等を直接メッセージキューを介して渡すのではなく，オンラインストレージ上に展開して URL を渡す事とした．まず実行順序制御部は，シミュレーショ ン実行用のパラメータファイルをオンラインストレージに公開設定でアップロードする．そして実行順序制御部はそのファイルの URL をメッセージとしてメッセージキューに送信し，VM はメッセージキューから取得した URL を基にオンラインストレージ上のパラメータファイルをダウンロードし, シミュ レーションを実行する．

4.5 エージェントモデル開発環境 Net Logo

Net Logo[8] は, Northwestern 大学の UriWilensky らによって開発された自然現象や社会現象をシミュレーションする為のプログラミング可能なモデル作成環境である．

オープンソースのソフトウェアとなっており Java 仮想マシン上で実行する．

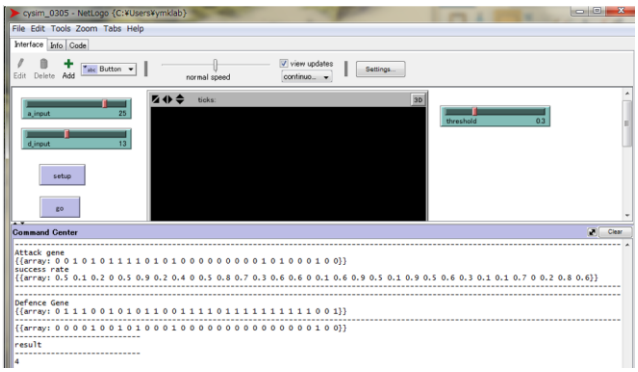

図６：NetLogo インターフェース

ミクロなルールによって 行動する多数のパーツによって，

より複雑なマクロな現象が引き起こされるようなモデルのシミュレーションを容易に行うことが出来る．

5. 遺伝的アルゴリズムにおける攻撃と防御

以下に遺伝的アルゴリズムに攻撃と防御のモデルを示す．

5.1 遺伝的アルゴリズムへの適応

- 個体表現

対象とする問題の解を記号列で表現する．

サイバー空間において攻撃側はあるエクスプロイトコードの選択を1，非選択を 0 で表せば，n種類の攻撃手法を持つ個体は，遺伝子の長さnのバイナリ記号列(1)で表現される．防御側はセキュリティ保護施策の選択を 1，非選択を 0 で表しm種類の防御手法を持つ個体は，

遺伝子の長さmのバイナリ記号列(2)で表現される．

$$\text{Attacker}\ (a_1, a_2, a_3, \ldots, a_{n-1}, a_n) = \mathbb{A} \quad (1)$$

$$\text{Defense}\ (d_1, d_2, d_3, \ldots, d_{m-1}, d_m) = \mathbb{D} \quad (2)$$

Attacker が攻撃個体でa はエクスプロイトコードあり，a_1 からa_nまで攻撃のセットを持つ．同様に Defense が防御個体でd はセキュリティ保護施策あり，d_1 からd_nまでのセキュリティ保護施策のセットを持つ．

- 戦略

攻撃側はエクスプロイトコードの選択した数 i，防御側はセキュリティ保護施策の選択した数を j,とし，攻撃の成功率 s，係数 k，手法を選択した場合のコストをc，情報資産(Resource) R とすると以下の式となる．

$$\text{標的価値} * \text{成功率} - \text{攻撃数} * \text{係数} = \sum a_i s_i - ik$$

$$\text{情報資産} - \text{防御対策コスト} = R - \sum c_j$$

エクスプロイトコードが対象とする攻撃の利得の大きさを標的価値とし，検知率として攻撃数* 係数とした．また，情報資産から防御施策のコストの合計を引いたものとする．攻撃と防御ともに少ない手数で最大の効果を生むように戦略を取っていく．

攻撃プログラムの組合せと脆弱性対策の組合せを基準として攻撃と防御共に，最小コストで最大の効果のある組合せを試行していくため，各々の戦略が相互に影響する．

防御側はセキュリティ保護施策を対策・回避・放置の３つに分けられる．防御側の方針を表１に示す．

表１：防御側の方針

	情報資産	コスト	リスク
対策	○	×	○
回避	×	○	○
放置	○	○	×

対策すれば情報資産は守られ，リスクも減るが脆弱性対策するコストがかかる．回避は情報資産を持たない，無くすと捉える．コストとリスクは下がる．放置は脆弱性対策を行わないためリスクが高くなる．

このような観点から防御側の方針を検討してゆく．

5.2 実行結果

攻撃モデルのシミュレーション結果を図７に示す．横軸にパラメータ世代数，縦軸に適応度を取った．このシミュレーションモデルでの最適値は適応度 1.0 である．

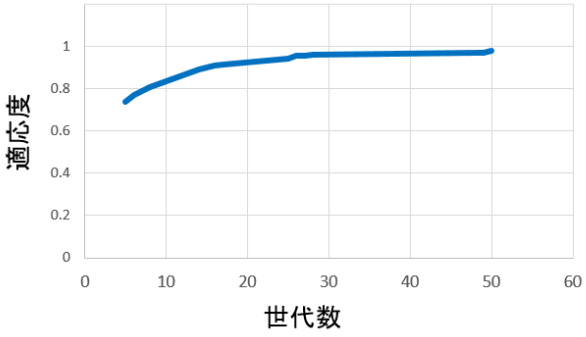

図７：攻撃側のシミュレーション結果

遺伝子配列は 64 ビットで行い，個体集団は 30 とした．個体集団全体の適応度の収束が約 50 世代であった．攻撃のみの結果は単調増加で遺伝的アルゴリズムによる進化が有効に行われたことを示す．

遺伝的アルゴリズムは与えられた問題を，ある遺伝子配列を持った種の進化として捉えて，３節で述べた手順を繰り返すことによって最終的に高い適応度を持った個体を求める．

5.3 攻撃と防御のモデル

実際の攻撃・防御の概略を図８に示す．

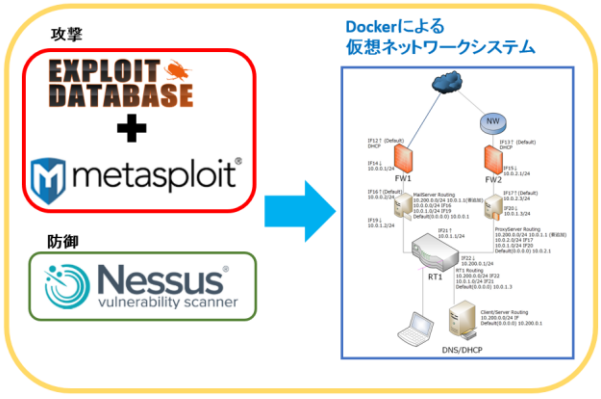

図８：攻撃・防御の概略図

これらはすべて AmazonEC2 上で動作するものである．攻撃側は Exploit DB と Metasploit を使用する。防御側は脆弱性スキャナの Nessus を使用する．シミュレーション対象は当研究室が開発している仮想ネットワークシステムは７節

において述べる．

攻撃個体のモデルは Exploit DB に公開されている Exploits by Metasploit を使用する．総数 1429 種の Exploit.xml から攻撃コードを参照する．

「Metasploit Framework」（以下、Metasploit）とは、攻撃コードの作成、実行を行うためのフレームワークソフトウエアである。Exploit Database の採番である EDB-ID から採番形式で検索して用いることが可能である．これらの攻撃コードが選択されて、ある組合せの攻撃個体，即ちマルウェアとして振る舞うことが可能である．

防御個体のモデルは防御施策として CVE を基準とした施策とする．

CVE(Common Vulnerabilities and Exposures)とは一つ一つの脆弱性を識別するための共通の識別子である．個別製品中の脆弱性に一意の識別番号「CVE 識別番号(CVE-ID)」を付与することにより、組織 A の発行する脆弱性対策情報と、組織 X の発行する脆弱性対策情報とが同じ脆弱性に関する対策情報であることを判断したり、対策情報同士の相互参照や関連付けに利用したりできる．

また，Nessus は、Tenable Network Security 社がコンピューター・セキュリティに関して開発した包括的な脆弱性検知スキャナである．

このツールの目的は、システム上の潜在的な脆弱性を検知することである．スキャンによって作成されたレポートは CVE を基準としている．

その検知された CVE のリストから遺伝子として構成する．CVE の対策を元にネットワークシステム全体においての防御施策とする．

Metasploit 上で CVE-ID を検索し，脆弱性をついて実際に検証・ログ収集していくことも可能である．さらに Metasploit では Nessus のプラグインも使用可能となっており攻撃と防御の相互から対策の検討をしていくことが可能である．

新種のウイルスに関しては仮の CVE を作り危険度や対象プラットフォームなどを模擬的に作成することを検討している．

6. 関連研究

2 節で述べたように進化計算を用いて攻撃手法を自動生成する研究はすでに存在する．Noreen らは GA を用いることで実在のマルウェアである．

Bagle ファミリに対して遺伝的アルゴリズムを適用し，Bagle ファミリの亜種の生成を行った．

また，Duchene は同じく GA を用いて Web アプリケーションに対する有効なクロスサイトスクリプティング(XSS)攻撃リクエストを生成する手法について検討している．

これらはいずれも，ある特定の攻撃対象について有効な攻撃パターンを生成するものであって，本校で述べた共進化のような，守備側が変化することまでを想定したものとはなっていない．

これら既存研究では，主導権は攻撃者側にあり，防御側はそれを検知することに主眼が置かれる．

これらに対する本研究の独自性は，標的型攻撃の戦略性に着目することで未出現の攻撃パターンを予測ないしシミュレーションして，先回りする形で防御対策につなげようとする点である．

7. 今後の進め方

本研究において今後，検証システムを構築しシミュレーションの評価をしていく．この検証システムは仮想計算機・仮想ネットワークによるシミュレータ構成[9]と攻撃監視システム[10]の２つについて述べる．

仮想計算機・仮想ネットワークによるシミュレータ構成は Docker, Open v Switch などオープン ソースの仮想システムを組み合わせることで，企業などのネットワークをシミュレートする．ネットワーク構成をクラウド上に展開するための自動化システムを構築する．

仮想ネットワークシステムとは実在する LAN を ソフトウェア上にて仮想的に実行する物である．仮想環境で行うことによって同時並行での検証が 可能になる．Amazon Web Services1 のクラウド上 に仮想マシンと仮想ブリッジを用いて仮想ネットワークの構築を実現した．

例を図９ に示す．

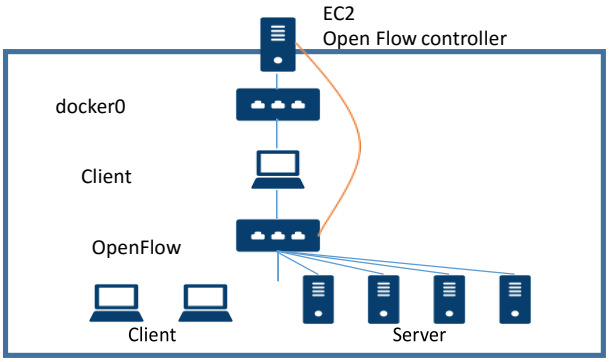

図９：仮想ネットワークシステムの例

インスタンス上に Open Flow controller（コントローラ）が動作し，インスタンス内に点線で囲った docker0（仮想ブリッジ），Open Flow switch（スイッチ），Server（サーバ），Client（クライアント）から構成される仮想ネットワークがある．Client，Server は Docker，Open Flow switch は Open v Switch ，Open Flow controller は Trema4 を用いて構成する．

仮想ネットワークシステムを用いて構築したネットワーク環境で攻撃と防御それぞれの有効性の検証を実現する

ためにネットワーク監視機能を構築している.
機能として,ネットワーク環境内に存在するホストの自動登録や監視項目の設定,監視結果の取得とデータの収集は監視対象のホスト毎に収集作業を行う.
仮想ネットワークシステムで使用できる攻撃監視専用の監視ツールがないため既存の監視ツールを採用している.

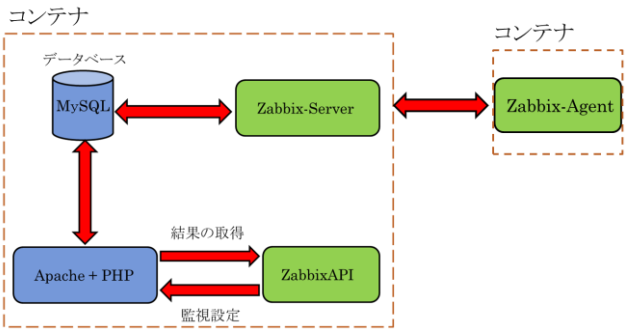

図１０：Zabbix のシステム構成

Zabbix のシステム構成を図１０に示す.
監視用コンテナ内では Zabbix API を使用したプログラムが設定や結果の取得を行う.Apache と PHP で構築されたフロントエンドは監視項目や監視結果が管理,保存されているデータベースと接続されている.また,Zabbix-Server とも連携しており,対象ホストに適用されている監視項目をデータベースから参照し監視対象コンテナの Zabbix-Agent に監視項目の実行を指示する.Zabbix-Agent は実行結果を Zabbix-Server に送信する.受信した結果をデータベースに保存する.

8. おわりに

マルウェアの進化に着目し，進化計算を用いたマルウェアを生成とその防御のモデル及びシミュレーションを検討してきた．今回， Exploit DB と Metasploit，脆弱性スキャナの Nessus 使用し，Exploit と CVE といった組合せ基準に攻撃内容と防御手法がお互いに適応して進化する「共進化」のモデルを示した．また，膨大な数の進化パターンを効率的にクラウド上で探索する進化計算機構を開発してきた．

進化計算の中で各エージェントを評価するにあたっては，ネットワークシミュレータを構築して，その中での攻撃と防御を模擬実行する．
我々は，実際のネットワークシステムを再現するような，ネットワーク構造を記述し，仮想環境内でネットワークシステム構築がクラウド上で実現している．また，その仮想ネットワークシステムにおいて攻撃を模擬実行し、どのようなログが残るか動作検証及び、攻撃判定するシステムがある．これらのシステムと本シミュレーション機構を統合することが今後の目標となる．
最終的にサイバー空間における攻撃と防御の共進化を大量に効率良く計算することで，その仮想環境内ネットワークシステムで起きうる攻撃や，それに対する合理的な施策の組合せが生成される．

謝辞

AI 技術のサイバーセキュリティ対策への応用について重要な示唆をいただいた東京電機大学・佐々木良一教授，並びに LIFT プロジェクトのメンバーに感謝いたします．本研究の一部はセコム科学技術振興財団の研究補助金により実施されました．

参考文献

[1] 独立行政法人情報処理機構セキュリティセンター,“『高度標的型攻撃』対策に向けたシステム設計ガイド,” 2014.
[2] 佐々木良一，八槇博史 “標的型攻撃に対する知的ネットワークシステム LIFT の開発(その 3)-今後の研究構想-” マルチメディア，分散協調とモバイルシンポジウム 2015 (DICOMO2015),
[3] Noreen, S. et al., “Evolvable Malware,” 11th Annual Conf. on Genetic and Evolutionary Computation, pp. 1569-1676, 2009.
[4] Duchene, F. et al, “Kameleon Fuzz: Evolutionary Fuzzing for Black-box XSS Detection,” 4th ACM Conf. on Data and Application Security and Privacy, pp. 37-48, 2014.
[5] Holand, J.H., “Adaptation in Natural and Artificial System”, University of Michigan Press, 1975.
[6] Barbosa, Helio JC. “A coevolutionary genetic algorithm for constrained optimization.” Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on. Vol. 3. IEEE, 1999.
[7] 吉田慎吾，八槇博史，岩崎裕太郎，“クラウドを用いたマルチエージェントシミュレーションの大量実行”，第 27 回人工知能学会全国大会予稿集，4F1-2, 2013.
[8] Tisue, S, & Wilensky. U., “Netlogo: A simple environment for modeling complexity.” International conference on complex systems. pp.16-21, 2004.
[9] 大石恵輔，八槇博史 ”サイバー攻撃実験のための仮想ネットワーク自動構成方式の検討”，電子情報通信学会 DS-2-4,2016
[10] 中山 能之，八槇博史 “仮想ネットワークシステムにおける自動攻撃と監視システムの実装”，電子情報通信学会 DS-2-5,2016