大規模環境における攻撃グラフを活用したセキュリティ対策立案方式

井ノ口　真樹[1,a)]　柳生　智彦[1,b)]

概要：近年、発電所などの重要インフラや工場を中心にサイバーセキュリティリスクを低減することがますます重要となっている。これらの施設はサイバー攻撃の被害を受けると、操業が停止することによる被害だけでなく、停電や物理的な誤動作による人的被害など多大な影響を及ぼす恐れがある。一方、これらの施設ではソフトウェアのアップデートや構成変更がシステムに悪影響を及ぼす可能性があるため、セキュリティ対策をとることが難しいという問題がある。それに対し攻撃グラフを用いてセキュリティ対策を立案する方法がある。攻撃グラフは、攻撃手順や攻撃に利用されるホストを可視化することができるため、少ないコストや変更箇所で、リスクを最小化する対策を立案可能となる。しかし、大規模な環境では攻撃グラフを生成するために必要な計算コストが非常に大きくなり、攻撃グラフを用いることが困難であった。本論文ではそれに対し、分析対象ホストを徐々に増やしていきながら複数サイクルに分けて対策立案を行う方式を提案する。本方式では、各サイクルで立案された対策が適用されたものとして次のサイクルの攻撃グラフ生成を行うことで、攻撃グラフ生成の所要時間を低減することを可能とする。14 台のホストが 3 つのサブネットワークに分かれて存在する環境を想定し、計算機を用いた実験で提案方式によって対策立案のために必要な攻撃グラフ生成の総所要時間が 98%軽減されることを確認した。

1. はじめに

近年、発電所などの重要インフラや工場がサイバー攻撃の被害を受ける事例が増加しており、サイバーセキュリティリスク対策の重要性はますます高まっている。例えば、核施設をターゲットとした Stuxnet や停電を引き起こした Black Energy3 などの事例が報告されている。重要インフラや工場はサイバー攻撃の被害を受けると数時間の操業停止であっても莫大な経済的損失が生じる。さらに、制御系システムの誤作動は物理的な機器の故障や人的な被害を引き起こす可能性がある。そのため、セキュリティリスクを把握し、適切な対策を施すことが極めて重要となる。

現在のセキュリティアセスメントは主に手作業で実施されており、セキュリティリスクを把握するための情報収集には様々なツールが存在する [1]。例えば、システムを構成するホスト情報を収集する Open-AudIT、脆弱性情報を収集するスキャナである OpenVAS や Nessus、空きポートと稼働しているネットワークサービス情報を収集する Nmap などがある。しかしながら、近年のサイバー攻撃は複数のホストにまたがる複数ステップの複雑な攻撃が行われるためホスト単体での情報収集では攻撃リスクの全体像を把握することが困難である。また、制御系システムはソフトウェアのアップデートや構成変更がシステムに影響を及ぼす可能性があり、セキュリティ対策をとることが難しいホストが存在する。これらのことから、単体のホストのリスクを把握するだけでなくシステム全体のリスクを把握し、少数のホストに対して適切な対策を実施することが求められる。

これに対し、攻撃グラフを用いてセキュリティ対策を立案する方法がある。攻撃グラフを用いることで複数のホストを経由するような攻撃を可視化することが可能となる。攻撃グラフは、攻撃を実行するための条件や攻撃によって生じる結果をノードとして表し、攻撃の結果を次の攻撃の条件とすることで、複数ステップの攻撃を表現する。攻撃グラフを用いることで、攻撃のために必要な条件、攻撃に使われる通信プロトコル、経由されるホストなどの情報が可視化される。そのため、少ないコストや変更箇所でセキュリティリスクを最小化する対策を立案することが可能となる。攻撃グラフは前述の情報収集ツールによって得られるアセスメント対象システムの情報をもとに、MulVAL[5] などのツールを用いて自動生成することが可能である。一方、攻撃グラフの生成は計算コストがかかるため、大規模環境で攻撃グラフを用いて対策を立案することは困難で

[1] NEC セキュリティ研究所
[a)] m-inokuchi@bu.jp.nec.com
[b)] yagyu@cp.jp.nec.com

あった。

本論文ではそれに対し、分析対象ホストを徐々に増やしていきながら複数サイクルに分けて対策立案を行う方式を提案する。本方式では、各サイクルで選択された対策が適用されたものとして次のサイクルの攻撃グラフ生成を行う。こうすることで各サイクルにおいて、それまでのサイクルで適用を仮定された対策によって攻撃パスが消滅するため、攻撃グラフのサイズが小さくなるとともにグラフ生成に要する時間を短縮することが期待される。

14 台のホストが 3 つのサブネットワークに分かれている環境をアセスメント対象として想定し、実際に対策立案を行わせて評価した。その結果、対策立案のために必要な攻撃グラフ生成の総所要時間が 98%軽減されることを確認した。

2. 関連研究

2.1 攻撃グラフ

攻撃グラフは攻撃条件や攻撃ステップを可視化したグラフである。Phillips[2]、Swiler[3] らがその概念を提案して以降、様々な攻撃グラフが提案されている。なかでも AND または OR の論理構造を各ノードの入力辺にもつ Directed Acyclic Graph (DAG) や木構造のグラフが最も広く利用されている [4]。このような攻撃グラフでは各ノードは攻撃者の状態やシステム状態を示し、子ノードは親ノードが成立する条件となる。根が攻撃者の目標を示し、葉はアセスメント時に想定する初期状態を示す。初期状態から可能な攻撃によって到達する状態が中間状態となり、葉ノードの親ノードとなる。以降中間状態が次の中間状態の条件となることで、DAG の構造を持つグラフが構成される。この時、初期状態から攻撃者の目標状態までに至るパスを攻撃パスと呼ぶ。

図 1 は攻撃グラフの例である。この例では、初期状態で攻撃者はホスト A 上で任意のコードを実行可能であり、目標はホスト B で管理者権限でコード実行可能となることである。攻撃者はまずリモートから攻撃可能な脆弱性 X をついて一般権限でログイン可能となる。その後、ホスト B の特権昇格の脆弱性 Y をつくか、ホスト B の管理者のクレデンシャルを取得することで、ホスト B 上で管理者権限でコード実行可能になることが示されている。

このようにして構成された攻撃グラフのノードをスコアリングすることで、リスクの高い攻撃パスを把握することが可能となる。例えば、ベイジアンネットワークの技術を用いてスコアを計算する方法が提案されている [7][8]。

攻撃グラフを活用することで、効率的な対策の立案が可能となる。例えば、文献 [13] では攻撃グラフを用いて対策の効果を算出し、対策立案を行う方式が提案されている。攻撃グラフはシステム全体のなかで攻撃によく使われるホストや通信プロトコル、ソフトウェアなどの情報を含むため、低いコストや少ない変更箇所で効率的にリスクを低減する対策を立案することができる。

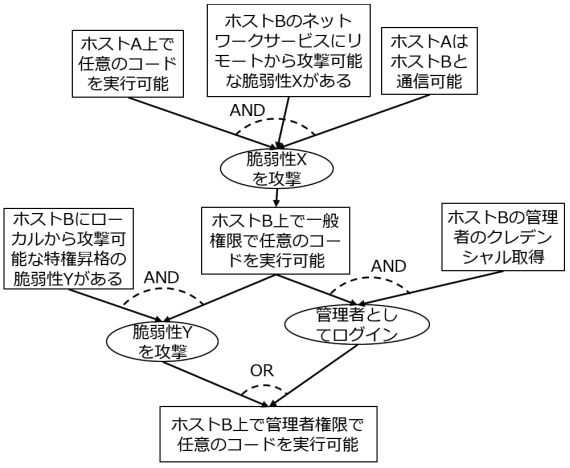

図 1 攻撃グラフの例

なお、DAG や木構造の攻撃グラフは攻撃ツリーと呼ばれることもある。またセキュリティアセスメントで使用されるフォルトツリーは本質的には本節で説明した攻撃グラフと同様のものである。

2.2 攻撃グラフ生成ツール

攻撃グラフを自動で生成する様々なツールが提案されている [9]。ここではその中でも高い拡張性を持つ攻撃グラフ生成フレームワークである MulVAL[5] について説明する。MulVAL は演繹推論により攻撃グラフを生成するツールである。MulVAL は環境情報などを記述したファイルとインタラクションルールと呼ばれるシステム要素の関係や攻撃に関する知識を一階述語論理で記述したファイルをもとに攻撃グラフを生成する。前者のファイルにはホスト情報やネットワーク構成などの環境情報に加え、アセスメントの前提（攻撃目標や攻撃者の初期位置など）が述語として記述される。MulVAL フレームワークに含まれるインタラクションルールを編集することで、扱える攻撃やシステムエンティティを適宜拡張することができる [12]。

例えば、図 1 と同等の攻撃グラフを MulVAL によって得る場合、「攻撃者がホスト A に初期状態で存在する」、「ホスト A、ホスト B が通信可能である」、「ホスト A にリモートからのコード実行を許す脆弱性 X が存在する」、「ホスト B に特権昇格の脆弱性 Y が存在する」、「攻撃者はホスト B の管理者のクレデンシャルを持つ」といった情報が環境情報として記述される。一方、インタラクションルールとしては攻撃する条件（脆弱性 X を攻撃するためにはリモートからアクセス可能など）と攻撃の結果（一般権限でコード実行可能など）といった関係性が記述される。

2.3 攻撃グラフ生成の計算コストと効率化手法

大規模環境における攻撃グラフ生成の計算は、攻撃モデ

ルやシステムモデル、システム構成、グラフ生成アルゴリズムに依存するが、一般には多くのコストを要する。文献 [6] では MulVAL を用いた評価を実施しており、全ノードが互いに通信可能な構成で、グラフサイズは対象システム内のホスト数に比例し、計算時間はホスト数 n に対して $O(n^2)$〜$O(n^3)$ 程度の計算量となった。

少ない計算コストで攻撃グラフを生成する方式も検討されている [10][11]。これらの方式ではシステムを複数のサブシステム分けサブシステムごとに攻撃グラフを生成することで、攻撃グラフ生成に要する計算コストを低減する。しかし、サブシステムに分割する場合に、サブシステムをまたがる攻撃パスが多い場合には計算コストを低減することができない。また、アクティブディレクトリやファイアウォール、DNS などの、システムを管理する役割を有するホストへの攻撃は、サブシステムを超えて影響をおよぼす可能性があるため、サブシステムに分割した場合にその影響を把握することが困難となる。

本稿で提案する方式は、初期に少数のホストで攻撃グラフを生成する点でこれらの手法と類似しているが、最終的には全てのホストを分析対象とするため、ファイアウォールやアクティブディレクトリなどへの攻撃も扱うことが可能となる。

3. 大規模環境向け対策立案方式

3.1 対策立案方式の動作

提案方式の動作について説明する。提案方式の基本的なアイデアを説明するために提案方式の挙動の例を図 2 に示す。提案方式における対策立案は複数サイクルに分けて実現される。各サイクルでは次の 3 つステップに分けて処理を行う。

(1) 当該サイクルで分析対象とするホストを選択
(2) 分析対象としたホストについて攻撃グラフを生成
(3) 生成された攻撃グラフについて対策を選択

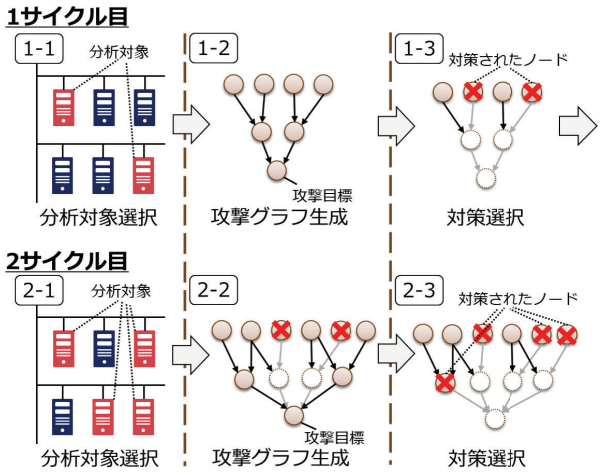

図 2　提案方式の挙動の例

提案方式は各サイクルにおける攻撃グラフのサイズを小さく抑えることでグラフ生成に要する時間を抑制する。図 2 の 1 サイクル目では、分析対象となるホスト数が少ないため、処理 1-2 における攻撃グラフサイズは全てのホストを分析対象とした場合と比べ小さくなることが期待される。また処理 1-3 で選択された対策が実施されたものとして 2 サイクル目の攻撃グラフ生成が行われる。すなわち処理 2-3 で生成される攻撃グラフは処理 1-3 で選択された対策によって条件が成立しなくなったノードの分だけグラフサイズが小さくなる。そのことから、ホスト数を増加させた場合でも各サイクルのグラフサイズが小さく抑えられることが期待される。提案方式の動作はリスト 1 のとおりである。以降、各処理について詳細に説明する。

なお、本方式では以下の情報が事前にユーザによって与えられているものとしている。

- アセスメント対象の環境情報
- 攻撃者の初期に侵入しているホスト
- 攻撃目標
- 当該システムで実施可能な対策の集合

リスト 1 提案方式の動作

Input: アセスメント対象の環境情報 Env_{all}, 攻撃者が初期状態で存在するホスト h_S, 攻撃目標ホスト h_G, 実施可能な対策集合 C_{all}
Output: 立案された対策 $C_{selected}$
1: 分析対象ホスト集合 $H_{selected}$ に攻撃者が初期状態で存在するホスト h_S と攻撃目標ホスト h_G を追加
2: 環境情報 Env_{all} に含まれるホストを H_{remain} に追加
3: **while** $H_{remain}! = \phi$ **do**
4: 分析対象 $H_{selected}$ に関連する環境情報を抽出し、分析対象環境 Env_{target} に格納
5: 選択済み対策群 $C_{selected}$ で消去される環境情報を Env_{target} から削除
6: Env_{target} を用いて攻撃グラフ G を生成
7: 攻撃グラフ G の攻撃リスクを最小化する対策を選択し、$C_{selected}$ に追加
8: H_{remain} からランダムに N_h 台のホストを取り出し、$H_{selected}$ に追加
9: **end while**

3.1.1 分析対象とするホストの選択

最初のサイクルにおける分析対象とするホストの選択では、攻撃の達成のために必ず利用されるホストを最低限分析対象として選択する。すなわち攻撃者が初期状態において存在するホストや、攻撃目標となるホストが分析対象として加えられる（リスト 1 の 1 行目）。また、システム構成や運用上、初期位置から攻撃目標に到達するために必ず経由するホストが分かる場合、それらのホストも分析対象とする。

2 サイクル目以降のホストの選択では、それまでに分析対象として選択されていないホストの中から各サイクルでのホストの増分 N_h だけランダムにホストを選択し、分析

対象に加える（リスト 1 の 8 行目）。すなわち、提案方式では分析対象とするホストは単調に増加する。

3.1.2 攻撃グラフの生成

各サイクルにおいて、分析対象として選択されたホストに関連する環境情報のみを用いて攻撃グラフを生成する（リスト 1 の 6 行目）。例えば、分析対象ホストが持つ脆弱性や稼働するサービスなどの情報や分析対象ホスト同士の間で通信可能かといった情報が分析に用いられる。

2 サイクル目以降の攻撃グラフの生成では、それまでのサイクルで選択された対策が実施されたものと仮定して攻撃グラフの生成を行う。例えば 1 サイクル目の対策立案処理において「ホスト A でパッチを適用し脆弱性 X を取り除く」対策が選択された場合、2 サイクル目以降の攻撃グラフの生成ではホスト A に脆弱性 X が存在するという情報を取り除いて攻撃グラフ生成を行う（リスト 1 の 5 行目）。このようにすることで、それまでのサイクルで選択された対策によって攻撃パスの数が減るため、ホストを増加させても各サイクルでの攻撃グラフサイズは小さく抑えられることが期待される。

3.1.3 対策の選択

各サイクルにおいて、前ステップで生成された攻撃グラフを用いて、攻撃のリスクを最小化するように対策を選択する（リスト 1 の 7 行目）。本方式では各サイクルで生成される攻撃グラフサイズは小さいものとなることが期待されることから、本ステップでは攻撃のリスクを最小化する対策の組み合わせのうち、対策コストが最も小さいものを総当たりで見つけるものとする。対策コストとしては、対策箇所の数や各対策を導入、試験するための人的コストやセキュリティ対策製品の金銭的コストなどを用いる。

分析対象にすべてのホストが加えられている場合、これまでに選択された全ての対策を立案された対策として出力し処理を終了する。

4. 実機評価

4.1 評価環境

提案方式の評価を行うため、実機を用いて攻撃グラフ生成に要する計算時間を調査した。本評価では制御系のシステムを有する企業ネットワークを想定し、図 3 に示すような、3 つのサブネットワークに分かれた環境を採用した。この構成では、事務系の処理を行う Enterprise Zone と、制御系の管理を行う Supervisory Zone、フィールドデバイスや保守用端末が存在する Field Network Zone が存在することを想定している。

Enterprise Zone にはホスト H_E^1〜H_E^4、Supervisory Zone にはホスト H_S^1〜H_S^4、Field Network Zone にはホスト H_F^1〜H_F^4 が存在する。攻撃者は初期に Enterprise Zone 内のホスト H_E^1 に侵入しているものとし、攻撃目標を Field Network Zone の特定のホスト H_F^1 とする。また、各サブネットワークの間にはファイアウォールが存在するものとし、Enterprise Zone のホスト H_E^2 からのみ Supervisory Zone のホストと通信可能、Supervisor Zone のホスト H_S^2 からのみ Field Network Zone のホストと通信可能となるように設定されているものとする。また、Enterprise Zone のホスト H_E^3 を Enterprise Zone と Supervisory Zone 間のファイアウォール（F_{ES}）の管理端末とし、Supervisory Zone のホスト H_S^3 を Supervisory Zone と Field Network Zone 間のファイアウォール（F_{SF}）の管理端末とする。すなわちこれらの端末に侵入することでファイアウォールにログインし、ファイアウォールの規則を書き換えることが可能となる。

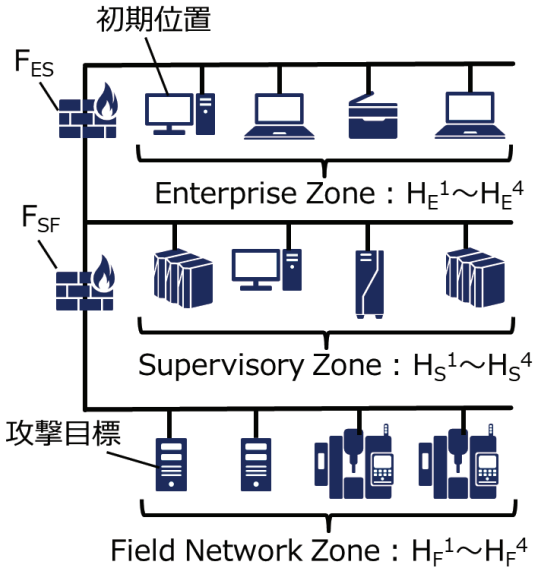

図 3 3 サブネット構成

また、簡単のため全てのホストにリモートから任意のコードを実行可能な脆弱性が存在するものとした。すなわち、本環境において攻撃者が Supervisory Zone のホストを攻撃するためには、1) ホスト H_E^2 に侵入し Supervisory Zone にアクセスするか、2) ホスト H_E^3 に侵入しファイアウォールの規則を書き換えたのち Enterprise Zone の任意のホストから Supervisory Zone のホストにアクセスする必要がある。同様に Field Network Zone のホストを攻撃するためにはホスト H_S^2 またはホスト H_S^3 を攻撃する必要がある。

対策としてはソフトウェアをアップデートすることで脆弱性をなくす対策のみを考慮する。一般にフィールドネットワークに近いほどソフトウェアアップデートが困難なケースが多くなることから対策のコストは Enterprise Zone のホストを最も低く、次に Supervisory Zone のホストを低く設定し、Field Network Zone のホストの対策コストを最も高く設定した。すなわち、本環境で最も低いコストで攻撃リスクを最小化する対策は「ホスト H_E^2 および H_E^3 のソフトウェアをアップデートすること」となる。

本評価では攻撃グラフ生成ツールである MulVAL を用いて攻撃グラフを生成した。ただし、デフォルトの MulVAL ではファイアウォールの規則を書き換える攻撃を表現できないことから、ファイアウォールにログイン可能となると、当該ファイアウォールがフィルタするサブネット間の任意の通信が許可されるような攻撃ナレッジを追加する改修を行った。なお、評価に用いた計算機の仕様は表 1 の通りである。

表 1 評価に用いた計算機の仕様

CPU	Intel(R) Core(TM) i7-3687U CPU @ 2.10GHz
コア数	2
メモリ	8GB
OS	Ubuntu 16.04.5

4.2 評価結果

攻撃グラフ生成に要した時間を表 2 に示す。なお、提案方式は各サイクルでの分析対象ホストの増分 N_h を 1 とした場合について評価した。また、所要時間は各サイクルでの攻撃グラフ生成に要した時間の和であり、分析対象ホストの選択や対策選択に要した時間は含めていない。

なお、最初のサイクルでは H_E^1 と H_F^1 のみを分析対象とし、以降、本試行では以下の順で各ホストが分析対象に加えられた。

$$H_F^2,\ F_{SF}\ ,H_S^2,H_S^4,H_F^3,F_{ES},H_E^3,H_E^4,H_E^2,H_F^4,H_S^3,H_S^1$$

表 2 に示されるとおり、提案方式では、一度にすべてのホストを分析対象とする従来の方式と比べ攻撃グラフ生成に要する時間を 98% 程度減少させた。

提案方式の各サイクルごとの所要時間と生成される攻撃グラフのサイズ（ノード数）を表 3 に示す。また、提案方式との比較のため従来方式でのグラフサイズ（全ホスト）及び、提案方式における 10 サイクル目の分析対象であった 11 ホストについて従来方式で攻撃グラフ生成を行った場合のグラフサイズを表 4 に示す。

提案方式における 1〜7 サイクル目までの試行では、分析対象とのホストだけでは攻撃目標に到達できないため、攻撃グラフは出力されず、攻撃グラフ生成は即座に終了する。8 サイクル目で H_E^3 が分析対象となった時に、一つの攻撃パスが成立し攻撃グラフが生成された。このとき、それまでのサイクルで分析対象として選択されたホストのみで構成される攻撃グラフであるため、グラフサイズが小さく、短時間で生成することが可能となった。10 サイクル目で H_E^2 が分析対象に加えられた時、二つ目の攻撃パスが成立した。このとき生成された攻撃グラフのサイズは表 4 に示される従来方式で同じホスト群を分析対象とした場合と比較し半分のサイズとなっている。提案方式では 8 サイクル目での対策を考慮して攻撃グラフが生成されるため、攻撃グラフサイズが小さく抑えられる。このように、提案方式では分析対象ホストを増やしてもそれまでのそれまでのサイクルで選択された対策によって攻撃パスが消滅しグラフサイズが抑制されるため、短時間での攻撃グラフ生成が可能となる。また、本試行では最適解と同じく H_E^2、H_E^3 にてソフトウェアをアップデートするという対策が選択された。

表 2 攻撃グラフ生成の所要時間 (CPU 時間 [s])

従来方式	83.528
提案方式	1.884

表 3 攻撃グラフ生成の所要時間 (CPU 時間 [s]) とグラフサイズの推移

サイクル	グラフ生成時間 [s]	グラフサイズ
1	0.072	0
2	0.084	0
3	0.060	0
4	0.080	0
5	0.080	0
6	0.076	0
7	0.072	0
8	0.188	128
9	0.088	0
10	0.204	102
11	0.092	0
12	0.080	0
13	0.060	0

表 4 従来方式における攻撃グラフサイズ

全ホスト	509
11 ホスト	203

4.3 考察

本評価では、隣接するサブネットワークへと侵入する攻撃パスが 2 通りであったため、攻撃グラフ生成が行われたサイクルも 2 回であった。攻撃パスのバリエーションが増えるにしたがって、攻撃グラフ生成が行われるサイクルの数も増えると考えられる。多くの攻撃パスが存在する環境ほど、グラフ生成の計算コストが増加することから、提案方式の時間短縮の効果が高まると考えられる。

一方、今回立案された対策は最適な対策と一致するものであったが、環境によっては提案方式で立案される対策が最適でなくなる可能性がある。例えば Enterprise Zone から Supervisory Zone へと侵入する攻撃パスに多くのバリエーションがあり、Supervisory Zone から Field Network Zone へと侵入する攻撃パスが少数であるようなケースで

は、システムの全ホストを考慮すると Supervisory Zone から Field Network Zone への侵入を防ぐような対策が低コストとなりうる。しかし、提案方式では各サイクルの対策立案ではシステムの一部のホストのみが考慮されるため、Enterprise Zone から Supervisory Zone へ侵入する攻撃を防ぐ対策が各サイクルで選択され、結果として効率の悪い対策が選ばれる可能性がある。ただし、今回評価で想定した環境は最適な対策が自明であったが、一般には最適な対策を探索するコストは無視できない。総当たりで対策を選択する場合、対策可能な箇所 n に対して対策案のバリエーションは各対策を採用するか否かを考慮した 2^n 通りあるため、従来方式のように大きいサイズの攻撃グラフを扱う場合には最適な対策を立案することは多大な計算コストを要する。現実的には準最適なアルゴリズムを用いることになると考えられる。以上のことから、今後の発展として、本方式をより正確に評価するためには従来方式、提案方式ともに対策立案まで含めて所要時間と対策の最適性を評価する必要がある。

5. まとめ

本稿では大規模環境向けの攻撃グラフを用いたセキュリティ対策立案方式を提案した。攻撃グラフを用いることで低い導入コストでリスクを最小化する対策を立案することが可能となるが、大規模環境では攻撃グラフの生成自体が困難であるという問題があった。本稿で提案する方式では複数サイクルに分けて攻撃グラフ生成と対策立案を行うことで、各サイクルの分析の規模を小さく抑え、短時間での分析を可能とする。ホスト 14 台の環境をアセスメント対象とした場合で評価し、提案方式で対策立案のために必要な攻撃グラフの計算時間を大幅に短縮しうることを確認した。また、今後の課題としては、対策立案まで含めた所要時間と対策の最適性の評価があげられる。

参考文献

[1] Yien Wang and Jianhua Yang. 2017. Ethical Hacking and Network Defense: Choose Your Best Network Vulnerability Scanning Tool. In 2017 31st International Conference on Advanced Information Networking and Applications: Workshops (WAINA). IEEE.

[2] Cynthia Phillips and Laura Painton Swiler. 1998. A graph-based system for network-vulnerability analysis. In Proceedings of the 1998 workshop on New security paradigms. ACM, 71-79.

[3] Laura P Swiler, Cynthia Phillips, David Ellis, and Stefan Chakerian. 2001. Computer-attack graph generation tool. In discex. IEEE, 1307.

[4] Barbara Kordy, Ludovic Pietre-Cambacedes, and Patrick Schweitzer. 2014. DAG based attack and defense modeling: Don’t miss the forest for the attack trees. Computer science review 13 13 (2014), 1-38.

[5] Xinming Ou, Sudhakar Govindavajhala, and Andrew W Appel. 2005. MulVAL: A Logic-based Network Security Analyzer. In USENIX Security Symposium, Vol. 8.

[6] Xinming Ou,Wayne F Boyer, and Miles A McQueen. 2006. A scalable approach to attack graph generation. In Proceedings of the 13th ACM conference on Computer and communications security. ACM.

[7] Nayot Poolsappasit, Rinku Dewri, and Indrajit Ray. 2012. Dynamic security risk management using bayesian attack graphs. IEEE Transactions on Dependable and Secure Computing 9, 1 (2012), 61-74.

[8] Peng Xie, Jason H Li, Xinming Ou, Peng Liu, and Renato Levy. 2010. Using Bayesian networks for cyber security analysis. In Dependable Systems and Networks (DSN), 2010 IEEE/IFIP international conference on. IEEE, 211-220.

[9] Shengwei Yi, Yong Peng, Qi Xiong, Ting Wang, Zhonghua Dai, Haihui Gao, Junfeng Xu, JitengWang, and Lijuan Xu. 2013. Overview on attack graph generation and visualization technology. In Anti-counterfeiting, security and identification (asid), 2013 IEEE international conference on. IEEE, 1-6.

[10] Jajodia, Sushil, and Steven Noel. ”Topological vulnerability analysis.” Cyber situational awareness. Springer, Boston, MA, 2010. 139-154.

[11] 島邉 遼佑, 浅井 健志, 河内 清人, ルールベース推論型アタックツリー自動生成ツールにおける効率的な生成方式の提案, SCIS 2019

[12] Inokuchi, Masaki, et al. ”Design Procedure of Knowledge Base for Practical Attack Graph Generation.” Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security. ACM, 2019.

[13] Poolsappasit, Nayot, Rinku Dewri, and Indrajit Ray. ”Dynamic security risk management using bayesian attack graphs.” IEEE Transactions on Dependable and Secure Computing 9.1 (2012): 61-74.