脆弱性情報提供Web API “AVIP” の開発

◎原田敏樹(筑波大)、金岡晃(筑波大)
加藤雅彦(IIJテクノロジー)、岡本栄司(筑波大)

Outline

- 背景と目的
- 関連研究
  - ネットワークシステム表現モデル
  - 脆弱性定量化
- 脆弱性情報取得の問題
  - 脆弱性情報データベース
  - 脆弱性情報提供Web API
- AVIP の開発
  - AVIPの応用
- まとめ

背景と目的

- 背景
  - 現状のネットワークシステム(以下NS)管理
    - システム構成：テキストや表形式で管理
    - ネットワーク構造：描画ツール等で表記
      ⇒　脆弱性影響判断が困難
  - 既存の脆弱性情報提供サービス
    - 主にブラウザベース(HTML)
    - 機械的に利用可能な情報を検索・取得できない

- 目的
  - 脆弱性影響判断の自動化
  - 自動化に必要な脆弱性情報取得方法の確立

NS表現モデル(NSQモデル)

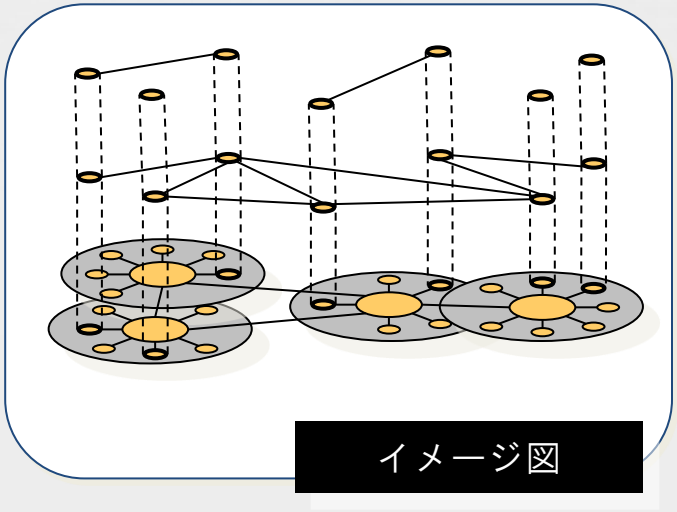
基本戦略

現状

物理的接続のみを反映したネットワーク表現

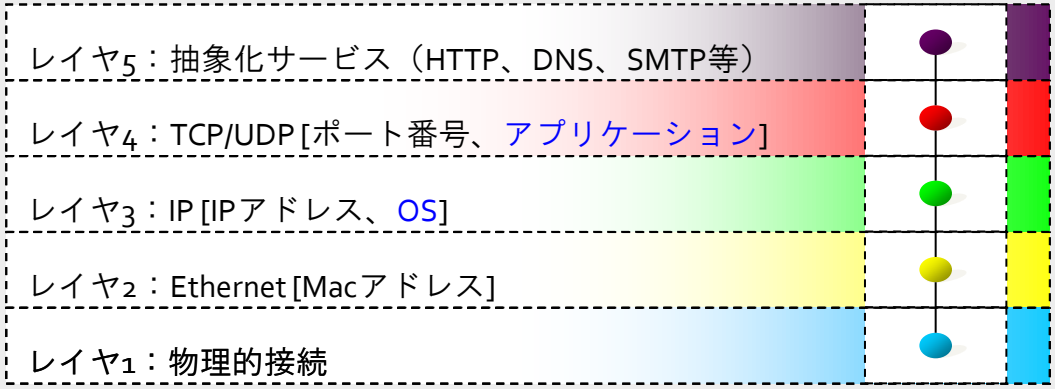
NSQモデル

TCP/IPの階層ごとに作られる論理ネットワーク
＋
階層ごとのネットワークの接続



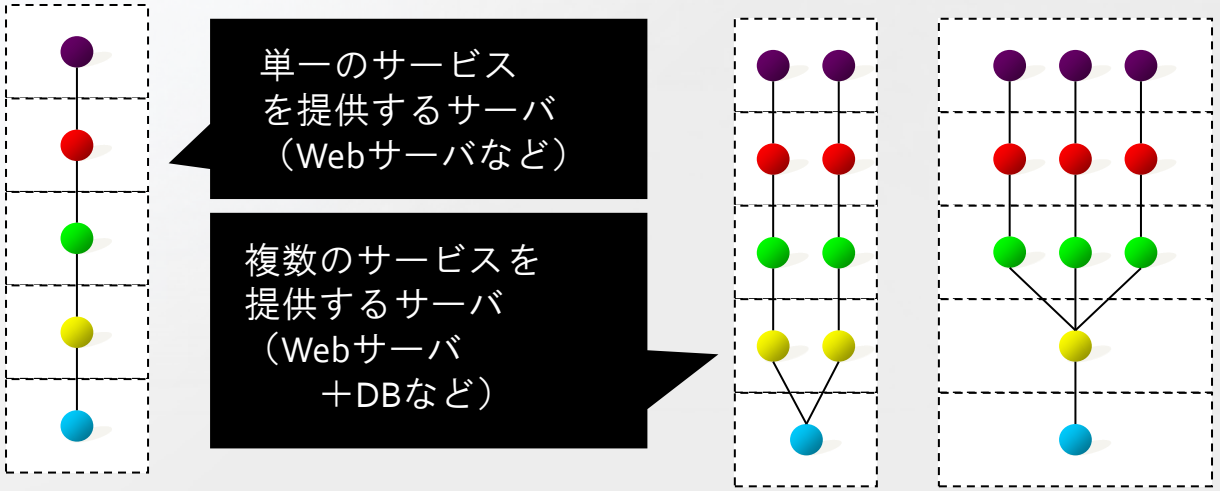
イメージ図

レイヤ定義

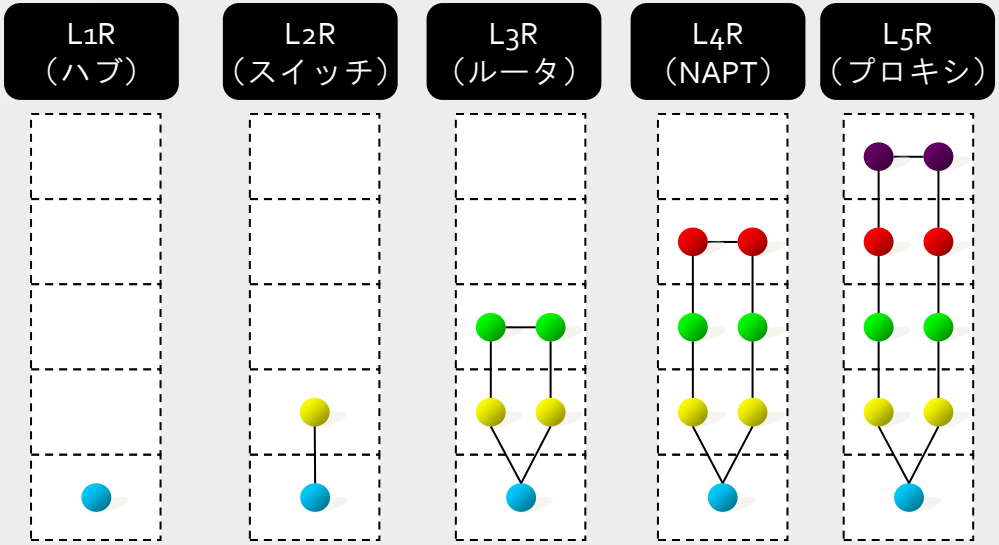


モジュール例

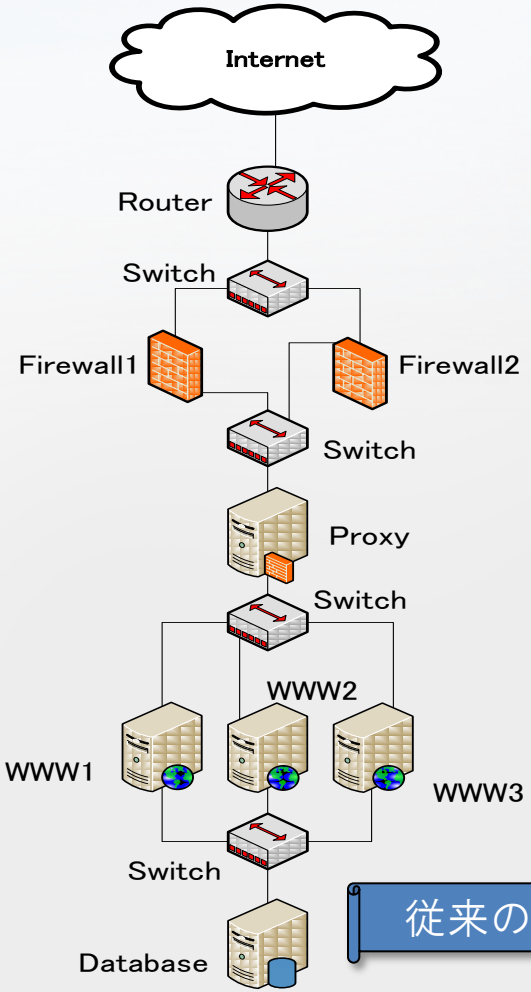
サーバ



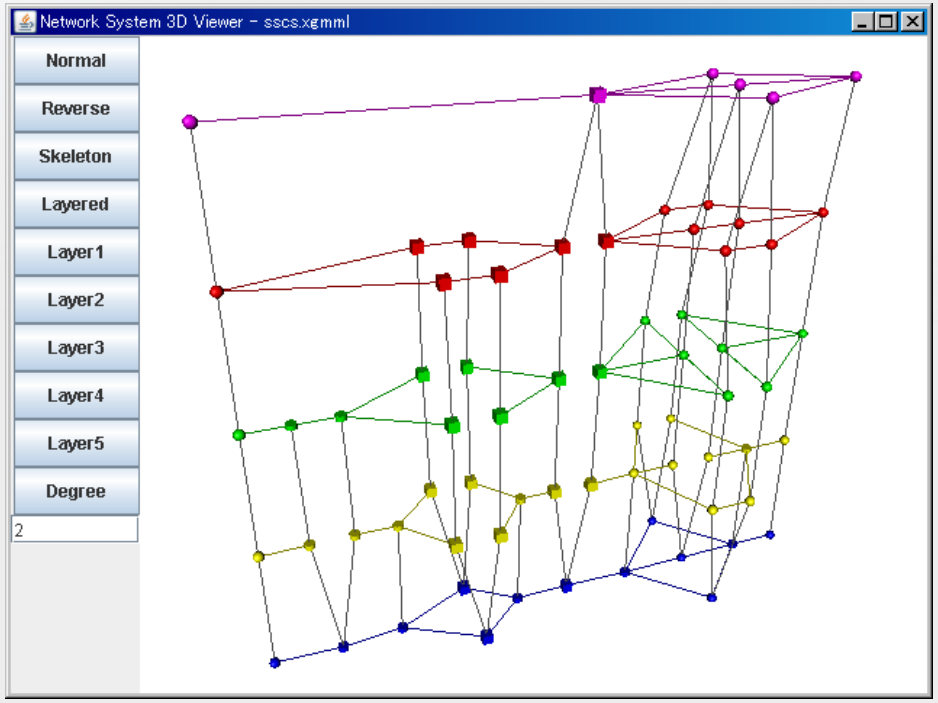
中継機器（機能）



NSQモデルによるNS表現例



従来の表現



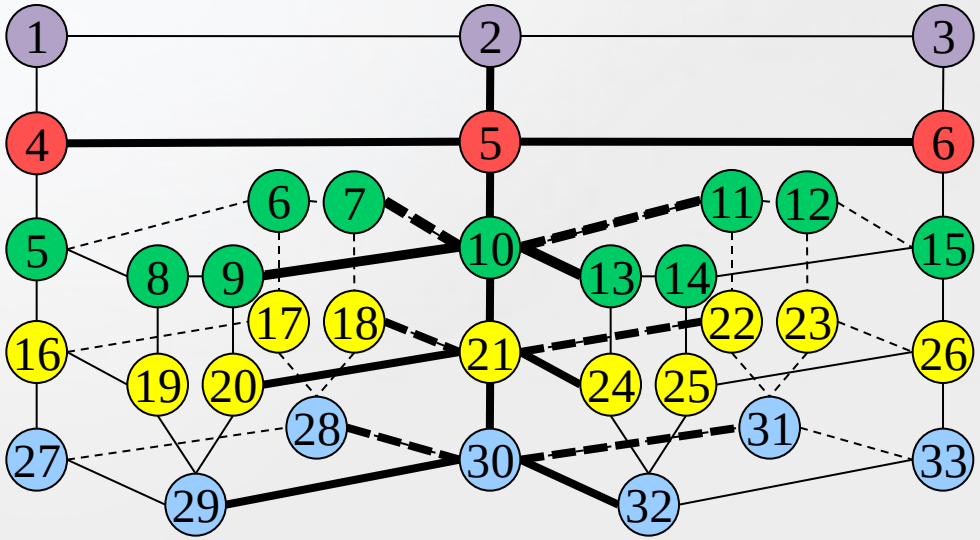
NSQモデルによる表現

脆弱性定量化

- Common Vulnerability Scoring System（CVSS）
  - 基本評価基準（Base Metrics）
    - 機密性、完全性、可用性に対する影響を攻撃容易性から評価（CVSS Vector）
    - 脆弱性固有の深刻度
    - 多くの組織で採用
  - 現状評価基準（Temporal Metrics）
    - 攻撃コードや対策情報の有無により評価
    - 脆弱性の現状の危険度
  - 環境評価基準（Environmental Metrics）
    - 攻撃を受けた際の二次被害規模や製品利用状況から評価
    - 最終的な脆弱性への対応を決定するための指標
    - 定性的で曖昧　⇒　定量評価が困難

各基準における評価は0.0～10.0の範囲で数値化

NSQモデルへのCVSS適用（1/2）



加藤らによる提案手法

- CVSS Vector：機密性・完全性・可用性のいずれかに全面的な被害があるか？
  - ない　⇒　脆弱性をもつノードに一次被害、その隣接ノードに二次被害
  - ある　⇒　脆弱性をもつノードを含むモジュール全体に一次被害、
    そのモジュールに隣接するノードに二次被害（レイヤ５を除く）
- 二次被害とは、あくまで周囲に影響が及ぶ可能性
  - 二次被害ノードの先のノードへの、更なる波及については言及しない

NSQモデルへのCVSS適用（2/2）

- CVSS環境値の算出に活用可能
  - CVSS環境値の定量評価

- CVSS Vectorを含む脆弱性情報が必要
  - プロダクトやバージョンで検索・取得
  - 機械処理が容易なデータ形式

既存の脆弱性情報提供サービスの調査を行う

既存の脆弱性情報データベース

- NVD（NIST）
  - National Vulnerability Database
  - ソフトウェア名、バージョン、ベンダ名、CVSS Vectorなどで検索可能（HTML）
  - RSS配信、XMLデータフィード（期間別）
- JVN iPedia（IPA）
  - ソフトウェア名、ベンダ名、CVSS基本値などで検索可能（HTML）
  - RSS配信
- OSVDB
  - The Open Source Vulnerability Database
  - ソフトウェア名、ベンダ名、脆弱性分類で検索可能（HTML）
  - XMLやMySQLでのエクスポート（XMLは更新停止中）

- JVN（IPA、JPCERT/CC）
  - Japan Vulnerability Notes
  - 告知が目的であるため、検索などは行えない

既存の脆弱性情報提供Web API

Web APIとは

- 他者が作成したデータなどをHTTPを用いてネットワーク上で処理を実行し、その結果を自分のコンテンツ内で使う為の技術
- 脆弱性情報の検索結果をXML形式で出力

- MyJVN Web API（IPA）
  - JVN iPediaが情報源
  - ソフトウェア名、ベンダ名で検索可能
  - レスポンスはRSS1.0ベース
- OSVDB API
  - OSVDBが情報源
  - 各種ID（ソフトウェア名、バージョン、ベンダ名）で検索可能

脆弱性情報取得の問題

	NVD	JVN iPedia	MyJVN Web API	OSVDB	OSVDB API
ソフトウェア検索	○	○	○	○	○
バージョン検索	○	×	×	×	○
XML	△ (期間別)	△ (期間別)	○	△ (全情報)	○
CVSS Vector	○	△ (基本値のみ)	△ (基本値のみ)	×	×

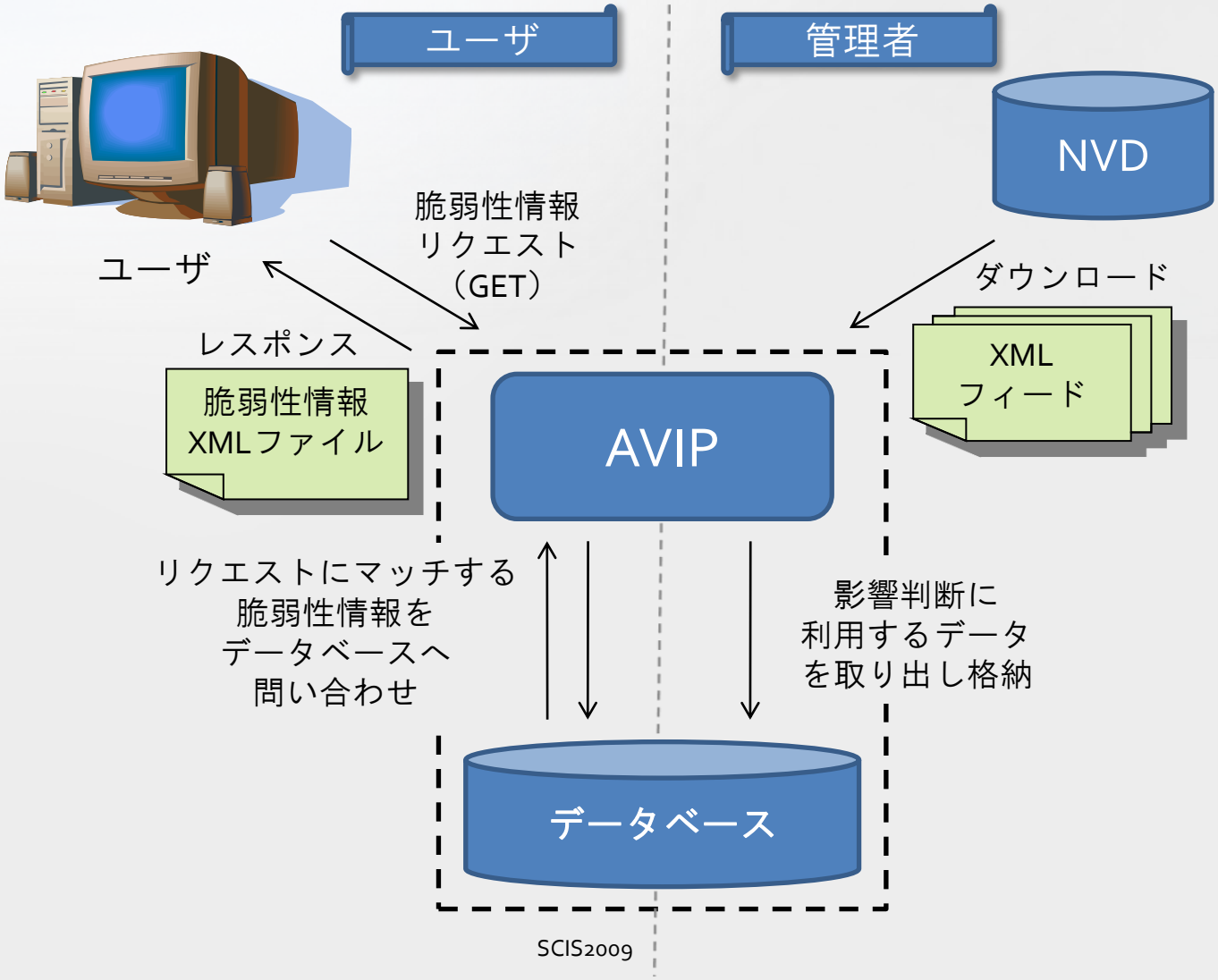
- 全ての項目を満たすものが存在しない
- NVDが最も多くの要項を満たす
  - 唯一CVSS Vectorを含む
  - XMLフィードを持つが、検索結果として返さない

NVDの情報をXMLで検索・取得するWeb APIを開発

AVIPの開発

- Automatic Vulnerability Information Provider
  http://avip.cipher.risk.tsukuba.ac.jp/
- 実装環境
  - Ruby on Rails 1.2.6 を利用
  - MySQL 5.0 を利用
- 脆弱性情報データフィードサービス
  - RESTインタフェース
  - NVD/CVE XML Feed with CVSS and CPE mappings version 1.2(NVD XML Feed) を利用
    - NVD の XML 情報を検索・取得

AVIP動作概要


ユーザ
管理者
NVD
脆弱性情報
リクエスト
（GET）
ユーザ
ダウンロード
レスポンス
脆弱性情報
XMLファイル
AVIP
XML
フィード
リクエストにマッチする
脆弱性情報を
データベースへ
問い合わせ
影響判断に
利用するデータ
を取り出し格納
データベース

リクエスト

- http://avip.cipher.risk.tsukuba.ac.jp/vulns?
  続けて以下のように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

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について検索 bでdate以前、aでdate以降
ベンダ名	match={full|part|front|rear} &vendor={ベンダ名}	matchで一致パターン指定 (fullで完全、partで部分、 frontで前方、rearで後方)
プロダクト名 (バージョン)	match={full|part|front|rear}& &product={ソフトウェア名} {&version={バージョン名}}?	ベンダ名検索と同様 バージョン指定を省くと 全バージョン対象

レスポンス例

- 内容はNVD準拠
- 脆弱性影響判断の自動化に不適な情報は省略
  - 内容説明
  - 対策方法
  - リファレンス

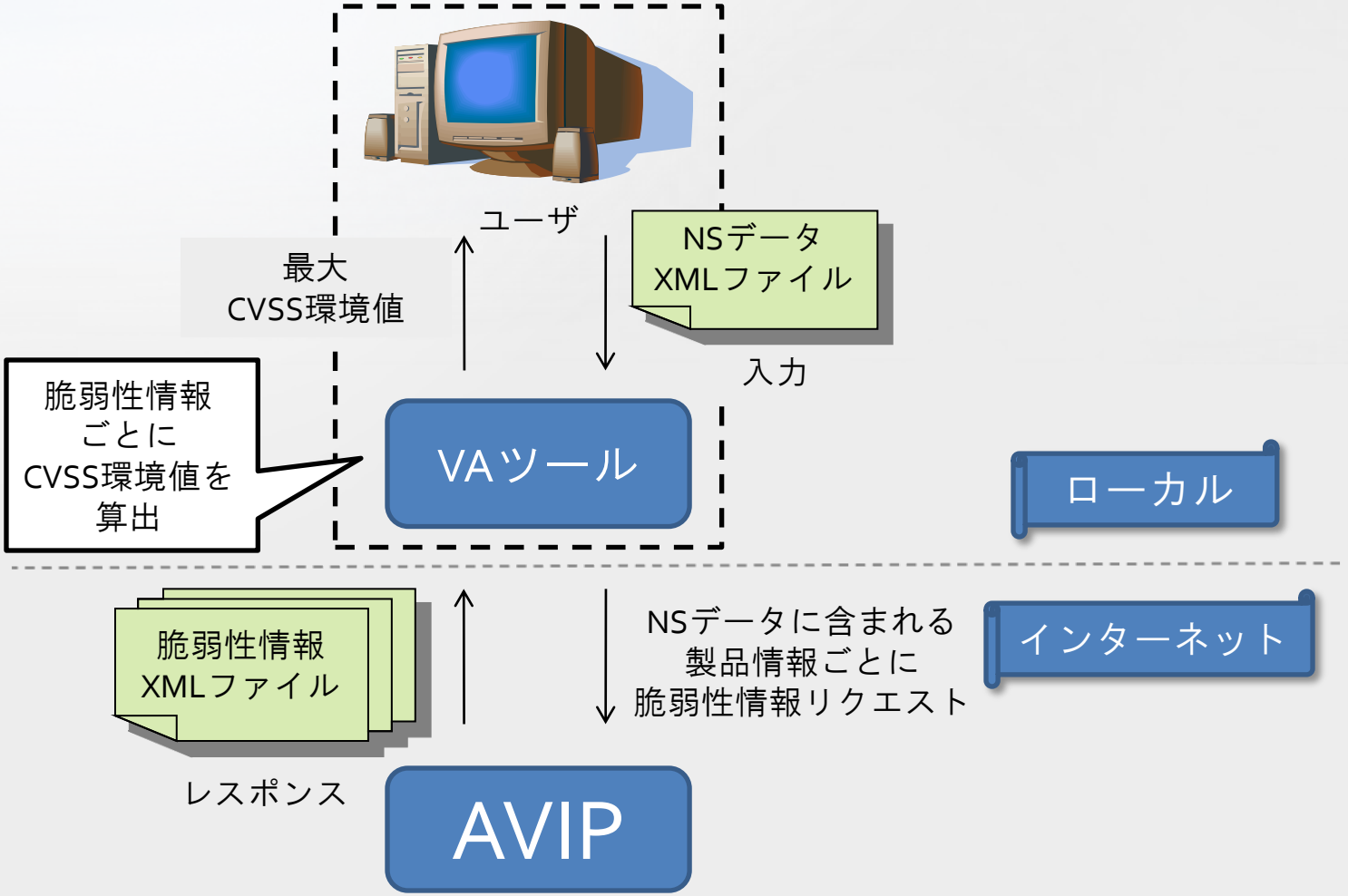
テキスト情報

```
<?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>
```

AVIPの応用

- VA(Vulnerability Assessment)ツール
  - AVIPで得られたXMLを利用してNS全体の脆弱性影響を簡易検査
  - 加藤らの提案手法を利用
  - 実装環境
    - Ruby 1.8.6

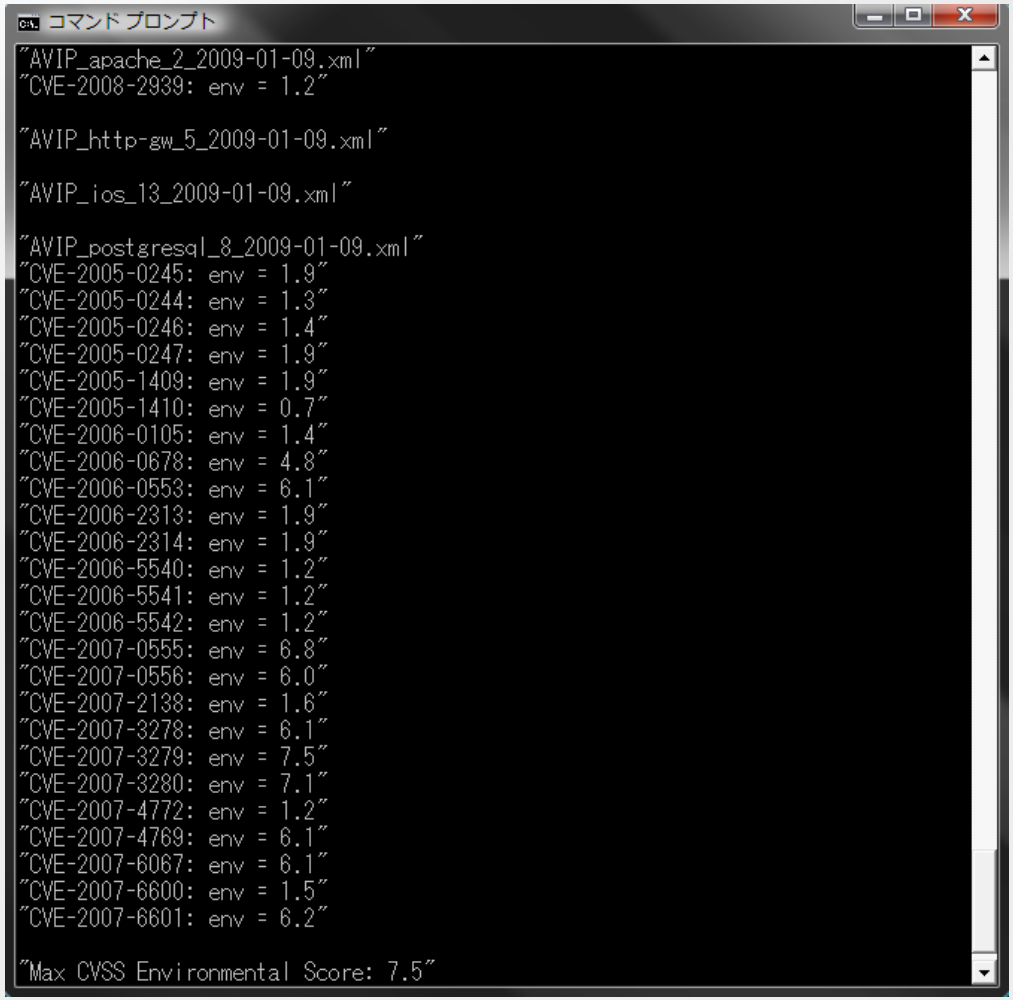
VAツール動作概要


ユーザ
最大
CVSS環境値
NSデータ
XMLファイル
入力
脆弱性情報
ごとに
CVSS環境値を
算出
VAツール
ローカル
脆弱性情報
XMLファイル
NSデータに含まれる
製品情報ごとに
脆弱性情報リクエスト
インターネット
レスポンス
AVIP

出力例

各脆弱性情報毎の
CVSS環境値

最大の
CVSS環境値



メリット

- 従来の脆弱性検査の問題点
  1. ネットワーク外部から検査
     - 外部からアクセス可能な一部の機器のみ
     - NS全体の検査が不可能
  2. ネットワーク内部から検査
     - セグメント毎に検査機器の再接続（労力）

- NS構成データを入力とすることにより

  1. 実機への接続が不要
  2. 設計時に検査可能
  3. NS全体を検査可能

まとめ

- 目的
  - 脆弱性影響判断の自動化
- 問題点
  - データ形式での脆弱性情報の検索・取得が不可能
- 解決手法
  - Web API “AVIP” の開発
  - AVIPを用いた簡易VAツールの開発
- 結果
  - 脆弱性情報をXMLで検索・取得が可能になった
  - NSQモデルとCVSSを基に脆弱性影響判断の自動化に成功した
- 今後
  - 早期警戒を促す運用ツールなど、AVIPの応用発展

ご清聴ありがとうございました

技術展示セッションにてAVIPをデモ展示中です