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と DB 構築

15T222 楠目 幹 (最所研究室)

インターネット上に公開されている脆弱性情報をもとに，システムへの影響範囲と対応策を管理者に提供する，ゼロデイ攻撃対策システム及びシステムで使用する DB について述べる．

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する．本稿では，ゼロデイ攻撃対策システムの概要，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する範囲や，緊急性等が把握できれば，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と考えた．

本研究では，事前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(図 1)．このシステムでは，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，影響算出部で脆弱性の影響範囲を予測し，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[1] がある．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．管理者情報や外部への公開情報は，対象システムを保有する組織の管理情報として収集する．これらの情報は，定期的なタイミングで更新し，最新の状態を保つようにする．

収集した情報をもとに，本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，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．生成する対策の例として，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通知する．サーバ情報には，マシン名や IP アドレス等が含まれる．通知する管理者は，脆弱性の影響範囲によって変化する．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する．

本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．

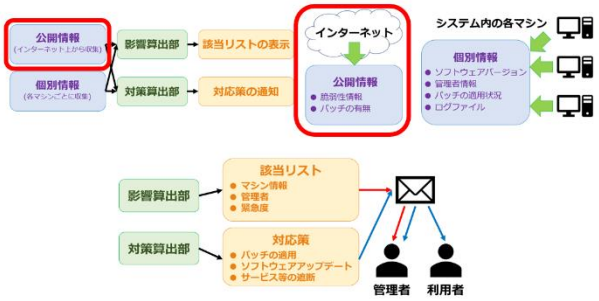

図 1 ゼロデイ攻撃対策システムの概要

3. 脆弱性情報 DB の設計

脆弱性の影響範囲の予測や対策の生成を行うには，より多くの脆弱性情報が必要であり，対策システムがその情報を保持しておく必要がある．本システムでは，情報源として JVN を用いる．

JVN では，脆弱性情報を JVN iPedia[2]に DB として格納している．JVN iPedia からの情報の収集には，MyJVN[3]が提供している MyJVN API を用いる．

MyJVN API は脆弱性の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 の 6 つのメソッドを持つ．また，メソッドごとにパラメータを詳細に指定でき，より詳細な情報を得ることができる．

JVN iPedia から取得できる情報のうち，本システムにおいて DB 化する必要のある情報は，脆弱性の内容，脆弱性情報の発見日及び更新日，対象の製品やソフトウェア，それらを提供しているベンダ，CVSS スコア，パッチ等の対策である．これらの情報は影響算出部での影響範囲の予測や，対策算出部での対策の生成に用いる．メソッドごとに取得できる情報が異なるため，メソッドごとに DB を作成する(図 2)．

alertDB では，注意警戒情報 ID，タイトル，発見日及び更新日，脆弱性情報のカテゴリをフィールドとする．vendorDB では，ベンダ ID，CPE 名，ベンダ名をフィールドとする．productDB では，製品 ID，ベンダ ID，CPE 製品名，製品名をフィールドとする．overviewDB では，脆弱性情報 ID，タイトル，概要が記載されたリンクの URL をフィールドとする．detailDB では，脆弱性情報 ID，タイトル，CVSS スコア，CVSS ベクトルをフィールドとする．

4. 脆弱性情報の収集及び DB 化

MyJVN API のメソッドを用いて JVN iPedia から取得した脆弱性情報は XML データで取得する(図 3)．その中から必要な情報を抽出し DB 化する．脆弱性情報の収集及び DB 化には Python で記述したプログラムを用いる．まず，プログラム内でリクエストとして JVN iPedia に送信する URL を生成する．次に，リクエスト結果として取得した 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．次に，変換した JSON データから DB 化する情報を抽出し，辞書データ化する．最後に，辞書データ化した情報を DB へ追加する(図 4, 5)．

5. 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の考案し，脆弱性情報の収集及び DB 化を行った．今後は個別情報の DB 化や影響算出部及び対策算出部の実装を目標としている．また，大量の脆弱性情報を効率的に収集する方法の検討や，構築した DB の妥当性の評価を行う必要がある．

6. 参考文献

[1] JVN(Japan Vulnerability Notes), available at https://jvn.jp/ (2019/02/18 参照)

[2] JVN iPedia, available at https://jvndb.jvn.jp/ (2019/02/18 参照)

[3] MyJVN, available at https://jvndb.jvn.jp/apis/myjvn/index.html (2019/02/18 参照)

メソッド名	DB名
getAlertList	alert
getVendorList	vendor
getProductList	product
getVulnOverviewList	overview
getVulnDetailList	detail

図 2 メソッド名と対応する DB 名

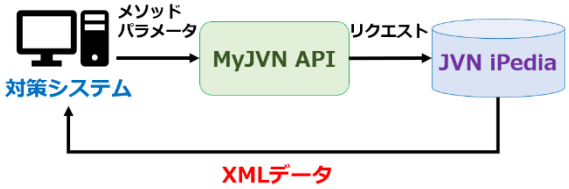

図 3 MyJVN API を用いた脆弱性情報の取得

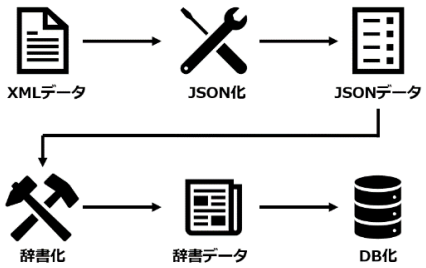

図 4 取得した脆弱性情報の DB 化

vid	cpe	vname
806	cpe:/:2daybiz	2daybiz
1200	cpe:/:2x	2X Software
1229	cpe:/:11in1	11in1
1320	cpe:/:111webcalendar	111webcalendar
1928	cpe:/:1-script	1-script
1929	cpe:/:1024cms	1024cms
1930	cpe:/:123flashchat	123flashchat
1931	cpe:/:1scripts	1scripts
1932	cpe:/:1st_news	1st news
1933	cpe:/:20_20_applications	20 20 applications

図 5 DB 化した脆弱性情報の例