2E-01

バイナリファイル解析によるアプリ脆弱性検知システムについての考察

藤松　由里恵†　　花谷　嘉一†　春木　洋美†

株式会社東芝†

1. はじめに

近年、サイバー攻撃の要因となりうるソフトウェアの脆弱性の報告が増加している。一般的に、あるソフトウェアの脆弱性を用いる新たな攻撃手法が発見されると、同様の脆弱性を含んだ別のソフトウェアも同じ攻撃を受ける恐れが高い。サイバー攻撃に迅速に対応するためには、脆弱性が報告されたソフトウェアのみを対象に逐次対策を行うのみではなく、他のソフトウェアに対しても同様の脆弱性を含んでいるかを検査して対処することが必要である。

本論文では、いくつかのソフトウェアに対する既知の脆弱性の共通項目を見つけることで、他のソフトウェアが同様の脆弱性を有するかを検知するフレームワークを検討する。

2. 関連技術・関連研究と課題

2-1. 脆弱性検知手法

脆弱性検知を行うツールやサービスが広く提供されおり、手法の種類は様々なものが知られている[1]。本論文では、検知対象のソフトウェアの脆弱性と攻撃手法の種類で既存手法を分類した（表１）。

脆弱性を持つことが既知のソフトウェアの検知ツールとして Nessus[2]などがある。また、最近のソフトウェアは Open-source Software (OSS) を一部に利用するものも多い。ソフトウェアが脆弱性を持った OSS を利用していることを検知するツールとして Black Duck Hub[3]が知られている。脆弱性と攻撃手法が共に未知の場合はファジングテスト等の検知手法が知られている。攻撃手法は既知だが、ソフトウェアにその攻撃手法による脆弱性を含むかが未知の場合、一般的に、専門家によるヒューリスティックな解析や開発フェーズにおけるソースコード解析による検知が行われる。しかし、近年のソフトウェア開発ではサードパーティ製品の利用や外部委託により自社開発部分の減少が進んでいる。こうした環境ではプログラムバイナリのみが手元にあるような場合もあり、バイナリ解析による脆弱性検知技術も必要となる。

表１：脆弱性検知手法

ソフトウェアの脆弱性	攻撃手法	検知手法
既知	既知	既知脆弱性検査（Nessus等）
一部既知（OSSを含むソフト）	既知	OSSチェックツール(BlackDuck等)
未知	既知	今回のターゲット
未知	未知	未知脆弱性検査（ファジング）

2-2. バイナリ解析による脆弱性検知技術と課題

プログラムバイナリ解析の関連研究として、中島らの研究[4]や Yaniv らの研究[5]がある。これらの関連研究では、既知脆弱性を含むバイナリコードと検知対象のバイナリコードに対してそれぞれ正規化を行い、それらを比較して類似度やハッシュの一致を検査することで、検知対象が同一の脆弱性を含むか否かを判定する。しかし、関連研究技術を組み込んだ脆弱性検知システムを実運用するためには、いかに脆弱性を含むバイナリコードを的確かつ迅速に収集するかが重要な課題となる。ある攻撃手法が明らかになると、次々にその攻撃手法を用いて他のソフトウェアが攻撃される傾向にある。こうした攻撃トレンドを把握し、優先的に対策を行わなければならない。

3. 脆弱性検知システム概要

本論文では、攻撃手法は既に判明しているが、検知対象のソフトウェアがその攻撃手法による脆弱性を含むかが未知の場合の検知フレームワークを提案する（図 2）。提案手法では、攻撃のトレンドを考慮した優先度付けを行い、悪用される可能性のある部分（以下、脆弱性コード）を抽出する。

A study of the vulnerability detection system for the Application with the binary file analysis,
†Yurie FUJIMATSU, Yoshikazu HANATANI, Hiroyoshi HARUKI
† Toshiba Corporation

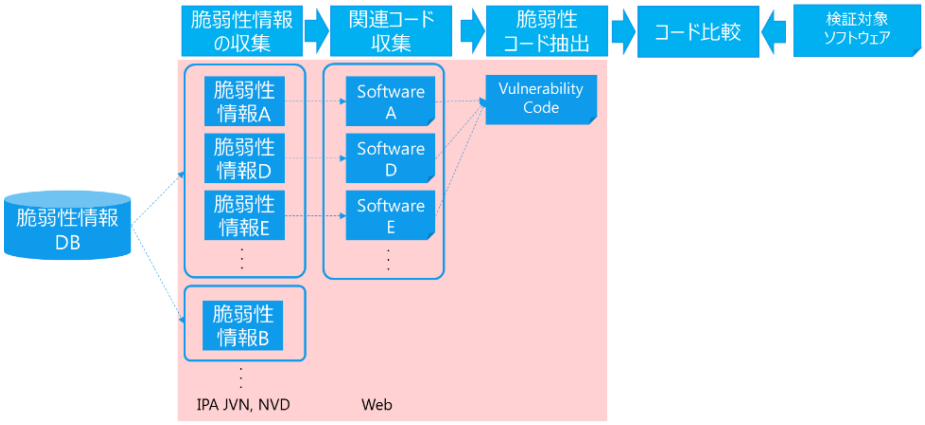

図 2：脆弱性検知システム処理フロー

抽出した脆弱性コードと検知対象のソフトウェアのコードを比較することで、検知対象のソフトウェアが脆弱性を含むかどうかを判定する。コード比較に関しては既存技術を利用する。

3-1. 攻撃トレンドの決定

はじめに、脆弱性情報を用いて攻撃トレンドを決定する。日本国内では JPCERT/CC と IPA により、Japan Vulnerability Notes (JVN) [6]が公開されている。JVN をはじめとした脆弱性情報を一定期間の範囲で収集し攻撃手法毎にグルーピングを行い、報告事例の多い攻撃手法を高優先と判定する。

JVN の ID はソフトウェア毎に割り振られており、攻撃手法毎の分類をそのまま行うことはできない。共通脆弱性タイプ一覧 Common Weakness Enumeration (CWE) も広範囲での識別であり、新規攻撃手法に対してはその他や分類するための情報不足とされるケースも多い。そのため、攻撃手法毎のグルーピングは JVN 内に記載されているタイトルや概要等の情報を分析し、攻撃手法を特定することで実現する。例えば、IPA のレポート[7]内で 2017 年第 2 四半期に急激に登録件数が増えた DLL 読み込みに関する脆弱性は、JVN のタイトルや概要内の頻出ワードをテキスト解析することで攻撃トレンドとして決定することが可能である。

3-2. 脆弱性コード抽出方法

攻撃トレンドの決定後、ウェブ上から該当するソフトウェアのプログラムバイナリを収集し、脆弱性コードを抽出する。JVN 等の脆弱性情報ではソフトウェア名の公開のみのため、手動もしくは自動でプログラムバイナリを収集しなければならない。また、ソフトウェアは複数のライブラリから構成されているため、脆弱性コード以外の要素を多く含む場合も想定される。そのため、収集したプログラムバイナリに対して、共通部分を抽出することで、脆弱性コードを限定する。

4. まとめと今後の課題

既知の攻撃手法による脆弱性が対象のソフトウェアに含まれるかどうかを検知するシステムについて提案を行った。一定期間内の脆弱性情報を分析し、攻撃を受ける可能性が高いトレンドを考慮した攻撃手法を抽出する。これにより、ソフトウェア提供側は複数のソフトウェアが同様の脆弱性を含むかを容易に検知することが可能となる。

今後は、検知対象との比較を行うために十分な精度の脆弱性コードを抽出できるかの調査を行っていく。

参考文献

[1] IPA, 脆弱性検査と脆弱性対策に関するレポート,2013
[2] nessus, https://www.tenable.com/products/nessus/nessus-professional
[3] BlackDuck Hub, https://www.blackducksoftware.com//products/hub
[4]中島明日香, 岩村誠, 矢田健, 機械語命令の類似度算出による複製された脆弱性の発見手法の提案,CSS2015
[5]Yaniv David,Nimrod Partush,Eran Yahav,Similarity of Binaries through re-Optimization,PLDI2017
[6] JVN, http://jvn.jp/
[7] IPA JPCERT, 脆弱性対策情報データベース JVN iPedia に関する活動報告レポート[2017 年第 2 四半期（4 月～6 月）],2017/07