L-009

IoT 機器のファームウェアにおけるハードウェアバージョンを考慮したソフトウェアの差分解析

Software diffing on IoT device firmware considering hardware version

兼松 智也 † 山内 利宏 ††
Tomoya Kanematsu Toshihiro Yamauchi

1. はじめに

ソフトウェアの差分解析は，パッチ解析やセキュリティ調査で用いられる手法である．先行研究 [1] では，IoT 機器のファームウェアを対象に，アップデートにより，バージョンが変化せずハッシュ値が変化した実行ファイルに対し，差分解析を実施した．しかし，先行研究 [1] では，ハードウェアバージョンを考慮せずに公開日時の時系列順に連続するファームウェアのペア（以降，ファームウェアペア）を選択しており，アップデート前後の関係にあるファームウェアペアを正しく選択できていない．

本稿では，ハードウェアバージョンを考慮してファームウェアペアを正しく選択し，先行研究 [1] の差分解析を正しく実施した結果を報告する．

2. ソフトウェアの差分解析における手法と課題

2.1 先行研究におけるソフトウェアの差分解析の手法

先行研究 [1] では，公開日時の時系列順に連続する同一製品のファームウェアペアについて，抽出した実行ファイルのペアを差分解析し，暗黙的パッチの適用の実態を明らかにする．暗黙的パッチとは，バージョンを変更せずに加えられた脆弱性やバグの修正である．

差分解析では，以下の 2 つの観点から暗黙的パッチを識別する．

プログラム構造の差分：あらかじめ決定した閾値以上のプログラム構造の類似度を持つものを暗黙的パッチと判定する．類似度は BinDiff [3] が算出する Similarity を用いており，閾値 0.885 以上の Similarity の場合を暗黙的パッチと判定する．

文字列の差分：プログラム構造に差分がなく，オブジェクト，もしくはライブラリ名を表す文字列に変更があるものを暗黙的パッチと判定する．IDA Pro [4] を使用して実行ファイルから抽出した文字列に対し，`diff` コマンドを使用して文字列の差分を解析する．

2.2 課題

先行研究 [1] のファームウェアペアの選択方法（以降，従来手法）では，ハードウェアバージョンを考慮せずに公開日時の時系列順にファームウェアペアを選択してしまうという課題がある．ハードウェアバージョンとは，製品の年式のようなものであり，部品調達の都合などで同じ製品でもハードウェアバージョンが変更されることがある [2]．特定のベンダでは，異なるハードウェアバージョンをもつ同一製品を販売しており，異なるハードウェアバージョンに対するファームウェア同士で互換性がないことがある．このような場合，異なるハードウェアバージョンに対するファームウェア同士は差分解析の対象として適切ではない．

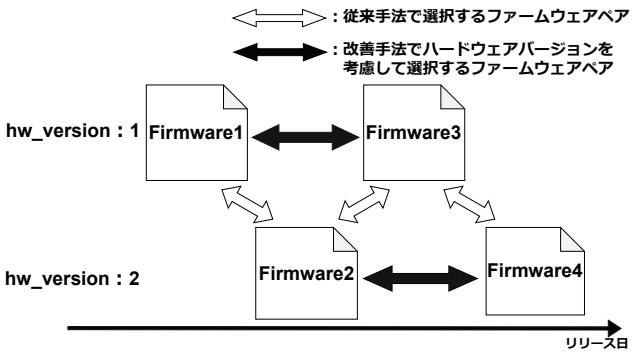

図 1 実行ファイル差分解析のためのファームウェアペアの選択方法

3. 先行研究の課題への対処方法

図 1 に従来手法と，正しくファームウェアペアを選択するためにハードウェアバージョンを考慮したファームウェアペアの選択方法（以降，改善手法）を示す．

改善手法では，以下の手順でファームウェアペアを選択する．

(1) ファームウェアをハードウェアバージョンの値の大小に基づき並び替える
(2) 同一のハードウェアバージョンを持つファームウェアの中で，公開日時の時系列順に並び替える
(3) 同一のハードウェアバージョンを持ち，隣り合ったファームウェアペアを選択する

4. ハードウェアバージョンの考慮によるソフトウェアの差分解析への影響の評価

4.1 評価目的

評価では，従来手法と改善手法でファームウェアペアを選択した際に，解析する実行ファイルのペアにどの程度違いが生じるかを明らかにする．また，改善手法で暗黙的パッチを識別した結果から，暗黙的パッチの適用の実態を明らかにする．

4.2 評価対象

評価では，先行研究 [1] と同じファームウェア 591 個を使用した．これらのファームウェアには，11 社のベンダ製ファームウェアが含まれていた．このうち，ハードウェアバージョンの考慮が必要であるベンダが 2 社存在し，使用したファームウェアの 119 個を占めていた．

また，評価において，先行研究 [1] における調査対象アプリケーション 37 種類の内，**表 1** に示す 23 種類を本研

† 岡山大学大学院環境生命自然科学研究科, Okayama University
†† 岡山大学学術研究院環境生命自然科学学域, Okayama University

表 1　調査対象アプリケーション

busybox, lighttpd, thttpd, ebtables, iptables, openvpn, dnsmasq, hostapd, iw, wpa_cli, wpa_supplicant, dropbear, openssh, stunnel, pppd, ethtool, conntrack, curl, iperf, lua, ntfs_3g, radvd, zebra

表 2　従来手法による実行ファイルのペア選択結果

	プログラム構造の差分		文字列の差分	
	正	誤	正	誤
patch	85	12	1	0
other	5	24	93	15
合計	90	36	94	15

究の調査対象アプリケーションとした．これらは，IoT 機器で広く利用されており，脆弱性が報告されているアプリケーションである．

4.3 評価方法

まず，従来手法と改善手法でファームウェアペアを選択し，暗黙的パッチを識別した．次に，従来手法と改善手法によって選択された実行ファイルのペアを比較し，それぞれの手法において解析した実行ファイルのペアの違いを明らかにした．加えて，プログラム構造の差分に対する暗黙的パッチの識別結果を累積分布関数に示し，比較した．

暗黙的パッチの識別では，2.1 節で説明した基準に従って，アップデートにより実行ファイルのハッシュ値が変化した要因を暗黙的パッチ (patch) かこれ以外 (other) かに分類した．

4.4 評価結果

従来手法でファームウェアペアを選択した場合，誤った実行ファイルのペアがどの程度生じるか評価した結果について，**表 2** に示す．従来手法では，プログラム構造に差分があった実行ファイルのペア 126 個中 36 個（約 28.6%），文字列に差分のあった実行ファイルのペア 109 個中 15 個（約 13.8%）に対し，間違えて解析していることが分かった．また，改善手法において，新たに選択した実行ファイルのペアが 5 つ存在した．

この結果から，ファームウェアを対象としたソフトウェアの差分解析において，ハードウェアバージョンを考慮しなければ，解析結果と解釈が間違ったものになる可能性があることが分かる．

次に，従来手法と改善手法でファームウェアペアを選択し，プログラム構造の差分に対する暗黙的パッチを識別した結果を**図 2** と**図 3** に示す．従来手法では，プログラム構造に差分があった実行ファイルのペア 126 個中 97 個（約 77.0%）が暗黙的パッチと識別されたのに対し，改善手法では，91 個中 86 個（約 94.5%）が暗黙的パッチと識別された．改善手法によって，暗黙的パッチと識別されたものの割合がより多いことが分かる．

改善手法による暗黙的パッチの識別結果から，IoT 機器で使用されているアプリケーションにおいて，暗黙的パッチが多数適用されていると推測できる．また，このことから，ファームウェアに対する脆弱性評価がバージョン番号に依存していると，信頼性が低下する可能性がある．これは，暗黙的パッチはバージョンに基づいた脆弱性検知において誤検知を引き起こす可能性があるためである．

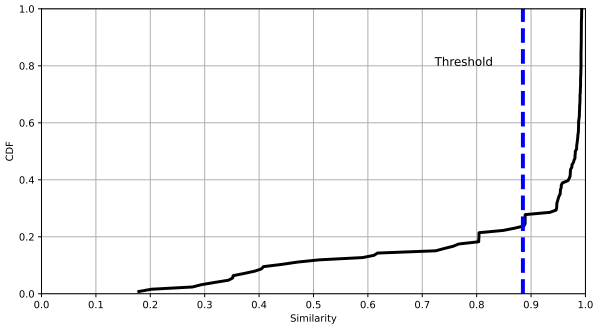

図 2　従来手法における Similarity の累積分布関数

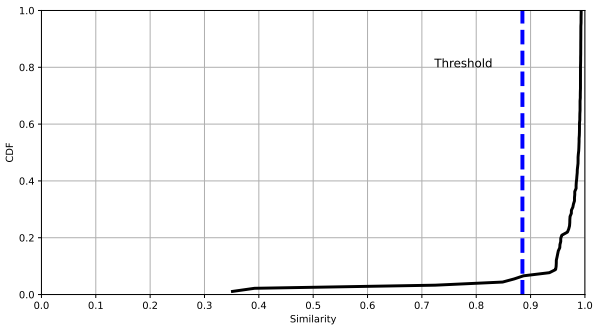

図 3　改善手法における Similarity の累積分布関数

5. おわりに

本稿では，ハードウェアバージョンを考慮することによって正しくファームウェアペアを選択し，先行研究 [1] の差分解析を実施した．この結果，ファームウェアを対象としたソフトウェアの差分解析において，ハードウェアバージョンを考慮しなければ，解析結果と解釈が間違ったものになる可能性があることが分かった．また，IoT 機器で使用されているアプリケーションにおいて，暗黙的パッチが多数適用されていると推測できた．

今後の課題として，類似度による推測でなく，詳細な分析に基づいた暗黙的パッチの識別手法の確立がある．

謝辞　本研究の一部は，JST【経済安全保障重要技術育成プログラム】【JPMJKP24K2】の支援を受けたものです．

参考文献

[1] M. Akiyama, S. Shiraishi, A. Fukumoto, R. Yoshimoto, E. Shioji, and T.Yamauchi：Seeing is not always believing: Insights on IoT manufacturing from firmware composition analysis and vendor survey, *Computers & Security,* vol. 133, p. 103389, 2023.

[2] TP-Link: ハードウェアバージョンとは？どこで確認できますか？, 入手先 〈https://www.tp-link.com/jp/support/faq/46/〉 (参照 2025-6-11).

[3] Google: BinDiff, 入手先〈https://github.com/google/bindiff〉 (参照 2025-6-12).

[4] Hex rays: IDA Pro, 入手先〈https://hex-rays.com/ida-pro/〉 (参照 2025-6-12).