動的モデリングに基づいたリスク評価システム

杉本 暁彦 †　　磯部 義明 ‡

†(株) 日立製作所　横浜研究所　　‡(株) 日立製作所　横浜研究所

あらまし　年間約 5,000 件の脆弱性情報が NIST から公開されており，情報システムの管理者は膨大な脆弱性に対してリスクを評価し，優先度をつけて効率よく対策していく必要がある．NIST などが公開する脆弱性情報には評価指標として，脆弱性の技術的な特性に基づいてセキュリティ専門家が評価した CVSS が提示されている．しかし，システム構成に基づいた対策の要否や優先度の評価は個々のシステム管理者に委ねられており，課題があった．そこで，本研究では，収集したシステム情報からリスク評価モデルを自動生成し，システム構成に基づいたリスク評価を行うシステムについて提案する．

Risk assessment system based on dynamic modeling

Akihiro Sugimoto†　　Yoshiaki Isobe‡

†Hitachi, Ltd., Yokohama Research Laboratory　　‡Hitachi, Ltd., Yokohama Research Laboratory

Abstract　About 5,000 vulnerabilities were disclosed in 2013 by the National Institute of Standards and Technology (NIST) of USA. As soon as vulnerabilities are disclosed, cyber-attacks that exploit the vulnerabilities increase suddenly. So system engineers must prioritize the vulnerabilities to deal with efficiently.

Common Vulnerability Scoring System (CVSS) was standardized for risk assessment. But risk assessment for individual systems is entrusted to system engineers. Therefore this study suggested a risk assessment system that automatically makes modeling for risk assessment based on system configuration.

1　はじめに

近年では，年間約 5,000 件のソフトウェア脆弱性情報が NIST(National Institute of Standards and Technology)[1] から公開されている [2]．一般的に，これら脆弱性情報が公開されると，公開直後から同脆弱性情報を利用した攻撃が急増する傾向にある．そのため，情報システムを運用するシステム管理者は膨大な脆弱性に対してリスクを評価し，優先度をつけて効率よく対策していく必要がある．

ソフトウェア脆弱性を評価する指標としては，CVSS(Common Vulnerability Scoring System)[3] と呼ばれる評価指標が存在する．CVSS は，脆弱性の技術的な特性に基づく基本評価 (Base Metrics)，脆弱性を取り巻く状況に基づく現状評価 (Temporal Metrics)，個々のシステム構成に基づく環境評価 (Environmental Metrics) から構成される．一般的に，NIST などが公開した脆弱性情報には，脆弱性の技術的な特性に基づいてセキュリティ専門家が評価した CVSS の基本評価値のみが付されている．

一方で，CVSS の環境評価など，個々のシステム構成に基づくリスク評価はシステム管理者に委ねられていた．しかし，個々のシステムの

リスク評価には，同システムに対する知識とセキュリティ知識の両方が必要となり，必ずしもシステム管理者が両知識を有しているとは限らない．そのため，システム管理者が脆弱性対策を行う上で，脆弱性がもたらすリスクを正しく評価できず，効率よく対策できない現状がある．

そこで，本研究では，収集したシステム情報からリスク評価モデルを自動生成し，システム構成に基づいたリスク評価を行うシステムについて提案する．本稿では，リスク評価システムの基本概念について説明する．

2　脆弱性のリスク評価指標

2.1　CVSS

CVSS は，脆弱性の技術的な特性に基づく基本評価，脆弱性を取り巻く状況に基づく現状評価，個々のシステム構成に基づく環境評価から構成される．例えば，基本評価では，攻撃元区分（Access Vector）として，脆弱性攻撃の際に攻撃者（攻撃ノード）と攻撃対象（対象ノード）の間で必要となるリモート接続のタイプに応じて，３段階で評価されている．

CVSS の基本評価は，NIST や IPA が公開する脆弱性情報に付されているため，システム管理者にとって重要な評価指標であるが，基本評価のみに従って，脆弱性対策の優先度付けをすることは望ましくない．具体的には，図１のような例において，適切にリスクを評価できない．

1. 技術的な観点からは脆弱性 V1, 脆弱性 V2 が同程度のリスクとして評価される場合がある．しかし，攻撃者にとって，ネットワーク経由で直接アクセス可能な脆弱性 V1 を有する機器への攻撃の方が遥かに容易であり，システム構成の観点からは脆弱性 V1 のリスクの方が高い．

2. (1) 同様，脆弱性 V2, 脆弱性 V3 が同程度のリスクとして評価される場合がある．しかし，攻撃者にとって，脆弱性 V1 を攻略することで踏み台とできる機器が存在する分，脆弱性 V3 へ攻撃する方が容易である．このように脆弱性の分布状況観点からは脆弱性 V3 のリスクの方が高い．

3. 技術的な観点からは脆弱性 V1 のリスクの方が, 脆弱性 V2 のリスクより高いと評価される場合がある．実際に，同脆弱性を攻略するだけであれば，脆弱性 V1 に対する攻撃の方が容易と考えられる．しかし，資産や他脆弱性の分布状況により，同脆弱性が攻略されることで，受ける影響が大きい場合，脆弱性 V2 のリスクを高く見積もる方が良い場合がある．

4. CVSS の基本評価のような技術的な観点にネットワーク階層構造の観点を加えて，リスク評価する方法も考えられる．しかし，攻撃者が脆弱性 V1 を攻撃するまでのパスが複数経路あるようなケースでは，適切にリスクを評価できない．

2.2　アタックグラフ

図１のようなリスクを分析する手法として，アタックグラフ (Attack Graphs)[4] と呼ばれる手法が存在する．アタックグラフとは，各機器における脆弱性や情報資産の保有状況，機器間のネットワーク接続性，アプリケーションサービスの稼働状況などから，脅威と脆弱性の関係をグラフモデル化する手法であり，アタックグラフによって脅威や脆弱性の関係を可視化されることで，適切なリスクの評価が可能になる．

アタックグラフは，当初セキュリティ専門家が机上においてセキュリティ分析する手段として，様々なモデルが検討され，近年では，要素技術として機械的にアタックグラフを生成する方式も検討されてきた [5][6]．しかし，文献 [5][6] では，システム構成情報の取得から，公開されたセキュリティナレッジの収集，動的なモデルの構築，リスクの評価までを自動化することはできておらず，情報システムを管理するシステム管理者に対するセキュリティ運用支援システムとして，実用には至っていなかった．

以上より，本研究では，システム構成情報の取得から，公開されたセキュリティナレッジの収集，動的なモデルの構築，リスクの評価までを自動化することを目的とする．

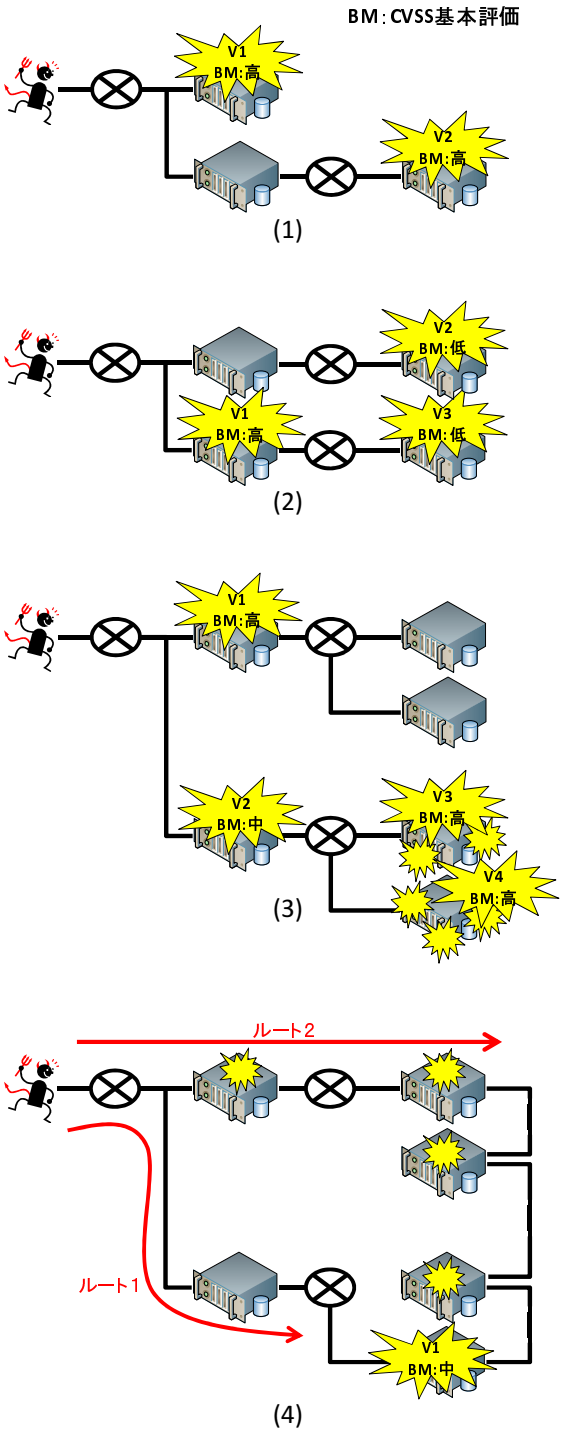

図 1: CVSS の基本評価のみでは評価しきれないリスク例

3 提案システムにおけるリスク評価モデル

本研究では，ペトリネット (Petri Net)[7] と呼ばれるグラフモデルを用いて，脅威と脆弱性の関係を構造化し，リスク評価するシステムを提案する．本章では，提案するリスク評価モデルについて説明する．

3.1 動的モデリングの提案方式

ペトリネットとは，図 2 に示すように，物事の状態を“プレース”，発生する事象を“トランジション”，状態と事象の接続関係を“アーク”，事象発生した場合にアークで接続された状態に遷移する確率を“発火確率”と定義し，システムをモデル化したグラフモデルである．

ペトリネットは，一般的な有向グラフと比べ，複数の事象が並列的に生起した場合を前提条件として，状態遷移が発生するようなシステムのモデル化を可能とする．

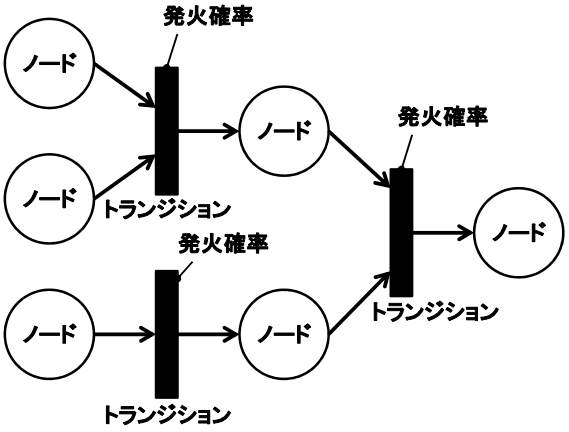

図 2: ペトリネットの例

提案方式では，上記ペトリネットを用いて，モデル化する上で，以下を定義する．

- 各機器において Exploit コードが実行できる状態，各機器が保有する認証情報が盗難された状態など，各機器が悪意のある攻撃者に侵害された状態をペトリネットにおける”プレース”として定義する．

- 特定の機器から特定の機器への攻撃手段をペトリネットにおける”トランジション”として定義する．攻撃手段としては，脆弱性をついた攻撃や，別機器から盗難した認証情報によるリモート操作，ハードニング不足によるセキュリティ設定の穴をついた攻撃などが考えられる．
- ネットワークトポロジに基づき，機器間においてメッセージの到達性があり，上記攻撃手段の前提条件に合致する”プレース”から”トランジション”へ”アーク”を接続するとする．例えば，上記攻撃手段が Exploit コードの実行を必要とする場合，Exploit コードの実行を表す”プレース” から同攻撃手段を表す”トランジション”に”アーク”が接続される．
- 上記攻撃手段により，発生する状態を表す”プレース”へ同攻撃手段を表す”トランジション”から”アーク”を接続するとする．例えば，脆弱性攻撃により Exploit コードの実行が可能となる場合，同脆弱性攻撃を表す”トランジション”から Exploit コードの実行を表す”プレース”へ”アーク”を接続する．

提案方式では，上記の定義により，機器間の関係性を図 3 のようなグラフモデルと構築する．

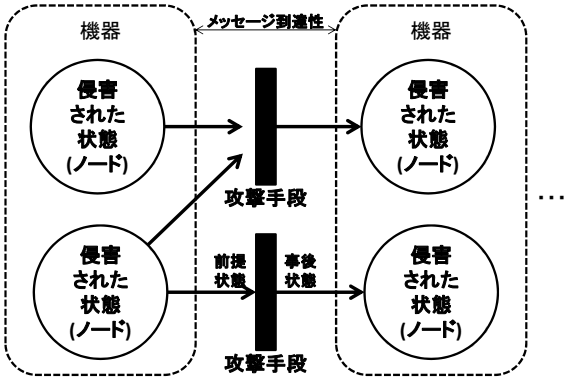

図 3: ペトリネットの例

3.2 提案方式の機能要件

上記グラフモデルを動的にモデリングするため，提案方式は，表 1 の 5 つの機能が必要と考える．

本研究では，まず，攻撃手段として脆弱性攻撃に限定し，検証を進めることとした．現在機能 F1～F3 まで，実装と検証が進んでいる．

表 1: 動的モデリングに基づいたリスク評価システムの機能要件

項番	機能名 機能概要
F1	脆弱性情報の収集・管理機能 インターネット経由でセキュリティナレッジ公開機関からソフトウェア脆弱性情報を収集する機能.
F2	システム情報の収集・管理機能 管理システムから機器情報やソフトウェアスタック情報などのシステム構成情報を収集する機能.
F3	機器と脆弱性の対応付け機能 機器情報と脆弱性情報をセマンティックに対応付ける機能.
F4	システムリスクのモデル化機能 ペトリネットにより脅威と脆弱性の関係性をグラフモデル化する機能.
F5	システムリスク値の計算機能 上記，グラフモデルからリスク値を計算する機能.

4 提案方式を用いたシステムの実現方法

本章では，表 1 の各機能の実現方法について説明する．提案システムの全体構成については，図 4 に示す．

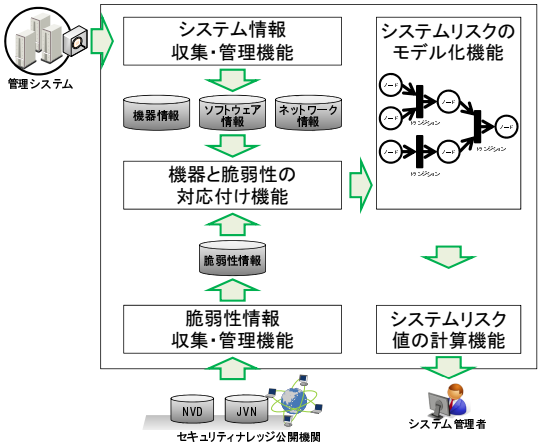

図 4: 全体構成

4.1 脆弱性情報の収集・管理機能

日本国内のベンダが開発したソフトウェアに関する脆弱性情報は，2004 年に情報処理推進機構（IPA）が策定した情報セキュリティ早期警戒パートナーシップガイドライン [8] により定められた所定の手続き・調査の後，必要性に応じて JVN(Japan Vulnerability Notes)[9] において登録・公開される．米国ベンダが開発したソフトウェアに関する脆弱性情報も，同様の手続き・調査の後，NIST が管理する NVD(National Vulnerability Database) において登録・公開される．各国の脆弱性情報は互いにシェアしており，他国の公的レポジトリの脆弱性情報についても，必要に応じて登録される．

本機能では，インターネット経由で上記レポジトリを取得することにより例えば，Apache Struts2 の脆弱性であれば，図 5 のような構造化された脆弱性情報を取得することが可能である．

```
CVE-2013-2251
├ Original release date : 07/19/2013
├ Last revised : 05/05/2014
├ Source : US-CERT/NIST
├ Overview : Apache Struts 2.0.0 through 2.3.15 allows ...
├ Impact
│ ├ CVSS Severity (version 2.0)
│ │ ├ CVSS v2 Base Score : 9.3 (HIGH) (AV:N/AC:M/Au:N/C:C/...
│ │ ├ Impact Subscore : 10.0
│ │ └ Exploitability Subscore : 8.6
│ └ CVSS Version 2 Metrics
│   ├ Access Vector : Network exploitable
│   ├ Access Complexity : Medium
│   ├ Authentication : Not required to exploit
│   └ Impact Type : Allows unauthorized disclosure of information; ...
├ References to Advisories, Solutions, and Tools : By selecting these ...
│ ├ Entrys
│ │ ├ External Source : MLIST
│ │ ├ Name : [oss-security] 20140114 Re: CVE Request: Apache Arch...
│ │ └ Hyperlink : http://seclists.org/oss-sec/2014/q1/89
│ └ ...
├ Vulnerable software and versions
│ └ Configuration 1
│   ├ cpe:/a:apache:struts:2.3.1.2
│   ├ cpe:/a:apache:struts:2.2.3.1
│   └ ...
└ Technical Details
  ├ Vulnerability Type : Input Validation (CWE-20)
  └ CVE Standard Vulnerability Entry : http://cve.mitre.org/ ...
```

図 5: Apache Struts2 の脆弱性の例

4.2 システム情報の収集・管理機能

脅威と脆弱性の関係をモデリングするためには，システム情報として，少なくとも機器情報，ソフトウェアスタック情報，ネットワーク情報を収集する必要がある．各情報は以下の方法によって収集する．

1. 機器情報

   機器情報に関しては，機器にエージェントを導入することにより，OS の標準機能を利用することで様々な情報が取得可能である．例えば，弊社では，一般的なシステム管理に必要な機器情報を OS の標準機能を利用して，取得する方法と取得するツール (IT Report Utility)[10] を一般公開している．本システムでは，IT Report Utility に基づいて機器情報を取得する．

2. ソフトウェア情報

   脆弱性があるソフトウェアを検査する標準的な仕様として，MITRE 社が策定した OVAL(Open Vulnerability and Assessment Language)[11] と呼ばれるセキュリティ検査仕様が存在する．OVAL の仕様では，インターネット上の OVAL レポジトリにおいて個々の脆弱性に関連するソフトウェアを検索するための OVAL クエリが公開されており，同定義を OVAL インタプリタと呼ばれる機器上で稼働するエージェントに読み込ませることで，ソフトウェアの有無の確認と脆弱性との対応付けが可能となる．

   しかし，OVAL のレポジトリでは，全てのソフトウェア脆弱性に対する OVAL クエリが公開されているわけではないため，提案システムでは，OVAL 情報に加え，Windows のレジストリ情報や Linux のパッケージ管理情報を参照することで，ソフトウェア情報を取得している．

3. ネットワーク情報

   ネットワークトポロジに基づき，機器間においてメッセージの到達性を判定するため，ネットワーク情報も必要となる．ネットワーク情報の取得方法としては，IETF により標準的なプロトコルとして，SNMP(Simple Network Management Protocol)[12] が策定されている．SNMP では，MIB と呼ばれるオブジェクトの階層構造を取るように，個々に情報に対して，Identifier が割付けら

れており，Identifier を指定することで情報の参照が可能になる．提案システムにより取得する情報を表 2 に示す．

表 2: 取得するネットワーク情報

項番	ネットワーク情報	取得コマンド (Alaxala の例)
	SNMP (iso.org.dod.internet.mgmt.mib-2)	
1	ポート情報	show port
	.ip.ipForwarding	
2	ネットワークインタフェース情報	show interfaces
	.interfaces.ifTable.ifEntry	
3	IP アドレス情報	show ip interface
	.ip.ipAddrTable.ipAddrEntry	
4	arp キャッシュ情報	show ip arp
	.ipNetToMediaTable.ipNetToMediaEntry	
5	MAC テーブル情報	show mac-address-table
	.dot1dBridge.dot1dTpFdbEntry	

4.3 機器と脆弱性の対応付け機能

前記，OVAL 仕様に基づき，取得したソフトウェア情報は脆弱性との対応付けられているため，機器と脆弱性の対応付けは容易に可能である．しかし，レジストリやパッケージ管理情報から取得したソフトウェア情報は，別途脆弱性と対応付けを行う必要がある．

図 5 の例の Vulnerable software and versions にあるように，脆弱性情報には，CPE(Common Platform Enumeration)[13] と呼ばれるソフトウェアやハードウェアの識別子が含まれている．CPE はベンダ名や製品名，バージョンなどを構成要素として持つ識別子であり，CPE に対応する正式な製品名を記述した辞書 (CPE 辞書) も公開されている．

提案システムでは，OVAL に基づいた機器と脆弱性の対応付けに加え，レジストリやパッケージ管理情報から取得したソフトウェアの名称やバージョンを CPE 辞書や文字列マッチングを利用して，CPE に変換することで，機器と脆弱性の対応付けを実現する．

4.4 システムリスクのモデル化機能

前記情報を基に，ペトリネットを用いてグラフモデルを構築する．現在，実装を行っている提案システムでは，図 3 の攻撃手段として，脆弱性攻撃に限定しているため，モデル構築手順の概要は以下となる．

1. ネットワーク情報から機器間でメッセージ（パケット）が到達するか判定する．
2. 脆弱性攻撃に限定しているため，各機器に対応付けられた脆弱性を列挙することで，各機器への攻撃手段とする．今後は，認証情報を使ったリモート操作やセキュリティ設定の穴を突いた攻撃なども攻撃手段としていく必要があると考えている．
3. ”Exploit コードを実行できる”や”機器内の資産を参照できる”，”機器の可用性を阻害できる”などを図 3 の機器の侵害された状態（ノード）とし，他機器への攻撃へと繋げることが可能な侵害状態を表すノードから他機器への攻撃を表すトランジションへアークを接続する．この時，機器間のメッセージ到達性を考慮する．

5 脆弱性リスク評価システムの評価

5.1 進捗報告

本章では，現在の前記提案システムにおける進捗状況について報告する．現状の提案システムでは，機能 F1〜F4 まで実装しているため，管理対象が保有する脆弱性を自動的にリスト化し，優先度を付けて，システム管理者に提示することが可能になっている（図 6）．リスク値を計算する機能 F5 については，更なる検討と検証が必要であるため，図 6 の例では，グラフモデルに基づき，CVSS 値を補正することで，優先度付けを行っている．今後は，機能の追加と共に，可視化なども必要と考えている．

5.2 考察

提案システムを開発していく上で，以下 4 つの知見を得た．下記については，今後も継続して，評価していく予定である．

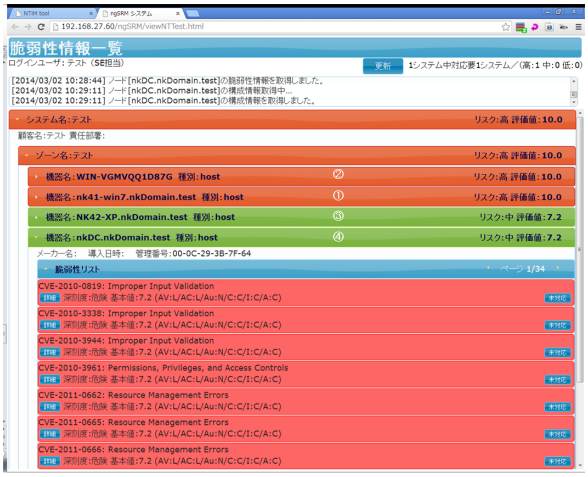

図 6: リスク評価システムの画面

1. 第一章でも記載したように，脆弱性情報が公開されると，公開直後から同脆弱性情報を利用した攻撃が急増する傾向にある．例えば，昨年度深刻な脅威として，問題となった Apache Struts2 の例では，公開直後の 7 月 17 日より攻撃が急増していた．しかし，公的レポジトリの登録には，時間を要す場合があり，Apache Struts2 の例では，7 月 16 日に Apache から脆弱性情報が公開された後，NVD に登録されたのは 7 月 20 日，JVN に登録されたのは 7 月 23 日であった．結果として，セキュリティ・オートメーション化する上で，公的なナレッジのみを参照するだけでは，セキュリティとしてのアジリティに欠ける場合があり，より根源的な情報ソースを追う仕組みを検討している．

2. 提案システムでは，標準化されたプロトコルで取得できる表 2 のネットワーク情報から機器間のメッセージ可達性を評価した．更に，ネットワーク内で通信可能なプロトコルや IP 単位で通信制限状況などの情報を収集することで，実際に機器間で攻撃のやりとりが可能かまでは評価できる見込みである．

3. 現在，脆弱性情報に関連するソフトウェアは CVE により ID 付けられているが，CVE はシステムからシステム情報として直接取得できる情報ではなく，また，ソフトウェア名という曖昧性のある情報で構成されているため，機器と脆弱性の対応付ける上で，ミスマッチが発生していた．提案システムでは，文字列のマッチング技術なども加えて，精度の向上を図っている．また，ISO/IEC 19770 において規格が策定したソフトウェア ID タグは，新しいソフトウェアの Identifier であり，システム情報として取得できることが期待されている．

4. 現在，実装を行っている提案システムでは，攻撃手段として，脆弱性攻撃に限定しているが，異なる攻撃手段もモデルに組み込む必要がある．特に，厳しくハードニングされているフロントシステムこそ脆弱性攻撃を受ける機会が多いが，ハードニングが甘くなりがちなバックエンドのシステムでは，セキュリティ設定の穴を突いた攻撃が増加すると考えられる．これら攻撃をモデルに組み込むためには，セキュリティ設定の取得方法なども検討していく必要がある．

6 おわりに

本論文では，情報システムの管理者が公開された膨大なソフトウェア脆弱性の対策を支援するため，システム情報や脆弱性情報を収集し，自動的に脆弱性のリスク評価モデルを構築して，リスクを評価することで，脆弱性に優先度付けし，セキュリティ運用を支援するシステムを提案した．

本論文では，特に公開されている脆弱性情報に付加されている CVSS の基本評価のみに基づいて，リスク評価した場合の問題点について指摘し，システム構成情報に基づいて，脅威と脆弱性の関係性を表すモデルを構築する方法を提案した．

本研究では，現在，上記方式に基づき，情報の取得からリスク評価までをオートメーション化するリスク評価システムのプロトタイプ開発に取り組んでいる．今後も継続して上記開発に取り組んでいくつもりである．

参考文献

[1] NIST(National Institute of Standards and Technology).
http://www.nist.gov/

[2] NIST, "National Vulnerability Database (NVD) CVE Statistics".
http://web.nvd.nist.gov/view/vuln/statistics-results?cves=on&pub_date_start_month=0&pub_date_start_year=2000&pub_date_end_month=11&pub_date_end_year=2014

[3] Mike Schiffman, Gerhard Eschelbeck, David Ahmad, Andrew Wright, Sasha Romanosky, "CVSS: A Common Vulnerability Scoring System", National Infrastructure Advisory Council (NIAC), 2004.

[4] Ou, Xinming, Singhal, Anoop, "Quantitative Security Risk Assessment of Enterprise Networks", Springer, 2011.

[5] Xinming Ou, Sudhakar Govindavajhala, and Andrew W. Appel, "MulVAL: A logic-based network security analyzer", 14th USENIX Security Symposium, Baltimore, Maryland, U.S.A., August 2005.

[6] Oleg Sheyner, Jeannette Wing, "Tools for Generating and Analyzing Attack Graphs", Lecture Notes in Computer Science Volume 3188, 2004, pp 344-371.

[7] Meseguer, J. Montanari, el al. "information and computation 88", 105-155, 1990.

[8] 独立行政法人情報処理推進機構, "情報セキュリティ早期警戒パートナーシップガイドライン（第 7 版）", 2011
http://www.ipa.go.jp/files/000002991.pdf

[9] JVN(Japan Vulnerability Notes).
https://jvn.jp/

[10] (株) 日立製作所, IT Report Utility.
http://www.hitachi.co.jp/Prod/comp/soft1/sjst/windows/0201/044451-K1.pdf

[11] OVAL(Open Vulnerability and Assessment Language).
https://nvd.nist.gov/scap/docs/conference%20presentations/workshops/OVAL%20Tutorial%201%20-%20Overview.pdf

[12] IETF, " A Simple Network Management Protocol (SNMP) ", RFC1157, https://www.ietf.org/rfc/rfc1157.txt

[13] CPE(Common Platform Enumeration, http://cpe.mitre.org/

[14] LAC, "Apache Struts2 の脆弱性 (S2-016) を悪用した攻撃の急増について", 2013, http://www.lac.co.jp/security/alert/2013/07/18_alert_01.html