公開脆弱性データベースを利用した対応順位決定支援ツール

山森　一人[a)]・齊藤　燎[b)]・相川　勝[c)]・井上　健太郎[d)]

Corresponence Order Decision Support Tool using Public Vulnerability Database

Kunihito YAMAMORI, Ryo SAITO, Masaru AIKAWA,Kentaro INOUE

Abstract

Internet is a convenient and an essential technology for contemporary life, but we are simultaneously confronted with cyber attack. Especially attacks via systems' vulnerability can not be prevented unless countermeasures are taken. However,system administrator for household use PC is usually the user himself/herself, computer skill and the knowledge for administration are sometimes not enough.Even if an administrator will have enough skill, the number of administrable computers is limited. In this research,We propose a correspondence order decision support tool using public vulnerability database.Our tool scans the software in the computers, and recoreds the vendor names, versions and so on.Then our tool inquires the vulnerabilities to the public database, and shows them in the critical order.

Keywords:　Security, Vulnerability, Database, CVE, CVSS value

1.　はじめに

近年、コンピュータやスマートフォンなどの情報通信機器の普及や、無料のアクセスポイントの整備などでインターネットへのアクセスが身近になった[1)]。インターネットは日常生活には欠かせないが、不特定多数のユーザが利用しているため、悪意あるユーザが一定数存在する。悪意あるユーザがサイバー攻撃を行い、個人情報などの情報資産に損害を与えることもある。例えば、2018年1月26日に仮想通貨取引所「コインチェック」がサイバー攻撃を受け、同社が保有していた仮想通貨「NEM」が不正送金された事件は記憶に新しい。詳しい原因は究明中であるが、セキュリティ対策が不十分であったことが原因であると言われている。こうしたサイバー攻撃から情報資産を守るためには、セキュリティ対策が重要である。

セキュリティ対策には多種多様な方法[2)]があるが、ファイアウォールや権限の制限によるアクセス制限により、不正アクセスなどに関しては防御することが可能である。しかし、いくらセキュリティ対策をしていても、未知の脆弱性を悪用した攻撃は防ぐことができない。未知の脆弱性は、発見されなければ対策が行われない。セキュリティホールは、発見報告されてから対策パッチの制作が行われるため、対策パッチが提供されるまでにはタイムラグが存在する。この間は、ソフトウェアを使うユーザ側で対策を行う必要がある。

脆弱性の情報を入手するためには、個人、またはセキュリティ担当者が各種脆弱性情報を調査したり、IPAが提供するリアルタイム配信型のサイバーセキュリティ注意喚起サービスicatなどを用い入手する必要がある。

コンピュータに脆弱性が存在するかを調べるためには、導入されているソフトウェアのバージョンの把握が必要となる。だが、セキュリティに対する知識やコンピュータスキルには個人差があり、ユーザが脆弱性の調査を行えるとは限らない[3)]。さらに、セキュリティ担当者が自身の管理下のコンピュータの脆弱性の有無を調べる場合、データセンターなど数千台のサーバに導入されているソフトウェアを網羅することは困難である。脆弱性の中には早急な対応が必要なものもあり、危険性の高い脆弱性への対応を後回しにすると、対応するまでの間に悪用される可能性がある。

本研究では、システム内のソフトウェアに潜在する脆弱性を検知警告し、対応順位の決定支援を行うシステムを提案する。脆弱性の検知には公開されている脆弱性情報を用い、順位決定には当該脆弱性の危険度を示す基準値を用いる。

a) 情報システム工学科教授
b) 情報システム工学科
c) 宮崎大学工学部教育研究支援技術センター技術職員
d) 情報システム工学科助教

2. 脆弱性情報

2.1. CVE

CVE（Common Vulnerabilities and Exposures）[4]は MITRE 社[5]が採番をしており、個々の製品の脆弱性に対し、ベンダー非依存の業界標準名を「CVE-西暦年-4 桁以上通番」の形式で付与している。本研究ではこれを CVE 番号と呼ぶ。

2.2. NVD

NVD（National Vulnerability Database）[6]は NIST（National Institute of Standards and Technology）[7]が管理している脆弱性データベースであり、各 CVE 番号に対応した詳細情報を提供している。NVD は、3.2 節で説明する CVSS（Common Vulnerability Scoring System）[8]により脆弱性の危険度評価を行っており、個々の脆弱性に対する深刻度の把握が可能となっている。

2.3. CVSS

2.3.1. 概要

CVSS は FIRST(Forum of Incident Response and Security Teams)が管理母体となり、適用推進や仕様改善を行っている。CVSS は脆弱性に対する汎用的な評価手法であり、ンダーに依存しない共通の評価方法を提供し、脆弱性の深刻度を同一の基準で定量的に比較することができる。2007 年 6 月には CVSSv2 が公開されている。その後、仮想化やサンドボックス化が進んできたことから、コンポーネント単位で評価する手法として、2015 年 6 月に CVSSv3 が公開された。CVSSv3 が公開される以前の NVD の脆弱性情報については、特別なものをのぞき再計算はされていない。そのため、過去の脆弱性については CVSSv2 による基本値が必要となる。

2.3.2. CVSSv2

CVSSv2 の基本値は、表 1 に示したパラメータを用いて式(1)で計算される。

$$\text{基本値} = \big((0.6 \times IM) + (0.4 \times EA) - 1.5\big) \times f(IM) \tag{1}$$

$$IM = 10.41 \times (1 - (1 - C) \times (1 - I) \times (1 - A))$$

$$EA = 20 \times AV \times AC \times Au$$

$$f(IM) = \begin{cases} 0, & (IM = 0) \\ 1.176, & (IM > 0) \end{cases}$$

表 1 CVSSv2 パラメータ名と評価値

基本評価基準	評価結果	値
攻撃区分(AV)	ローカル(L)	0.395
	隣接	0.646
	ネットワーク	1.0
攻撃条件の複雑さ(AC)	高(H)	0.35
	中(M)	0.61
	低(L)	0.71
攻撃前の認証要否(Au)	複数(M)	0.45
	単一(S)	0.56
	不要(N)	0.704
機密性への影響(C)	なし(N)	0.0
	部分的(P)	0.275
	全面的(C)	0.660
完全性への影響(I)	なし(N)	0.0
	部分的(P)	0.275
	全面的(C)	0.660
可用性への影響(A)	なし(N)	0.0
	部分的(P)	0.275
	全面的(C)	0.660

表 2 CVSSv2 深刻度レベル分け

深刻度	CVSS 基本値
危険	7.0~10.0
警告	4.0~6.9
注意	0.0~3.9

式(1)で導き出された基本値に基づき、当該ソフトウェアの脆弱性は表 2 の深刻度に分類される。

2.3.3. CVSSv3

CVSSv3 の基本値は、表 3 に示したパラメータを用いて式(2)で計算される。なお基本値の最大値は 10 となっている。

$$\text{基本値} = \begin{cases} 0 & , \quad (\text{IM} = 0) \\ IM + EA \quad , (S = U) & \\ 1.08 \times (IM + EA) \ , (S = C) & , \quad (\text{IM} > 0) \end{cases} \tag{2}$$

$$BM = 1 - (1 - C) \times (1 - I) \times (1 - A)$$

$$\text{IM} = \begin{cases} 6.42 \times \text{BM} \ , & (\text{S} = \text{U}) \\ 7.52 \times (\text{BM} - 0.029) & \\ -3.25 \times (\text{BM} - 0.02)^{15}, & (\text{S} = \text{C}) \end{cases}$$

$$\text{EA} = 8.22 \times \text{AV} \times \text{AC} \times \text{PR} \times \text{UI}$$

表 3　CVSSv3 パラメータ名と評価値

評価項目	評価結果	値	スコープ変更あり
攻撃元区分(AV)	ネットワーク(N)	10.85	
	隣接(A)	0.62	
	ローカル(L)	0.55	
	物理(P)	0.2	
攻撃条件の複雑さ(AC)	低(L)	0.77	
	高(H)	0.44	
必要な特権レベル(PR)	不要(N)	0.85	
	低(L)	0.62	0.68
	高(H)	0.27	0.5
ユーザ関与レベル(UI)	不要(N)	0.85	
	要(R)	0.62	
スコープ(S)	変更なし(U)	−	
	変更あり(C)	−	
機密性への影響(C)	高(H)	0.56	
	低(L)	0.22	
	なし(N)	0	
完全性への影響(I)	高(H)	0.56	
	低(L)	0.22	
	なし(N)	0	
可用性への影響(A)	高(H)	0.56	
	低(L)	0.22	
	なし(N)	0	

表 4 CVSSv3 深刻度レベル分け

深刻度	CVSS 基本値
緊急	9.0~10.0
重要	7.0~8.9
警告	4.0~6.9
注意	0.1~3.9
なし	0.0

式(2)で導き出された基本値に基づき、当該ソフトウェアの脆弱性は表 4 の深刻度に分類される。CVSSv3 の深刻度は CVSSv2 の深刻度よりも詳細に分類される。

3.　提案手法と評価

3.1.　要件定義

本研究で提案する手法では、システム内の脆弱性を走査し、脆弱性への対応順位を決定するために、深刻度の高い順に脆弱性を表示する。提案するツールの要件を以下に示す。

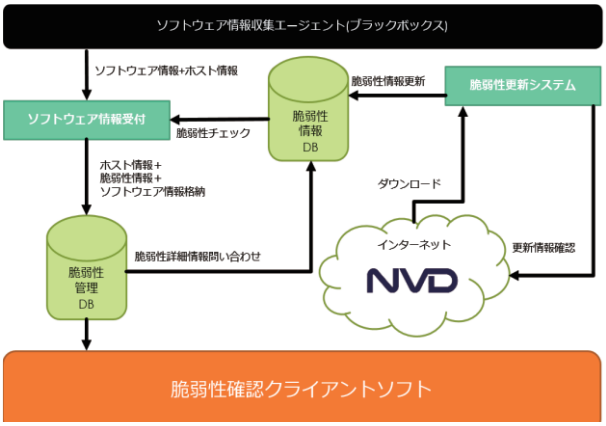

図 1 提案手法のブロック図

- 3.2 節で説明するソフトウェア情報を基に脆弱性を特定できる
- 特定された脆弱性に対応する CVSS の評価値が得られる
- 評価値をもとに、深刻度が高い順に脆弱性のあるソフトウェアを示すことができる
- 管理するコンピュータが多い場合、サブネットごとに表示できる

3.2.　ツール概要

本研究で提案するツールの動作概要を図 1 に示す。管理するシステム内のコンピュータに対し、ソフトウェアの情報を収集するエージェントを配置する。エージェントはソフトウェア情報を収集後、本システムのサーバに対し情報を送信する。エージェントから送信される情報は以下の通りとする。

- ホスト名
- IPv4 アドレス/プリフイックス長
- ソフトウェア情報

ここでソフトウェア情報とは、ベンダー名、プロダクト名、バージョンのセットを指す。

エージェントからデータを受け取ったサーバは、ソフトウェア情報をもとに、当該ソフトウェアの脆弱性の有無について脆弱性情報 DB に問い合わせを行う。脆弱性を有していた場合、該当する脆弱性の CVE 番号とソフトウェア情報、ホスト名、IP アドレスを脆弱性管理 DB に格納する。ソフトウェアに脆弱性が存在しなかった場合、CVE 番号以外を脆弱性管理 DB に格納する。これは、将来脆弱性が見つかった場合の検知漏れを防ぐためである。

提案システムは定期的に、NVD のサーバに脆弱性情報の更新を確認し、更新があった場合は脆弱性情報 DB を更新する。

表 5 検証用の脆弱なソフトウェア

ホスト名	ベンダー名	プロダクト名	バージョン
serverPC	isc	bind	9.9.9
Business PC1	Microsoft	visio	2016
		skype	7.36
Business PC2	adobe	flash_player	27.0.0.18

表 6 ホスト一覧

ホスト名	IP アドレス
serverPC	192.168.0.250/24
businessPC1	192.168.11.10/24
businessPC2	192.168.11.15/24

脆弱性管理 DB に格納したデータは、ツールの一部であるクライアントソフトによって確認できる。クライアントソフトは、脆弱性管理 DB に対し問い合わせを行い、格納されたデータを取得する。脆弱性管理 DB は問い合わせ時に、脆弱性情報 DB に対し CVE 番号を基に CVSS の基本値を問い合わせ、基本値に基づき対応順位を決定、表示する。

3.3. 利用する脆弱性データ

NVD の脆弱性情報は、JSON 形式、または XML 形式で提供されている。本研究では、JSON 形式のファイルを利用する。具体的には、NVD の脆弱性情報から以下を取得する。

- CVE 番号
- ベンダー名 プロダクト名 バージョン情報
- 概要
- リファレンス群
- CVSSv2 基本値
- CVSSv3 基本値

3.4. テストデータ

実装したツールの動作の検証にあたり、表 5、及び表 6 に示す環境を用意した。表 5 のソフトウェアは、すべて脆弱性を含んでいる。

3.5. 検証と評価

図 2 にクライアントの動作の様子を示す。図 2 より、準備した 4 つのソフトウェアに対し、CVSS の基準値をもとに色分けした警告、およびそれらの警告が深刻度順に表示できていることがわかる。また、図 2 の左側に、サブネットまたはホストを選択する欄を設けることで、

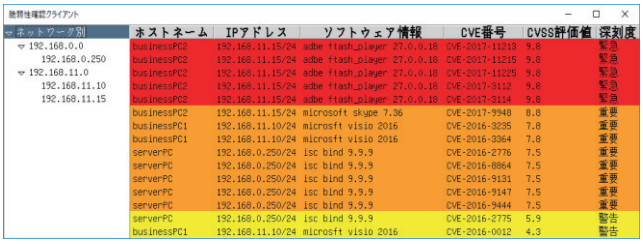

図 2 クライアント動作の様子

それらにかかる脆弱性を抽出できている。これにより、大規模なコンピュータシステムを管理する上で、脆弱性が潜在するホストの特定が容易になる。

4. おわりに

本研究では、システム内に潜在する脆弱性を検知し、脆弱性への対応順位を決定するために深刻度の高い順に表示するツールの開発を目的とした。公開された脆弱性情報を用いて脆弱性の検知を行い、CVSS 評価値により深刻度順に脆弱性を抱えるソフトウェアを表示するツールを実装し、動作を確認した。

今後の課題として、調査エージェントの開発、脆弱性の詳しい情報の表示などが挙げられる。

参考文献

1) 総務省 | 平成 29 年版情報通信白書 |http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h29/pdf/index.html .
2) 中村逸一, 兵藤敏之, 曽我正和, 水野忠則, 西垣正勝. “セキュリティ対策選定の実用的な一手法の提案とその評価”, 情報処理学会論文誌, Vol. 45, No. 8,pp. 2022–2033, 2004.
3) 諏訪博彦, 原賢, 関良明、“情報セキュリティ行動モデルの構築―人はなぜセキュリティ行動をしないのか.”, 情報処理学会論文誌, Vol. 53, No. 9, pp.2204–2212, 2012.
4) Cve. https://cve.mitre.org/ .
5) Mitre. https://www.mitre.org/ .
6) Nvd. https://nvd.nist.gov/ .
7) Nist. https://www.nist.gov/ .
8) Cvss. https://www.first.org/cvss/ .