4ZA-03

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 DB 構築

楠目幹† 最所圭三† 喜田弘司†
香川大学

1. はじめに

ソフトウェアの脆弱性は日々発見され続け，パッチがリリースされるまでの間に行われるゼロデイ攻撃が深刻な問題となっている．ゼロデイ攻撃そのものを防ぐことが非常に難しいことも，問題をより深刻化させている．本研究では，インターネット上で公開されている脆弱性情報等をもとに，ゼロデイ攻撃への早期対応と，被害の緩和を目的としたゼロデイ攻撃対策システムを考案している．本稿では，脆弱性情報の収集及び DB 化について述べる．

2. ゼロデイ攻撃対策システムの概要

脆弱性が発見され，パッチがリリースされるまでの期間，パッチを待つだけで，何も対策を取っていない場合が多い．しかし，事前に脆弱性が影響を及ぼすかどうか，及ぶ場合はどのマシンに影響が及ぶのか，何台のマシンに影響が及ぶのか，誰が管理するマシンに影響が及ぶのか，加えて早急に対応する必要があるかどうかが把握できれば，パッチがリリースされるまでの間に何らかの対策を取ることができる．

本研究では，このような対策を講じることを支援するゼロデイ攻撃対策システムを考案した[1]．提案システムを図 1 に示す．インターネット上に公開されている脆弱性情報等の公開情報と，システム内のマシン情報等の個別情報を利用し，影響算出部で影響範囲の予測を行い，対策算出部でそれに合わせた対策を生成する．

脆弱性情報やパッチの有無は，インターネット上から入手する．代表的なサイトとして，JVN[2]がある．管理者情報，ソフトウェアのバージョン情報，パッチの適用状況は，対象システム内の各サーバから収集する．これらの情報は，定期的なタイミングで当システムにアップロードし，最新の状態に保つようにする．

収集した情報をもとに，当システム内の影響算出部で，脆弱性情報とソフトウェアの合致、パッチの有無等から，対象システムへの影響範囲を予測する．対策算出部では，管理者に対して予測した影響範囲に合わせた対策を生成する．例えば，パッチの適用，ソフトウェアアップデート，影響を受けるサービスの遮断等がある．これらの情報は，管理者や利用者に通知される．

影響範囲は，脆弱性の該当リストとして表示される．サーバ情報には，マシン名や IP アドレス等が含まれる．脆弱性の影響範囲によって，対応する管理者は変化する．例えば，特定のサーバ内のみに脆弱性が存在する場合は，マシンの管理者が対象となる．システム全体に脆弱性が存在する場合は，システム全体の管理者が対象となる．緊急度は，影響範囲や対象システムのパッチの適用状況に応じて判断される．対応策は，管理者や利用者にメールで通知される．

当システムを活用することで，ゼロデイ攻撃に対する事前対策を取ることができる．

3. 脆弱性情報の収集と DB 化

JVN では、脆弱性情報を JVN iPedia[3]にデータベースとして格納している．JVN iPedia からの情報の収集には，MyJVN[4]が提供している MyJVN API を用いる(図 2).

MyJVN API は 6 つのメソッドを持つ．注意警戒情報を取得する getAlertList，ベンダ一覧を取得する getVendorList，製品一覧を取得する getProductList，脆弱性対策の概要情報を取得する getVulnOverviewList，脆弱性対策の詳細情報を取得する getVulnDetailInfo，共通脆弱性評価システムである CVSSv3 及び CVSSv2 の統計情報を取得する getStatistics である．また，メソッドごとにパラメータを詳細に指定でき，より精度の高い情報を得ることができる．

これらの情報は XML データで取得され，それを DB 化する．DB 化には Python で記述したスクリプトを用いる．まず，スクリプト内で MyJVN API ごとに取得した XML データを JSON データに変換する．この JSON データではタグや属性等が key となり，それに対応する値が value となる．

The Database Construction in Countermeasure System for Zero Day Attack Using Vulnerability Information
†Motoki KUSUME, Kagawa University
†Keizo SAISHO, Kagawa University
†Koji KIDA, Kagawa University

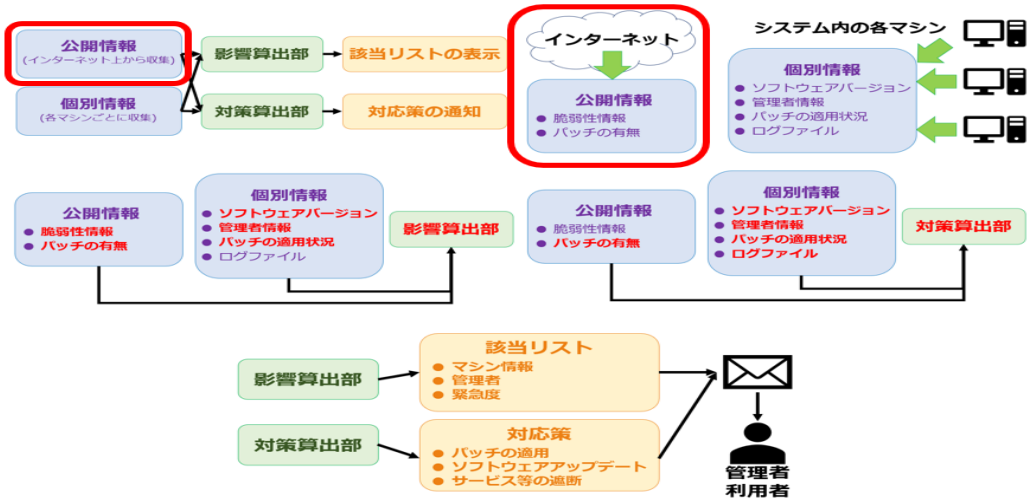

図 1 ゼロデイ攻撃対策システムの概要

次に，変換したデータから DB 化したい情報を取り出し，辞書化する．最後に，辞書化した情報を DB へ追加する(図 3)．DB のテーブルは，MyJVN API ごとに分割する(図 4).

getAlertList では注意警戒 ID，タイトル，リンク，発見日，更新日，カテゴリを用いる．getVendorList ではベンダ ID，ベンダ名，CPE 名を用いる．getProductList ではプロダクト ID，ベンダ ID，CPE 名，製品名を用いる．getVulnOverviewList ではセキュリティ情報 ID，タイトル，セキュリティ情報のリンクを用いる．getVulnDetailInfo では脆弱性情報 ID，タイトル，CVSS スコア，CVSS 短縮表記を用いる．getStatistics は単純な統計データであるため，現時点では DB 化しない．

DB 化した情報は，影響算出部での影響範囲の推測や，対策算出部での対策の生成に利用する．

4. おわりに

脆弱性情報を利用したゼロデイ攻撃対策システムを考案している．今回は MyJVN API を用いた脆弱性情報の収集と，収集した情報の DB 化を行った．今後は個別情報の DB 化と影響算出部及び対策算出部の実装を目標としている．

5. 参考文献

[1] 楠目幹, 最所圭三, 喜田弘司 脆弱性情報を利用したゼロデイ攻撃対策システムの考案
平成 30 年度 電気関係学会四国支部連合大会論文集, 16-4, p.194, 2018.9
[2] JVN(Japan Vulnerability Notes), available at https://jvn.jp/ (2019/01/11 参照)
[3] JVN iPedia, available at https://jvndb.jvn.jp/ (2019/01/11 参照)
[4] MyJVN, available at https://jvndb.jvn.jp/apis/myjvn/index.html (2019/01/11 参照)

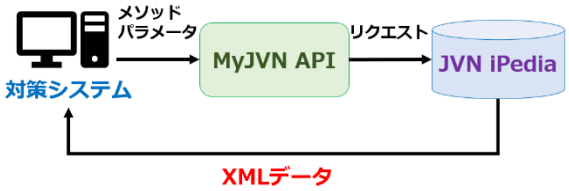

図 2 MyJVN API を用いた脆弱性情報の取得

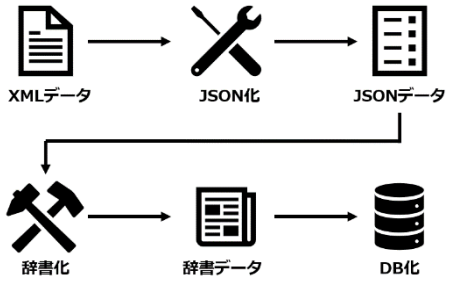

図 3 取得した脆弱性情報の DB 化

API名	table名
getAlertList	alert
getVendorList	vendor
getProductList	product
getVulnOverviewList	overview
getVulnDetailList	detail

図 4 API ごとの DB におけるテーブル名