5F-04

ソフトウェア脆弱性の自動テストツール*

赤坂航平† 中村章人†

会津大学†

1 はじめに

ソフトウェアの脆弱性はサイバー攻撃の要因であり、これを適切に把握して解消することは重要なセキュリティ対策の一つである。本論文では、ある脆弱なソフトウェアの振る舞いや実行結果を確認しようとしたときに、その脆弱性識別子といくつかのパラメータを与えるだけで、テスト環境の構築、テストの実施、レポートの作成までを自動で行うツールの設計と実装について述べる。

ソフトウェア開発者は、自ら開発するソフトウェアの動作確認やバグ修正にこのツールを利用できる。システム管理者は、運用システムに影響を与えないように、レプリカを容易に作成してテストを実施できる。また、教育や技術競技等の目的でも本ツールを活用できる。

2 脆弱性テスト環境の類型

2.1 ソフトウェア動作環境の類型

まず､脆弱性のテスト環境を構築するにあたり、ソフトウェアの動作環境の類型を表 1 に示す。各レベルは環境構築に必要な要素が異なり、基本的に一つ下のレベルの要素も必要とする。

表 1: ソフトウェア動作環境の類型

類型	環境構築の要素
レベル 1	OS のインストール
レベル 2	脆弱なソフトウェアのインストール
レベル 3	環境の設定（chmod, configure 等）
レベル 4	アプリケーションコンテンツの追加（データベースレコード、HTML ファイル、プログラム等）
レベル 5	分散環境の構築（ネットワークの設定等）

レベル 1 は OS に脆弱性がある場合、レベル 2 は OS 以外のソフトウェアに脆弱性がある場合で、両者ともソフトウェアのみで脆弱性が発現する。レベル 3 はソフトウェアのビルド方法やファイル等に特別な設定が必要な場合である。レベル 4 はアプリケーション固有のコンテンツを必要とする場合である。レベル 5 は複数のホスト上のアプリケーションが連携して動作する場合である。

2.2 攻撃環境の類型

次にテストの実施に関して、攻撃環境の類型を表 2 に示す。ローカル型では、脆弱なソフトウェアの動作環境を構築したホスト上に攻撃ツールとそのモジュール、またはエクスプロイトコードを配備する。リモート型では、攻撃用のホストを別途構築してこれらを配備する。

表 2: 攻撃環境の類型

類型	攻撃対象ホストと攻撃コードとの関係
ローカル	攻撃対象の同一ホスト内からのみ攻撃可能
リモート	攻撃対象とは別のホストから攻撃可能

3 機能と実装

脆弱性の自動テストを行うために必要な機能と、ツールの実装について述べる。

3.1 機能

脆弱性テストには以下の機能が必要である。

- 脆弱なソフトウェアの動作環境の構築: ソフトウェア動作環境の類型に基づいて必要な要素を実行し、脆弱性の発現を確認できる環境（攻撃対象ホスト）を構築する。
- 攻撃環境の構築: 攻撃環境の類型に基づいて、攻撃用ホストの構築や攻撃コードの配備を行う。
- 攻撃の実行: 攻撃コードを実行して、脆弱性の発現を確認する。
- レポートの作成: 脆弱性およびテスト環境に関する情報をまとめて出力する。

* “Automation tool for software vulnerability tests”, Kohei AKASAKA, Akihito NAKAMURA
† University of Aizu

3.2 システム構成

我々は前述の機能を自動実行するツールを開発している。コマンドラインインタフェースにより起動し、環境構築、攻撃、レポート出力までのプロセス全体を実行する。そのシステム構成を図 1 に示す。主な要素は以下の通りである。

- メタデータ: 脆弱性の基本情報(CVE 識別子、CVSS スコア、概要等）[2]、ソフトウェア動作環境に関する情報、攻撃コードに関する情報から構成される。
- 仮想化・構成管理ツール: テスト環境の構築に用いる。攻撃対象および攻撃用のホストは仮想マシンとして作成する。Virtualbox[6]、Vagrant[5]、Ansible[4]を利用する。
- 攻撃コード: Metasploit[2]とそのモジュール、またはスクラッチのエクスプロイトコードを用いる。

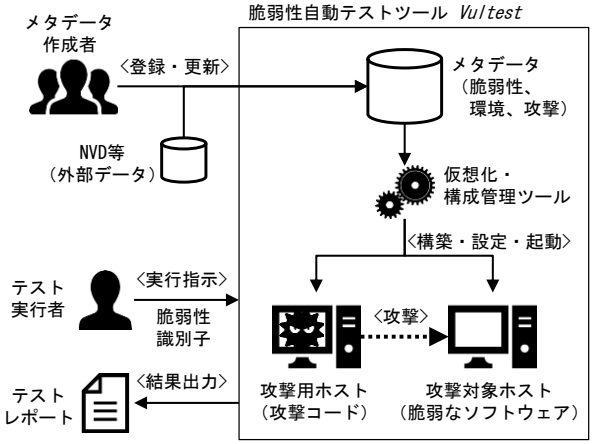

図 1: システム構成

3.3 テスト処理手順

脆弱性テストの処理手順の概略を示す。

(1) テスト実行者からテストしたい脆弱性の識別子（CVE）が指示される。
(2) CVE をキーにしてメタデータを取得する。
(3) メタデータに基づいて仮想化・構成管理ツールを駆動し、脆弱なソフトウェアの動作環境（攻撃対象ホスト）を仮想マシンとして構築する。
(4) 攻撃環境がリモートの場合、攻撃用ホストを起動して攻撃コードを配備する。
(5) 攻撃コードを実行し、攻撃の成否を判定する。
(6) テストレポートを作成し出力する。

3.4 実装

本ツールは、今のところ Linux 系の OS のみを対象としている。レベル 4 までの動作環境は実現しているが、レベル 5 は開発途上である。ツールの主機能は Ruby 言語で実装しており、約 1,000 行である。

これまでに作成したメタデータでテストできる脆弱性を表 3 に示す。意図したとおり正しく攻撃を実行できるオープンな攻撃コード、すなわち Metasploit モジュールまたはエクスプロイトコードの入手可能性が低いためにこの程度の数に留まっている。

表 3: 動作確認した脆弱性

CVE	脆弱なソフトウェア	テスト環境の類型
CVE-2014-6271	Bash	4/リモート
CVE-2015-1318	Apport	1/ローカル
CVE-2015-1328	Linux kernel	2/ローカル
CVE-2015-3224	Ruby on Rails	4/リモート
CVE-2016-0752	Ruby on Rails	4/リモート
CVE-2016-4557	Linux kernel	2/ローカル
CVE-2017-7308	Linux kernel	3/ローカル
CVE-2017-11467	OrientDB	2/リモート
CVE-2017-16995	Linux kernel	2/ローカル

4 おわりに

ICTセキュリティにおけるソフトウェアの脆弱性に着目し、脆弱なソフトウェアの動作環境の構築、攻撃、レポートの作成までを自動で行うツールの設計と実装について述べた。仮想化および構成管理の技術を駆使して自動化を可能にした。Beuranらのシステム [1]よりも汎用性が高いものと考える。

脆弱性毎に環境構築および攻撃に必要な情報を収集してメタデータを作成する必要がある。これらのメタデータと攻撃コードは、オープンソースの作法で集積と共有を推進していくつもりである。

参考文献

[1] R. Beuran, et al.: “Cybersecurity Education and Training Support System: CyRIS”, *IEICE Trans. on Information and Systems*, Vol. E101-D, No. 3, 2018, pp.740-749.
[2] NVD: https://nvd.nist.gov/
[3] Metasploit: https://www.metasploit.com/
[4] Ansible: https://www.ansible.com/
[5] Vagrant: https://www.vagrantup.com/
[6] Virtualbox: https://www.virtualbox.org/