Computer Security Symposium 2020
26 - 29 October 2020

イベント依存モデルの定義と それを用いた被害予測システムの実装

富家 琴音 [1,*]　古門 良介 [1]　白石 善明 [1]　森井 昌克 [1]

概要：システムやネットワークに対する不正アクセスの検知のために IDS（侵入検知システム）が広く用いられる．しかし，IDS は不正アクセスを検知するのみであり，被害予測や事前の対処を行うものではない．先に，被害予測のためにイベント依存モデルが提案された．IDS のシグネチャ情報をもとに依存関係を構築し，検知された不正アクセスから派生する可能性のある不正アクセスを絞り込み，次に受ける可能性のある不正アクセスを予測できることが示された．本稿では，その依存関係の新たな定義の提案とグラフデータベースを用いた実装を行った．

キーワード：不正アクセス，IDS（侵入検知システム），被害予測，依存モデル

Definition of Event Dependent Model and Implementation of Damage Prediction System of Unlawful Access Using It

Kotone Tomiie[1,*]　Ryosuke Kokado[1]　Yoshiaki Shiraishi[1]　Masakatu Morii[1]

Abstract: IDS (Intrusion Detection System) is widely used to detect unauthorized accesses to systems and networks. However, IDS only detects unauthorized accesses, and does not provide damage prediction or proactive measures. Previously, an event dependent model was proposed for damage prediction. It was shown that the dependencies can be constructed based on the IDS signatures, and that the unauthorized access that may be derived from the detected unauthorized access can be predicted. In this paper, we proposed a new definition of the dependency and implemented them using a graph database.

Keywords: unauthorized access, IDS (Intrusion Detection System), damage prediction, dependent model

1. はじめに

IDS(侵入検知システム:Intrusion Detection System) とは，システムやネットワークに設置され，不正アクセスやその兆候を検知した場合にアラートを出力して管理者に知らせるシステムである．IDS は不正アクセスへの防御として，侵入防止システム（Intrusion Prevention System:IPS）やファイアウォール，アンチウイルス，VPN などと併せて広く導入されている．

一般に，IDS を利用して不正アクセスによる被害を防ぐためには，ローカルドメインのネットワーク管理者にセキュリティに関する高度な専門知識が要求される．第一の理由は，IDS はあくまで不正アクセスやその兆候を検知するのみであり，適切な対処は管理者が行う必要があるからである．第二の理由は，一般に IDS は大量のアラートを出力するため，その中から対処が必要なアラートを管理者が見つけ出す必要があるからである．第三の理由は，アラート同士の関連性を IDS は考慮しないからである．このように，管理者には高度な知識や経験が要求されるため，ローカルドメインのネットワーク管理者は IDS を正しく運用できていない場合が多くある．このような問題点を解決する手段として，ローカルドメインに設置した IDS の遠隔監視サービス [1] や被害解析支援システム [2] などが存在する．ローカルドメインで何らかのインシデントが発生し，IDS が不正アクセスを検知した場合，IDS の遠隔監視サービスではサービス提供者がアラートを受け取る．そしてローカルドメインの管理者に代わってアラートを分析し，インシデントに関する報告書を作成する．被害解析支援システムではそのシステムが IDS のアラートを受け取り，ローカルドメインの管理者に代わって被害の検出，原因の特定および対策の提示を行う．

遠隔監視サービスや被害解析支援システムを用いることで，管理者はある不正アクセスに対して事後の対処を行うことができる．しかし，これらのシステムを用いても，今後発生しうる不正アクセスを予測し，未然に防ぐことはできない．そのため，ある不正アクセスを起点としてその後発生しうる被害を予測し，未然に対処することによって不正アクセスの被害を最小限に抑えることが必要とされている．

1 神戸大学
Kobe University
* tomiie@stu.kobe-u.ac.jp

© 2020 Information Processing Society of Japan

ある不正アクセスからその後発生しうる被害を予測するため，イベント依存モデルが提案された [3]. イベント依存モデルは，不正アクセスによって起こるイベントとその不正アクセスが起こる条件であるトリガに着目し，連鎖的に次に起こりうる不正アクセスを予想するものである．イベント依存モデルを使うことで，ローカルドメインのネットワーク管理者は被害を抑えるための対策方法を容易に得ることができる．そして，得た対策を実施することで，不正アクセスに対する事前予防が可能となる．しかし，本研究で用いる現在の IDS では，不正アクセスによって起こるイベントとその不正アクセスが起こる条件であるトリガを従来の方法で定義することが困難である．

本研究では，アラートごとのトリガとイベントの定義を現在の IDS でも行うことができるようなイベント依存モデルの定義を提案，実装し，その有効性を示す．本システムを用いることで，ローカルドメインのネットワーク管理者は出力された IDS のアラートから次に起こりうる不正アクセスのアラートを予想することができ，被害をおさえるための対策方法を得ることが容易になる．2 章で従来のイベント依存モデルとそれを用いた被害予測システムについて述べる．3 章で新しいイベント依存モデルの定義と実装について述べ，4 章でイベント依存モデルの動作例を紹介する．

2. 従来のイベント依存モデルと被害予測システム

2.1 従来のイベント依存モデル

イベントを不正アクセスによって起こる状態の変化であると定義する．図 1 のような，ある不正アクセスによって発生したイベントがさらなる不正アクセスのトリガとなり，新たなイベントが発生するような不正アクセスの連続性に着目する．ある不正アクセスのイベントが別の不正アクセスのトリガの条件を満たすとき，この 2 つの不正アクセスの関係を依存関係と呼ぶ．IDS のアラートから得られる不正アクセスの痕跡をもとに依存関係を構築する．イベントの依存関係を知ることにより，多数存在する不正アクセス手法の中から次の不正アクセスを絞り込み，その結果被害を予測することが可能となる．

トリガとイベントの定義について述べる．攻撃者がある不正アクセスを実施するために，攻撃対象システムでの権限，アカウント情報などの情報を知っていること，そして攻撃対象の環境といった条件が必要な場合がある. よって，不正アクセスが行われる条件であるトリガは，権限，取得済み情報，環境（OS, Software, Port, 所要ツール）の組であると定義する．攻撃によって起こる状況として，攻撃者の権限の上昇や情報の取得が考えられる．よって，不正アクセス実施後に起こるイベントは権限，環境（OS, Software, Port, その他の情報）の組であると定義する．

イベント依存モデルのルールについて述べる．あるアラートを A_i，そのアラートのトリガを T_i，イベントを E_i とする．トリガとイベントの各要素は，権限：*auth*，OS：*os*，Software：*soft*，Port：*port*，所要ツール；*tool*，取得済み情報：*info*，そしてその他の情報：*other* とする．このとき，$T_i = \{ T_{i.auth}, T_{i.info}, T_{i.os}, T_{i.soft}, T_{i.port}, T_{i.tool}\}$，$E_i = \{ E_{i.auth}, E_{i.os}, E_{i.soft}, E_{i.port}, E_{i.other}\}$ と表すことができる. A_i と A_j について，次の式(1)の関係を満たすとき，依存関係があると定義する．A_i が依存元，A_j が依存先である．

$$E_i \supseteq T_j \tag{1}$$

以下の式(2)～(6)の条件をすべて満たした場合，式(1)が成り立つとする．

$$E_{i.auth} \supseteq T_{j.aith} \tag{2}$$

$$E_{i.os} \supseteq T_{j.os} \tag{3}$$

$$E_{i.soft} \supseteq T_{j.soft} \tag{4}$$

$$E_{i.port} = T_{j.port} \tag{5}$$

$$E_{i.other} \supseteq T_{j.tool} + T_{j.info} \tag{6}$$

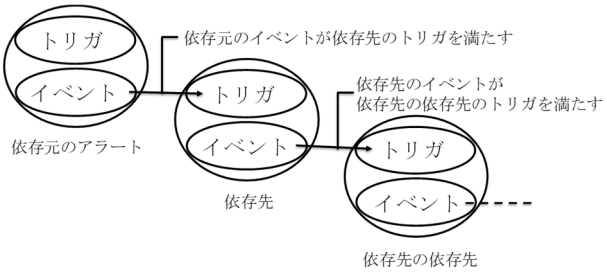

図 1　不正アクセスの連続性

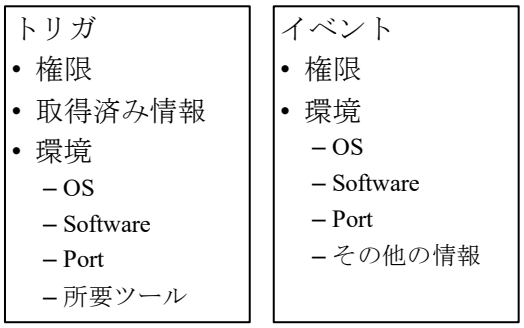

図 2　従来のトリガとイベントの定義

2.2 従来の被害予測システム

図 3 は被害予測システムの構成と処理の流れを示している．被害予測システムは登録部，アラートログ DB と予測部から構成される．被害予測システムを単一サイト内で用いる場合，IDS をサイト内に設置しなければならずネットワーク管理者にとって負担が大きい．なぜなら，IDS の能力を最大限発揮するためにはシグネチャを常に最新のものに更新し続けなければならないからである．この問題を解決するために，監視形態をセンター集中型にすることで複数のサイトを監視する．

数種類のネットワークベース IDS が出力したアラートは被害予測システムの登録部に送られる．登録部では各アラートから送信元 IP などの通信に関する情報が抜き出され，アラートログ DB に登録される．予測部はイベント依存モデルを用いアラートログ DB に登録された情報から被害を予測する．そして，予測した被害とその対策を含めた報告書を監視サイトの管理者に通知する．

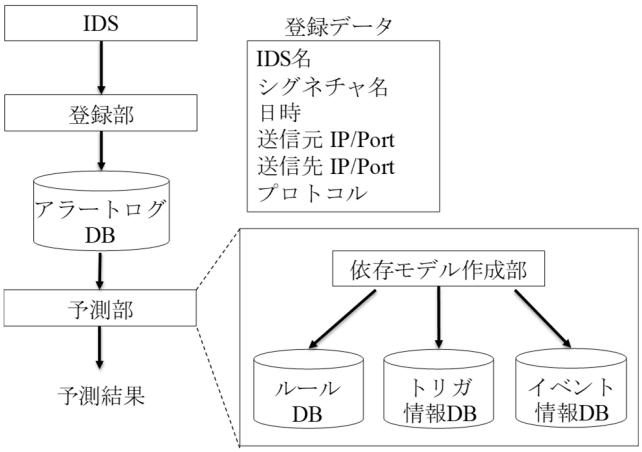

図 3　従来の被害予測システムの構成と処理の流れ

2.3 問題点

一般に，IDS を利用して不正アクセスによる被害を防ぐためには，ローカルドメインのネットワーク管理者にセキュリティに関する高度な専門知識が要求される．第一の理由は，IDS はあくまで不正アクセスやその兆候を検知するのみであり，適切な対処は管理者が行う必要があるからである．第二の理由は，一般に IDS は大量のアラートを出力するため，その中から対処が必要なアラートを管理者が見つけ出す必要があるからである．第三の理由は，アラート同士の関連性を IDS は考慮しないからである．このような問題を解決するため，イベント依存モデルが提案された．しかし，イベント依存モデルが現在でも有効な被害予測の手段たりうるかは検証が必要である．従来のイベント依存モデルの定義では，今回使用した現在の IDS のシグネチャからアラートごとに自動でトリガとイベントを定義することが困難である．現在の IDS のシグネチャ数では手動でトリガとイベントを定義することも非常に困難である．この問題点を解決するため，イベント依存モデルの定義を一部改変し，現在の IDS に対応させることで，ある不正アクセスを起点としてその後発生しうる被害を予測することができると考えられる．また，グラフデータベースを用いることで従来用いられていた RDB（Relational Database）より柔軟にアラート同士の依存関係の記述を行うことができると考えられる．

3. イベント依存モデルの定義と実装

この章では，イベント依存モデルとその実装について述べる．まず，本研究で用いる IDS とグラフデータベースについて述べる．次に，それらに合わせて改変したイベント依存モデルの定義と被害予測システムについて述べる．

3.1 IDS

イベント依存モデルを構築するにあたり，本研究では Snort [4] で提供されるルールを用いる．バージョンは Snort3.0.1 を用いた．Snort とはオープンソースのネットワーク型 IDS である． Snort で利用されるシグネチャ（検出時に用いられる攻撃パターンの特徴を抽出したもの）は世界中の開発者により日々更新されている．シグネチャは攻撃のパターンの特徴を抽出したもので，シグネチャとそのようなパターンを検知したときに Snort がどのような動作をとるのか組み合わせたものが Snort ルールである．Snort ルールは図 4 のようにルールヘッダとルールオプションによって構成され，ルールヘッダ部分にはどのようなパケットに対して適用するか，ルールオプション部分にはパケットに対してより細かい指定をすることができる．

```
alert tcp $EXTERNAL_NET any
-> $HOME_NET $HTTP_PORTS
( msg:"SERVER-APACHE  Apache Jetspeed User Manager
service unauthorized API access attempt";
flow:to_server,established; http_uri;
content:“/jetspeed/services/usermanager/users/”,fast_pattern
,nocase; metadata:policy max-detect-ips drop;
 service:http; reference:cve,2016-2171; classtype:policy-
violation; sid:46336; rev:2; )
```

図 4　Snort ルールの例

3.2 グラフデータベース

NoSQL と呼ばれるデータベースが存在する．NoSQL では RDBMS がテーブルを中心にデータ処理を行うのに対して，より多彩な形式でのデータの保存，検索を行なうことができるため，より柔軟な設計が可能である．NoSQL の一種にグラフ構造の保存に適したグラフデータベースがある．イベント依存モデルの依存関係はグラフとして記述することができるため，本研究ではデータベースとしてオープンソースのグラフデータベース Neo4j [5]を採用した．バージョンは Neo4j3.5.14 を用いた．

グラフデータベースはノード（node），属性（property），関係性（relationship）という 3 つの基本要素により構成され，ノード間の関係性を表現することができる．またデータの操作は SQL ライクな CypherQL（Cypher QueryLanguage）を通して行い，Neo4j 標準の WEB インターフェースやシェル，REST API など複数の方法で実行できる．これを利用することで，RDB よりも柔軟に依存関係を記述する．それに

よって，RDB では対応が難しかった，依存関係そのものの検索をクエリ言語で簡単に行うことができる．また，データベースのノードやリレーションを視覚的に表示でき，直感的な把握や操作ができるという特徴を持つ．

3.3 イベント依存モデル

3.3.1 トリガとイベントの定義

トリガとイベントの内容について述べる．攻撃者がある不正アクセスを実施するために，攻撃対象システムでの任意コード実行権や管理者権限，そして攻撃対象の環境といった条件が必要な場合がある．よって，不正アクセスが行われる条件であるトリガは，任意コード実行権の有無，管理者権限の有無，OS，Software，Port，そしてその他の組であると定義する．その他の項目には，フォルダ階層や言語が入る．攻撃によって起こる状況として，攻撃者の権限の上昇や情報の取得が考えられる．よって，不正アクセス実施後に起こるイベントは任意コード実行権の有無，管理者権限の有無，OS，Software，Port，そしてその他の組であると定義する．その他の項目には，フォルダ階層や言語が入る．

トリガとイベントの各項目の定義方法について述べる．任意コード実行権の有無や管理者権限の有無は Snort ルールのルールオプションの msg の項目や classtype の項目を参照し，定義する．例えば，msg に “remote code execution attempt” などの記述がある場合，イベントの任意コード実行権をオンにし，classtype に “attempted-admin” などの記述がある場合，イベントの管理者権限をオンにする．OS と Software は Snort ルールのルールオプションの reference の項目に CVE（Common Vulnerabilities and Exposures）[6] が指定されているときは，CVE の情報から OS 名や Software 名およびバージョン情報を引き出す．reference に指定されていない場合や CVE に OS や Software の情報がない場合は，Snort ルールのルールオプションの msg の項目を参照し，定義する．Port は Snort ルールのルールヘッダを参照し，定義する．その他は Snort ルールのルールオプションの content の項目や msg の項目を参照して定義する．例えば，content の内容がファイルパスの場合は，そのパスをイベントのその他に，ひとつ上位のディレクトリまでのパスをトリガのその他に格納する．そのほかにも，content の内容が言語を特定できる拡張子のときは，その言語を格納する．

3.3.2 イベント依存モデルのルール

あるアラートを A_i，そのアラートのトリガを T_i，イベントを E_i とする．トリガとイベントの各要素は，任意コード実行権の有無：*code*，管理者権限の有無：*admin*，OS：*os*，Software：*soft*，Port：*port*，その他；*other* とする．このとき，$T_i = \{ T_{i.code}, T_{i.admin}, T_{i.os}, T_{i.soft}, T_{i.port}, T_{i.other} \}$，$E_i = \{ E_{i.code}, E_{i.admin}, E_{i.os}, E_{i.soft}, E_{i.port}, E_{i.other} \}$ と表すことができる．A_i と A_j について，次の式(1)の関係を満たすとき，依存関係があると定義する．A_i が依存元，A_j が依存先である．

$$E_i \supseteq T_j \tag{7}$$

以下の式(8)～(13)の条件をすべて満たした場合，式(7)が成り立つとする．

$$E_{i.code} \supseteq T_{j.code} \tag{8}$$

$$E_{i.admin} \supseteq T_{j.admin} \tag{9}$$

$$E_{i.os} \cap T_{j.os} \neq \phi \tag{10}$$

$$E_{i.soft} \cap T_{j.soft} \neq \phi \tag{11}$$

$$E_{i.port} \cap T_{j.port} \neq \phi \tag{12}$$

$$E_{i.other} \cap T_{j.other} \neq \phi \tag{13}$$

実際のアラート入力時にこのルールを採用した場合，OS，Software と Port が複数格納されているときに意図しない依存関係が生まれることが考えられる．例えば，OS が Windows7 と指定されているアラートの依存先に OS を問わないアラートが選ばれ，その次の依存先に Windows10 と指定されているアラートが選ばれる場合がある．これを防ぐため，データベース構築時は式(8)～(13)のみを採用するが，実際のアラート入力時は式(14)～(16)も満たす場合に式(7)が成り立つとする．これまでに通ったイベントとトリガの中で一番厳しい条件を S とする．

$$S_{os} \cap E_{i.os} \cap T_{j.os} \neq \phi \tag{14}$$

$$S_{soft} \cap E_{i.soft} \cap T_{j.soft} \neq \phi \tag{15}$$

$$S_{port} \cap E_{i.port} \cap T_{j.port} \neq \phi \tag{16}$$

トリガ	イベント
• 任意コード実行権の有無 • 管理者権限の有無 • OS • Software • Port • other 　– ファイル階層 　– 言語	• 任意コード実行権の有無 • 管理者権限の有無 • OS • Software • Port • other 　– ファイル階層 　– 言語

図 5　トリガとイベントの定義

3.4 被害予測システム

図 6 に被害予測システムの構成と処理の流れを示す. IDS のアラートが予測部に入力されると，依存モデル作成部でデータベースを参照し，予測結果を出力する．データベースにはアラート情報及びアラート同士の依存関係が格納されている．アラート情報として，Snort ルールのルールオプションの msg と sid, そして reference に CVE が指定されている場合にはそのアラートの深刻度として CVSS（Common Vulnerability Scoring System）[7]の Base Score を格納する．msg はアラートメッセージを示し，sid はアラートのシグネチャの id を示す．アラートの深刻度が高い場合に，予測結果として依存先のアラートが出力される．

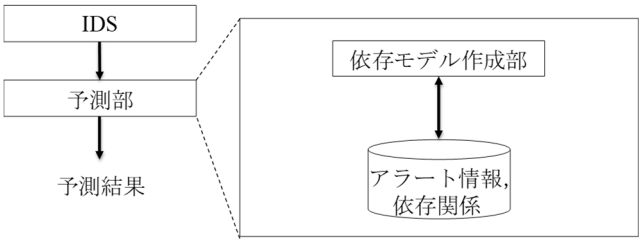

図 6　被害予測システムの構成と処理の流れ

4. 動作例

この章では，提案システムの動作例として IIS SERVER の脆弱性を利用した例と Wordpress の脆弱性を利用した例を紹介する．

4.1 IIS SERVER の例

図 7 にこの例の提案システムでの出力を示す．図中に丸で示されるものがノードであり，このノード一つ一つがアラートを表している．このノードの中にアラート情報やトリガとイベントの情報が格納されている．ノードの中の数字は Snort ルールが指定するアラートごとの番号の sid である．入力は，sid が 1029，アラートメッセージが “SERVER-IIS scripts-browse access” のアラートである．これは，WEB サーバの一つである IIS の “/scripts/” ディレクトリに対しての情報収集行為を示すアラートである．この不正アクセスはそれ自体が致命的な不正アクセスではない．その後，sid が 994 の “SERVER-IIS /scripts/iisadmin/default.htm access” などが予測できるアラートとして表示されている．その後，sid が 1026 の “SERVER-IIS perl-browse newline attempt” が成功すれば攻撃者に任意コード実行権を取得される．この時点では，被害側が適切にユーザ権限を分けていれば被害は一部に収まる．そしてその後，sid が 46935 の “OS-WINDOWS Microsoft Windows DNSAPI remote code execution attempt” の不正アクセスが成功すると OS の脆弱性を利用して攻撃者に管理者権限を獲得される恐れがある．そのため，これは危険なアラートである．管理者は依存関係のあるそれぞれのアラートのトリガ情報や Snort のルールドキュメント[8] およびそのリファレンスを見て，IIS の “/scripts/” ディレクトリを削除したり，OS のバージョンを変更したりする対処が可能になる．

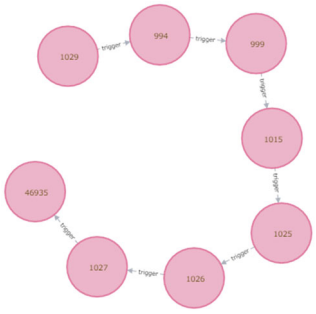

図 7　IIS SERVER の例での被害予測結果のグラフデータベースによる可視化

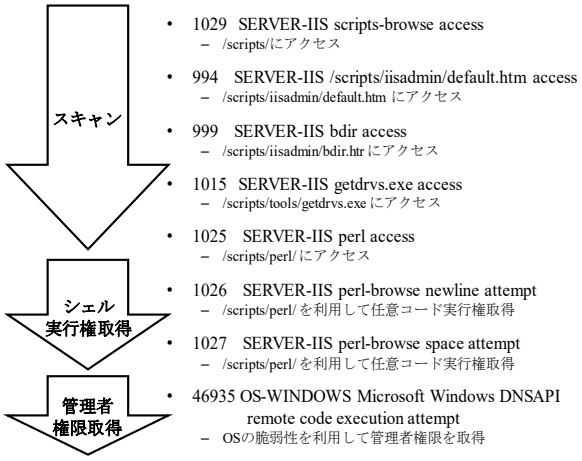

図 8　IIS SERVER の例での被害予測結果のシナリオ

4.2 Wordpress の例

図 9 にこの例の提案システムでの出力を示す．入力アラートは sid が 34475，アラートメッセージが “SERVER-WEBAPP Wordpress username enumeration attempt” のアラートである．これは Wordpress の脆弱性を用いて Wordpress のユーザ ID の一覧を得る攻撃である．この攻撃はそれ単体で管理者権限を取得されるなどの致命的な不正アクセスではない．その後，sid が 26557 で “SERVER-WEBAPP Wordpress brute-force login attempt” で示されるブルートフォース攻撃によって Wordpress の管理者権限の取得に攻撃者が成功した場合，攻撃者によって図 10 に示すようにマルウェアを設置されたり，web サイトを改ざんされたりする可能性がある．そのため，これは深刻なアラートである．管理者は依存関係のあるそれぞれのアラートのトリガ情報や Snort のルールドキュメントおよびそのリファレンスを見て，Wordpress のログインページの URL を変更したり，ブルートフォース攻撃の対策ができるプラグインを導入したりする対処が可能となる．

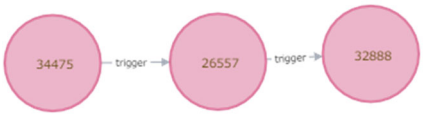

図 9　Wordpress の例での被害予測結果のグラフデータベースによる可視化

- 34475 SERVER-WEBAPP Wordpress username enumeration attempt
  - http://site.com/wordpress/?author=1 などにアクセスすることで Wordpressのユーザ IDの一覧を得る
- 26557 SERVER-WEBAPP Wordpress brute-force login attempt
  - 得たユーザIDを用いてブルートフォース攻撃を行い，Wordpress の管理者権限を得る
- 32888 INDICATOR-COMPROMISE Potential Redirect from Compromised WordPress site to Fedex - Spammed Malware Download attempt
  - マルウェアを仕込む

図 10　Wordpress の例での被害予測結果のシナリオ

5. むすび

不正アクセスへの未然の対処を目的として先にイベント依存モデルが提案された．本研究では，先に提案されたイベント依存モデルを現在の IDS に適用するために定義の改変を行い，ネットワークベース IDS の Snort のシグネチャである Snort ルールとグラフデータベースの Neo4j を用いて実装を行った. IIS SERVER の脆弱性を利用して最終的には攻撃者が管理者権限を獲得する例と Wordpress の脆弱性を用いた例を示し，提案したイベント依存モデルの定義の有効性を示した．この被害予測システムを用いて，管理者は将来実施される可能性のある不正アクセスの情報を得て，被害を最小限に抑えることができる．

謝辞　本研究は JSPS 科研費　20K11810 の助成を受けたものです.

参考文献

[1] 不正アクセス監視サービス(IDS), http://www.intellilink.co.jp/security/services/scrutiny/01.html , (参照 2020-08-17).
[2] Y.Tachibana, H.Takeuchi, H.Kurauchi and M.Morii, “Damage Analysis Support System for Illegal Access”, Proc. of 7th World Multi Conference on Systems,Cybernetics and Informatics(SCI2003),Jul.2003.
[3] 栗林利光，白石善明，森井昌克, “イベント依存モデルによる不正アクセスの被害予測”, 2004 年暗号と情報セキュリティシンポジウム(SCIS2004), 2004 年 1 月.
[4] Snort., https://www.snort.org/, (参照 2020-08-17).
[5] Neo4j, https://neo4j.com/, (参照 2020-08-17).
[6] CVE(Common Vulnerabilities and Exposures)., http://cve.mitre.org/index.html, (参照 2020-08-17).
[7] CVSS(Common Vulnerability Scoring System), https://www.first.org/cvss/, (参照 2020-08-17).
[8] Snort Rule Documents, https://www.snort.org/rule docs/, (参照 2020-08-17).