脆弱性対策のためのセキュリティ保護システムにおける脆弱性情報収集部の改善

18T327　中村 友昭 （最所研究室）

1　はじめに

当研究室では，脆弱性を利用した攻撃から組織内の機器を保護するセキュリティ保護システム BEYOND(Bring Enhancement Your Own Non-Vulnerable Device) の開発を行っている [1]．BEYOND はインターネット上に公開された脆弱性情報を収集する脆弱性情報収集部，組織内の機器情報を管理する IT 資産管理部，脆弱性情報と機器情報を照らし合わせて脆弱性の検知と制御方針を算出する影響算出部，影響算出部で算出された制御方針に基づいてアクセス制御を行うネットワーク制御部で構成される．BEYOND の構成を図 1 に示す．本稿では，BEYOND における脆弱性情報収集部の改善について述べる．

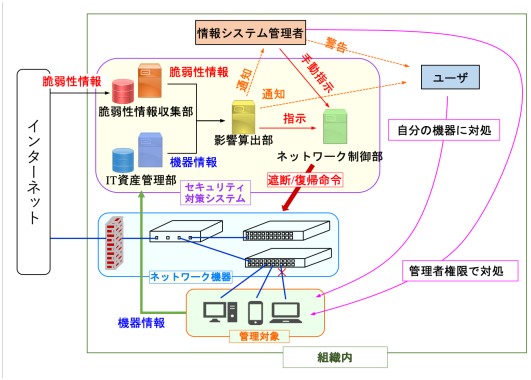

図 1: BEYOND の全体図

2　先行研究の課題

楠目により，開発された脆弱性情報収集部では，JVN iPedia から脆弱性情報を収集していたが，次に示す課題があった．

課題①: 影響算出部では，脆弱性情報 DB と IT 資産管理 DB の内容をマッチングして脆弱性を持つ機器を判定するが，本機構の DB 構成では，テーブル同士が関連付けられていなかった．そのため，影響算出の際に必要な情報を取り出すことが難しい状態となっていた．

課題②: 脆弱性情報収集部では常に新しい情報を収集することが求められる．しかし，本機構では収集範囲を手動で設定する実装となっていた．そのため，継続的な情報収集が考慮されていなかった．

課題③: JVN iPedia に登録されている書式の表記ゆれを考慮していなかったため，影響算出部で必要な情報が一部抽出できておらず，DB に登録されていなかった．

課題④: 先行研究では，バージョン情報が誤って DB に登録されていた．これは，バージョンの開始位置と終了位置が入れ替わって登録されていたり，バージョン情報とは関係のない情報が DB に登録されていた．そのため，影響算出の際に扱うことができなかった．

3　課題の解決方針

2 章で述べた課題の解決方針を以下に示す．

①DB 構成を再構成する．関連する情報が登録されているテーブル同士を紐付け，冗長なテーブルは切り分け，必要ないと判断したテーブルは削除し，今まで収集できていなかった情報を登録するためのテーブルを追加した．

②初回収集時と 2 回目以降の収集時で処理を分け，日付指定で情報を収集することで，継続的に情報収集を行えるようにする．

③JVN iPedia で公開されている情報のフォーマットに合わせて収集する．

④JVN iPedia ではバージョン情報は日本語で記載されていたり，CVE を参照しているものが多いため，バージョン情報は NVD から収集を行う．その際，バージョン情報の記述方式が複数あるため，それらを識別できるように DB に登録する．

4　提案システム

3 章で述べた課題の解決方針に基づいた実装について述べる．

① DB の再構成: 再設計した脆弱性情報 DB を図 2 に示す．先行研究と大きく異なる点について説明する．先行研究で内容が他のテーブルと重複しているテーブルや，使用用途が不明なテーブルは削除した．また，脆弱性の深刻度を表す CVSS スコアはバージョンが 2 つ存在するが，先行研究では，片方のバージョンのみ登録していた．CVSS スコアは今後新たにバージョンが追加されるため，CVSS のバージョンごとにテーブルを追加した．さらに，バージョン情報を表すテーブルにバージョンの開始位置と終了位置の組み合

わせを識別するカラムを追加した．最後に，バージョン情報はNVDから収集しているため，JVN iPediaの情報と紐付けるためにそれぞれの情報源の識別子を関連付けるテーブルを作成した．これにより，関連する情報を取り出せるようになった．

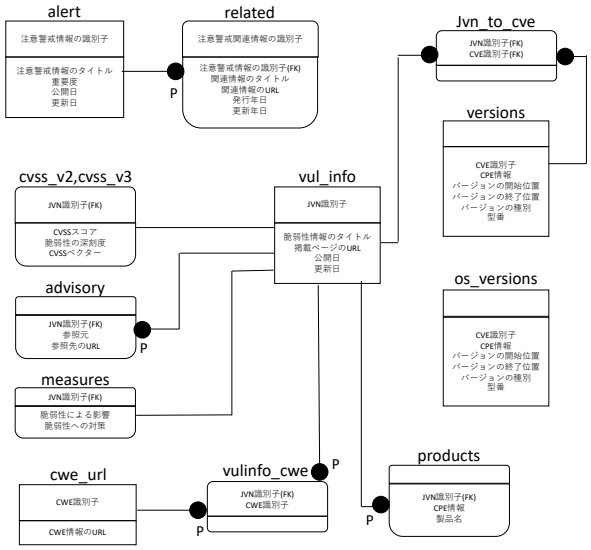

図 2: 再構成したDB構成

② 継続的な情報収集: JVN iPediaとNVDからAPIを用いて直近3日間に追加，変更された脆弱性情報を収集する機能を実装し，継続的な情報収集を行えるようにした．新規の脆弱性情報はDBに追加し．変更があった脆弱性情報はDBに登録されている情報と比較して，更新処理を行った．

③ 表記ゆれの対応: JVN iPediaから収集する情報はJSON形式のデータに変換して扱っている．変換したデータはPythonにおける辞書型とリストが混ざったデータとなっている．先行研究における取得漏れの原因は辞書型とリストを区別していないことによるものだった．例を挙げると，CVSSスコアはバージョンが2つ記載されている場合はそれぞれの情報がリストの中に格納され，1つだけの場合は辞書型で格納されている．これらの場合分けを行うことで，取得漏れを無くした．

④ バージョン情報の取得: バージョン情報の取得はJVN iPediaでは表記ゆれが大きいため，NVDを用いて行う．バージョン情報は，バージョン情報が存在しない場合，OSやハードウェアなどの環境に依存せずに脆弱性が影響する場合，特定の環境でのみ脆弱性が影響する場合の3パターンがある．これらの場合分けを行い，OSやハードウェアのバージョンを示すテーブルと脆弱性を持つ製品のバージョンを示すテーブルで分けて登録する．また，バージョンの開始位置と終了位置の組み合わせを識別するために，バージョンタイプという情報をDBに登録している．これらの組み合わせは表1のように9通り存在し，それぞれに番号を割り当て判定を行う．

表 1: バージョンの対応関係

バージョンタイプ	組み合わせ
0	以上以下
1	以上未満
2	以上
3	超過以下
4	超過未満
5	超過
6	以下
7	未満
8	型番に記載

5 評価

開発した脆弱性情報収集部の機能評価と先行研究との比較について述べる．機能評価は表2について行った．その結果取得漏れは無かった．また，正確に取得できているか調査を行った結果，JVN iPediaから収集したものは全て正確に登録できていた．versionsテーブルに関しては，2021年の脆弱性情報を100件先頭から取り出して行った結果1件登録するテーブルが間違っていた．次に，先行研究との比較を行った．これは，先行研究で問題のあったCVSSスコアとバージョン情報に限定して行った．2021年の1年間分の脆弱性情報を用いて比較した結果，先行研究では201件取得漏れが存在し，改善したシステムでは，全て取得できていた．バージョン情報は2021年の脆弱性情報を100件先頭から取り出して行った結果，5件取得漏れが存在し，残りの95件中81件が正確に取得できていなかった．改善したシステムでは，100件中99件正確に取得できていた．

表 2: 評価を行うテーブル

情報源	テーブル名
JVN iPedia	vul_info,cvss_v2,cvss_v3,measures,advisory
NVD	versions,os_versions

6 終わりに

本稿では，BEYONDにおける脆弱性情報収集部の改善を行った．脆弱性情報収集部の改善の結果収集精度を大きく向上させることができた．JVN iPediaだけでなくNVDから収集を行うことにより，先行研究に比べ多くの脆弱性情報を収集できるようになった．

参考文献

[1] 楠目幹 他，“脆弱性情報を利用したゼロデイ攻撃対策システムにおける構成情報収集機能の実装及び脆弱性評価機能の設計”ISEC2019-12，Vol.119, no.140, pp.1-6(2019),