SCIS 2009 The 2009 Symposium on Cryptography and Information Security
Otsu, Japan, Jan. 20-23, 2009
The Institute of Electronics, Information and Communication Engineers

脆弱性情報提供 Web API “AVIP” の開発
The Development of the Vulnerability Information Provision Web API “AVIP”

原田 敏樹 * 金岡 晃 † 加藤 雅彦 ‡ 岡本 栄司 †
Toshiki Harada Akira Kanaoka Masahiko Kato Eiji Okamoto

あらまし 現在，インターネットを通じて公開されている脆弱性情報は，ブラウザを介しての人手による確認を前提としており，その情報をデータとして機械的に読み込むことができない．すなわち，現状では脆弱性の影響判断を，システム管理者等が人手で行わなければならず，脆弱性情報が日々増加する中で，このことは大きな負担となっている．本研究では，自動的かつ定量的な脆弱性の影響判断を可能にすることを目的として，脆弱性情報の検索結果をデータとして読み込み可能な XML 形式で返す Web API “AVIP” の開発を行った．さらに，AVIP で得られた情報と CVSS を連携することによって，ネットワークシステムにおける脆弱性影響を自動的に数値化するツールを試作した．

キーワード ネットワークシステム，脆弱性情報，Web API，脆弱性影響，自動化

1 はじめに

インターネットを通じて 1 つのサービスを提供する場合，さまざまなアプリケーションが個別にアクセス制御を行い，全体として 1 つのサービスを提供するネットワーク化したシステム（ネットワークシステム，以下 NS）となっていることが一般的である．近年インターネット利用者の増加やセキュリティに対する要求の高度化により NS の規模や複雑さが肥大化する一方，NS を管理運用する現場では技術者の経験を主とするセキュリティ設計，セキュリティ運用手法が利用されていることが多い．そうした経験者依存の NS 管理の現場では，システム構成要素をテキストや表形式にしてファイルやデータベースで管理し，ネットワーク構造を描画ツール等を使用して表記するという旧来のデータ記述手法が未だ主流となっている．そのため，システムのつながりを瞬時に判断するのが難しく，脆弱性の影響判断を手作業で一つ一つ確認するといったことが起こっているのが実情である．そのような状況で，米国では情報セキュリティに関わる技術面での自動化と標準化を目指し，Security Content Automation Protocol（SCAP）[1] と呼ばれる技術仕様の策定が行われている．

しかし，SCAP を利用した National Vulnerability Database（NVD）[2] で提供されている脆弱性情報は主としてブラウザでの利用を前提としており，脆弱性の影響判断の自動化には至っていない．XML 形式による脆弱性情報の公開も行われているが，これは年別もしくは最新情報をまとめたものであり，その中から手作業によって欲しい情報を取捨選択する必要がある．

また，SCAP の 1 つに Common Vulnerability Scoreing System（CVSS）[3] と呼ばれる脆弱性影響度の定量化手法があり，これによりベンダ，管理者，ユーザ等の間で，脆弱性の深刻度を同一の基準の下で定量的に比較することが可能となる．CVSS には製品の利用環境に依存する脆弱性の深刻度を表す環境評価基準という値が存在するが，現状の NS 管理においてこの値の算出はネットワーク管理者が人的に行うことになり，再現性のある評価や運用管理の自動化が困難なものとなっている．

そこで本研究では，まず既存の脆弱性影響の定量化手法に関する調査を行った．そしてその手法の自動化に必要な情報を取得するため，既存の脆弱性情報提供サービスについて調査を行い，そこで見つかった問題点を解決する Web API “AVIP（Automatic Vulnerability Information Provider）”[4] を開発した．さらに，AVIP で得られた情報を用いて，NS の脆弱性影響を自動的に検査・評価するツールを試作した．

* 筑波大学, 茨城県つくば市天王台 1-1-1, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, Japan
† 筑波大学大学院 システム情報工学研究科, 茨城県つくば市天王台 1-1-1, Graduate School of Systems and Information Engineering, University of Tsukuba , 1-1-1 Tennodai, Tsukuba, Ibaraki, Japan
‡ 株式会社アイアイジェイテクノロジー, 東京都千代田区神田神保町 1-105 神保町三井ビルディング, IIJ Technology, Inc., 1-105 Kanda jinbo-cho, Chiyoda-ku, Tokyo, Japan

第 2 節では，本研究の関連研究について紹介する．また第 3 節で，既存の脆弱性情報提供サービスの調査を行い，現在の提供形態の問題点を挙げる．第 4 節において，その問題点を解決するため開発した Web API “AVIP” について述べ，つづく第 5 節では，AVIP で取得した情報を利用して NS の脆弱性影響評価を行うツールを試作したので，それを紹介する．最後に第 6 節でまとめる．

2 関連研究

2.1 NSQ モデル

一般的に NS は複数の機能を持つ機器群から構成されるが，これまでの研究によるコンピュータネットワークの解析は，ルータや Autonomous System（AS）などの同一機能をもつもの同士のつながりに関するネットワーク解析であり，複数機能を有する NS の解析ではない．NS の脆弱性判断を行うためには，アクセス制御情報，システム構成情報，脆弱性情報といった複数の情報をまとめて扱う必要があり，同一機能を前提としたネットワークモデルを直接適用することは困難である．そこで，金岡らは複数機能を有する NS の新たな表現モデル（以下 NSQ モデル）[5][6][7] の提案を行った．NSQ モデルは NS を通信層（レイヤ）毎に分解し，それぞれのレイヤに存在する，ネットワーク機能として認識可能な ID をノードとして表現する（図 1）．また，複数のレイヤにまたがってノードを接続したものをモジュールと定義し，一般的な機器はモジュールで表現する．さらに，ノードが同一レイヤ内でどのノードからアクセスされるか，上位レイヤのノードがどの下位レイヤのノードに依存して存在しているかをノード間の接続として表すことで，NS 内の識別可能なノードのアクセス制御と依存性が表現可能となっている（図 2）．

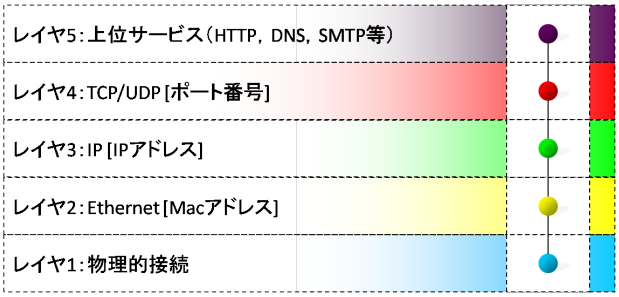

図 1: NSQ モデルレイヤ定義

2.2 脆弱性定量化

脆弱性による影響を数値化する代表的な手法として CVSS があげられる．CVSS は基本評価基準，現状評価基準，環境評価基準の 3 つの値で構成されている．基本評価基準は脆弱性そのものの特性として攻撃経路情報，認証の必要性，攻撃の複雑さ，機密性，可用性，完全性に対する影響を各項目毎に評価したもの（CVSS Vector）により求められる値である．現状評価基準は脆弱性の対応状況による深刻度を，環境評価基準は実際に攻撃を受けるシステムへの影響度を表している．これらの基準のうち基本評価基準は評価機関により定められるが，環境評価基準はユーザの環境に応じてユーザが判断する必要があるため，結果として値が定量的に定まらない一因となっている．

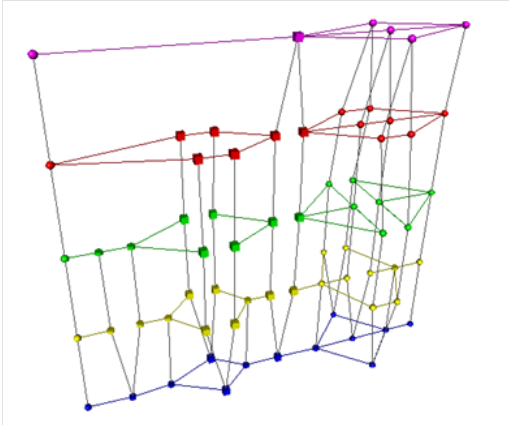

図 2: NSQ モデルを用いた NS 図

2.3 NSQ モデルへの CVSS の適用

加藤らの研究 [8] において，NSQ モデルにおけるノード属性として構成情報を含ませ，NSQ モデルの定義が拡張された．具体的には，レイヤ 3 に OS 名とバージョン，レイヤ 4 にアプリケーション名とバージョンの情報が含まれている．また，その拡張された NSQ モデルに CVSS を適用することで，脆弱性の影響範囲の特定と影響度の定量化が行われた．なお，ここでの影響範囲特定には CVSS Vector が用いられている．

3 既存の脆弱性情報提供サービス

コンピュータシステムに発見される脆弱性の数は毎月 300 件以上にもなり，これらすべての情報を追跡する労力はシステム担当者やネットワーク管理者にとって大きな負担となっていた．こうした負担をなくすため，現在ではベンダに依存しないさまざまな脆弱性情報を取り扱うデータベースや Web API が公開されている．そこで我々は，NS における脆弱性の影響判断に必要なソフトウェア情報や CVSS Vector を，機械的に読み取り可能

なデータ形式で得ることが既存のデータベースや Web API を用いることで実現可能かどうかを確認するため，これらの調査を行った．

3.1 脆弱性情報データベースの調査

規模の大きなデータベースには，NVD や Japan Vulnerability Notes（JVN）[9]，JVN iPedia[10]，The Open Source Vulnerability Database（OSVDB）[11] がある．それぞれで得られる情報の内容と形式等について調査した．

- NVD
  米国国立標準技術研究所（NIST）が公開している．米国政府が公開した資料や企業・業界団体が編集した脆弱性情報をデータベース化することによって作成された．現在では 30,000 件以上の脆弱性情報をデータベース化しており，脆弱性が発見されたソフトウェア名とバージョン番号，ソフトウェアを開発したベンダ名，CVSS 基本評価基準や CVSS Vector などによって検索できる．検索結果は HTML によるリンク一覧として返される．また，最新情報の RSS 配信や，年別もしくは最新情報の XML によるデータフィードも行っており，毎日更新される．

- JVN
  情報処理推進機構（IPA）と JPCERT コーディネーションセンター（JPCERT/CC）が共同で公開している．CERT/CC や，Centre for the Protection of National Infrastructure（CPNI）の他，一般に公開された脆弱性情報を独自に収集し，製品開発者との調整を行って情報を掲載している．対応状況や対策情報をいち早く周知することを目的としており，検索などは行えない．

- JVN iPedia
  IPA が公開している．国内のソフトウェア製品開発者が公開した脆弱性対策情報，JVN で公表した脆弱性対策情報，NVD が公開した脆弱性対策情報，以上の 3 つの中から国内で利用されている製品に関する情報が登録されている．現在では 5,000 件以上の脆弱性情報が登録され，ベンダ名や製品名，CVSS 基本評価基準などによって検索できる．検索結果は HTML によるリンク一覧として返される．年別もしくは最新情報の RSS によるデータフィードも行っている．

- OSVDB
  民間のセキュリティコミュニティが運営している．脆弱性情報件数は 40,000 件を超え，他組織や Nessus 等の脆弱性スキャンソフトウェアが所有する脆弱性情報と相互参照が可能である．ベンダ名や製品名の他，50 種近くの脆弱性分類によって検索できる．また，XML や MySQL 形式を用いたデータベースのエクスポートも提供しているが，XML のエクスポートは，2008 年 9 月末を最後に更新を行っていない．

以上 4 つのデータベースの中では，OSVDB が最も情報量が多いが，CVSS など，脆弱性影響の定量評価指標が得られない．次に情報量の多い NVD では，XML によるデータフィードも行っており，このデータには唯一 CVSS Vector が含まれる．そのため，脆弱性影響判断の効率化，自動化という点においては最も進んでいると言える．上記の 4 つの脆弱性情報データベースは全て検索結果は HTML によるリンク一覧で返され，XML 等のデータフィードを持つものは年別もしくは最新情報のみでまとまったものしか得られない．HTML での提供はつまり，ネットワーク管理者が人手で確認，脆弱性判断を行うことを意味する．データフィードでの提供においても，一まとまりになっていることから，欲しい情報の判断が困難である．

3.2 脆弱性情報提供 Web API の調査

検索結果を XML 形式で返すサービスに，MyJVN Web API[12] と OSVDB API がある．それぞれの XML 取得方法や，XML の内容について調査した．

- MyJVN Web API
  JVN iPedia と同様，IPA が公開している．情報源は JVN iPedia である．製品名の記述に Common Platform Enumeration（CPE）[13] の適用を試行している．ただし，本来 CPE は，種別，ベンダ名，製品名，バージョン，アップデート，エディション，言語の 7 つの情報から構成され，提供される製品情報 XML にはこれらの情報が全て含まれているが，検索のキーとしては種別，ベンダ名，製品名，言語のみの指定となり，バージョン，アップデート，エディションの情報は検索に用いることができない．レスポンスとなる XML は RSS1.0 をベースにしており，既存の RSS リーダをアクセスツールの一つとして利用できる．情報源が JVN iPedia であることから，CVSS 基本評価基準が情報に含まれる．

- OSVDB API
  情報源は OSVDB である．脆弱性情報の要求には，

OSVDB 内で決められているベンダ ID が必要であり，製品 ID，バージョン ID での絞り込みが可能である．それらの ID はあらかじめそれぞれの名称で検索を行い取得しておくことが必要となる．また，キーワード検索も可能である．脆弱性情報の内容は，他組織における脆弱性 ID などのリファレンスや，脆弱性分類 ID などが主となっており，脆弱性定量評価を示す値は含まれていない．

以上，2 つの Web API について調査を行った．MyJVN Web API においては，バージョン情報をキーとしての検索が不可能であることと，情報源が JVN iPedia のみである事により CVSS Vector が得られない点が問題点として挙げられる．OSVDB API は各種 ID を用いた一意的な検索が可能だが，ここで提供される情報そのものにはやはり CVSS が存在せず，NSQ モデルに基づいた脆弱性影響評価のためのデータとしては不適当である．

4 Web API “AVIP”

前節の調査結果から，提供情報に CVSS Vector が含まれているのは NVD のみであることが分かった．すなわち，NVD の脆弱性情報の検索結果を XML で得ることで，脆弱性の影響判断の自動化，定量化が可能となる．そこで本節では，その要項を満たすために本研究で開発した脆弱性情報提供 Web API “AVIP”について述べる．

4.1 動作概要

図 3 に示すように，まず NVD から XML ファイルをあらかじめダウンロードし，利用するデータを取り出して自らのデータベースに格納しておく．ユーザから脆弱性情報のリクエストがあった際には自データベース内で検索し，1 つのリクエストに適合するレコード群を 1 つの XML ファイルで返す．

4.2 XML 内容

AVIP は NVD の XML フィードを情報源としているため，提供される脆弱性情報の内容は NVD に準拠する．ただし，本研究では，脆弱性の説明や対策方法，リファレンスなど，脆弱性の影響判断の自動化に不適と判断したテキストベースの情報は省略してある．返される XML の構造と，各タグの説明を表 1 に示す．なお，各要素の詳細については，NVD XML Feed Documentation[14] を参照されたい．

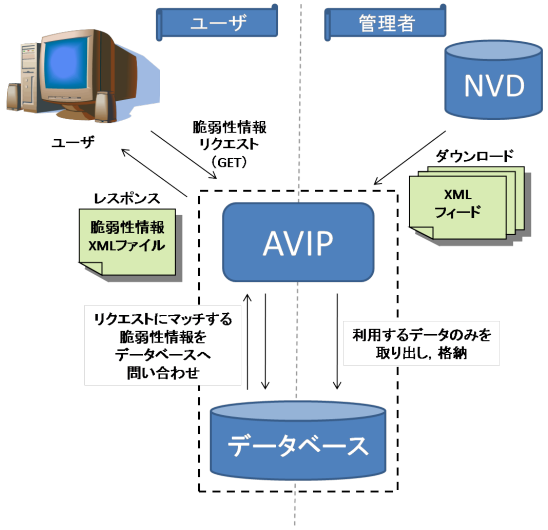

図 3: AVIP の動作概要

4.3 リクエスト

リクエストを送る際の基本 URL は以下の通りである．

```
http://avip.cipher.risk.tsukuba.ac.jp/vulns?
```

この後に続けて，表 2 に示すリクエストパラメタを指定することで，リクエストが完成する．AVIP では，CVE 番号，公開/更新年月日，製品名，ベンダ名の 4 つの内，1 つをキーに指定して検索する．リクエストパラメタを指定して GET にて HTTP リクエストを発行すると，XML 形式のレスポンスが返ってくる．

例として，以下のようなリクエストを発行した場合，図 4 に示すような XML が取得できる

```
http://avip.cipher.risk.tsukuba.ac.jp/vulns?
match=full&product=avip&version=1
```

5 脆弱性影響の検査・評価ツール

AVIP で取得した XML の利用例として，加藤らの研究で述べられている CVSS の適用法を用いて，NS における脆弱性影響を評価するツール（Vulnerability-effect Assessment ツール，以下 VA ツール）を試作した．本節では，その VA ツールについて述べる．

5.1 動作概要

図 5 に示すように，VA ツールへの入力には，NS 構成を表現した XML ファイルを用いる．この XML ファイルは，金岡らの提案した NSQ モデルを表現したものとなっており，NS 内で利用している OS やアプリケー

表 1: AVIP の XML 内容

要素名	直下子要素	説明
vuln_list	AVIP_published, entry	ルート要素
AVIP_published	なし	XML 出力時刻
entry	name, published, modified, severity, cvss, loss_types, vuln_types, range, vuln_softs	脆弱性情報エントリ (複数)
name	なし	CVE 番号
published	なし	公開年月日
modified	なし	更新年月日
severity	なし	深刻度
cvss	cvss_version, cvss_vector, cvss_base_score, cvss_impact_subscore, cvss_exploit_subscore,	CVSS に関する情報
loss_types	avail, conf, int sec_prot	脆弱性による損失の種類
vuln_types	access, input, design, exception, env, config race, other	脆弱性の種類
range	local, local_network network, usr_ init	攻撃元となりうる範囲
vuln_softs	vuln_soft	該当ソフトウェア群
vuln_soft	vendor, product, version	該当ソフトウェア情報 (複数)

ションの名称やバージョンの情報が含まれている．VA ツールはこれらの情報を取り出し，その情報をリクエストパラメタとして AVIP へリクエストを送り，脆弱性情報の XML ファイルを受け取る．表 2 で示すように，この XML には CVSS Vector が含まれる．VA ツールは，各脆弱性情報に対して CVSS Vector を読み込み，加藤らの提案する方法に従って CVSS 環境値を算出し，値として保持しておく．得られた全ての脆弱性情報に対して環境値を算出後，その中の最大値を NS 全体の影響度として最終的に出力する．

なお，CVSS では，脆弱性単体がシステムに与える影響度の算出が可能となっている．しかし管理者の観点では，システムが複数の脆弱性を抱える場合における，全体の影響を知ることが必要であり，これは CVSS では提供され得ない．したがって，本稿の VA ツールでは，算出した CVSS 環境値の中の最大値をシステム全体の影響度としたが，これに関してはさらに検討する余地を残している．

5.2 従来の脆弱性検査

従来の脆弱性検査には，大きく 2 つの問題点がある．

まず NS の外部から検査する場合，外部ネットワークからアクセス可能な，一部の機器のみの検査となり，一度に NS 全体を検査することが出来ない．次に NS の内

```
<?xml version="1.0" encoding="utf-8" ?>
- <vuln_list published_date="Thu Jan 1 12:34:56 +0900 2009"
    service_name="AVIP XML Feed">
  - <entry>
      <name>CVE-2009-0000</name>
      <published>2009-01-01</published>
      <modified>2009-01-02</modified>
      <severity>Medium</severity>
    - <cvss>
        <cvss_version>2.0</cvss_version>
        <cvss_vector>(AV:L/AC:M/Au:S/C:C/I:C/A:C)</cvss_vector>
        <cvss_base_score>6.6</cvss_base_score>
        <cvss_impact_subscore>10.0</cvss_impact_subscore>
        <cvss_exploit_subscore>2.7</cvss_exploit_subscore>
      </cvss>
    - <loss_types>
        <avail>true</avail>
        <conf>true</conf>
        <int>true</int>
        <sec_prot>admin</sec_prot>
      </loss_types>
    - <vuln_types>
        <access />
        <input />
        <design>true</design>
        <exception />
        <env />
        <config />
        <race />
        <other />
      </vuln_types>
    - <range>
        <local>true</local>
        <local_network />
        <network />
        <user_init />
      </range>
    - <vuln_softs>
      - <vuln_soft>
          <vendor>avip</vendor>
          <product>avip</product>
          <version>/1/</version>
        </vuln_soft>
      </vuln_softs>
    </entry>
</vuln_list>
```

図 4: AVIP の XML 出力例

部から検査する場合，実稼働環境に検査機器を持ち込むことになる．しかし内部においても，セグメント毎に検査機器を接続して検査しなければならず，非常に労力がかかる．また，そうした検査費用は高価である．

5.3 VA ツールのメリット

今回試作した VA ツールによって，前述の 2 つの問題点を解決することができる．すなわち，検査に必要な入力情報が XML ファイルのみであることから，実 NS 環境に直接接続せずに NS 全体を検査するといったことが可能となる．

6 まとめ

CVSS による定量評価を含む脆弱性情報を，検索結果として機械的に読み込み可能なデータ形式で取得するこ

表 2: AVIP に指定するリクエストパラメタ

検索キー	メソッド	説明
CVE 番号	cve={CVE 番号 }	CVE 番号の一致する脆弱性情報をリクエスト
公開/更新年月日	p_or_m={p|m}& date={yyyy-mm-dd}& b_or_a={b|a}	p で published，m で modified に対して検索を行う．b で date 以前，a で date 以降の情報に絞り込む
ベンダ名	match={full|part|front|rear}& vendor={ ベンダ名 }	full で完全，part で部分，from で前方，rear で後方の，4 つの一致パターンでベンダ名検索する
プロダクト名 (+バージョン)	match={full|part|front|rear}& product={ 製品名 } {&version={ バージョン名 }}?	match に関してはベンダ名検索と同様で，製品名検索を行う．バージョン名の指定は任意

とが，既存の脆弱性情報提供サービスでは不可能であることが調査により分かった．本論文では，脆弱性情報の検索結果を，CVSS を含む XML として返す Web API “AVIP” を開発した．また，加藤らの提案した NSQ モデルへの CVSS 適用法を用いた，NS における脆弱性影響評価ツールを試作し，AVIP を用いた NS の脆弱性影響評価の自動化，定量化を簡易ながら行った．

今後は，毎日自動的にネットワークを検査することで早期警戒を促すことを目的とした運用ツールなど，AVIP を利用した本論文の評価ツールを発展させた運用ツールの開発を目指す．

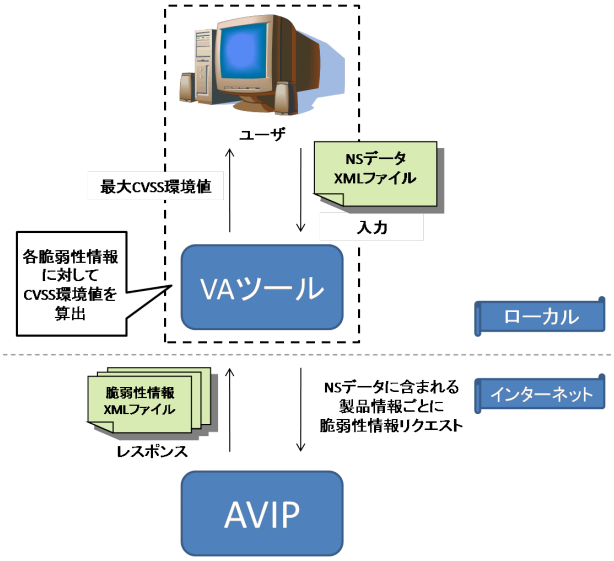

図 5: VA ツールの動作概要

参考文献

[1] Security Content Automation Protocol
http://nvd.nist.gov/scap.cfm

[2] National Vulnerability Database
http://nvd.nist.gov/

[3] Common Vulnerability Scoreing System
http://www.first.org/cvss/

[4] AVIP
http://www.avip.cipher.risk.tsukuba.ac.jp/

[5] 金岡晃，藤堂伸勝，加藤雅彦，岡本栄司,“ネットワークシステムの安全性定量化に向けた新たな表現モデルとアクセス制御解析,” 暗号と情報セキュリティシンポジウム 2008（SCIS2008），2008

[6] 金岡晃，藤堂伸勝，加藤雅彦，岡本栄司,“適切なアクセス制御状態にあるネットワークシステムの特徴抽出,” コンピュータセキュリティシンポジウム 2008（CSS2008）論文集，pp.557-562，2008

[7] A.Kanaoka, M.Kato, N.Todo, E.Okamoto, “Networked System Modeling and its Access Control Characteristic Analysis,” Proceedings of World Academy of Science, Engineering and Technology(WASET), Vol.35, pp.125-133, 2008

[8] 加藤雅彦，金岡晃，藤堂伸勝，岡本栄司,“ネットワークシステムにおける脆弱性影響度の定量化と可視化,” コンピュータセキュリティシンポジウム 2008（CSS2008）論文集，pp.551-556，2008

[9] Japan Vulnerability Notes
http://jvn.jp/

[10] JVN iPedia
http://jvndb.jvn.jp/

[11] The Open Source Vulnerability Database
http://jvn.jp/

[12] MyJVN
http://jvndb.jvn.jp/apis/myjvn/

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

[14] NVD XML Feed Documentation
http://nvd.nist.gov/download/nvdcve-xmldoc.cfm