IPA

2020 年度 未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当 PM

田中 邦裕（さくらインターネット株式会社　代表取締役社長）

2. クリエータ氏名

杉山 優一（東京大学大学院情報理工学研究科）

3. 委託金支払額

2,736,000 円

4. テーマ名

ハードウェアセキュリティ検査システムの開発

5. 関連 Web サイト

検出したバグ：

- https://github.com/lowRISC/ibex/issues/1277
- https://github.com/lowRISC/ibex/issues/1282
- https://github.com/rsd-devel/rsd/issues/37
- https://github.com/rsd-devel/rsd/issues/38
- https://github.com/rsd-devel/rsd/issues/39

6. テーマ概要

本プロジェクトでは最新のファジング技術を用いて RISC-V プロセッサのバグ・脆弱性を発見する、RISC-V プロセッサ用のファザーを提案・実装した。本プロジェクトの特徴は実行時の情報を用いて各プロセッサのマイクロアーキテクチャに合わせたテスト入力を作成し、既存のツールに比べて効率よく RISC-V プロセッサのバグ・脆弱性を発見できることである。本プロジェクトの成果では RISC-V プロセッサのバグ・脆弱性を効率よく見つけることが可能となった。実際に開発したファザーによって、オープンソースで開発されている RISC-V プロセッサから複数のバグ・脆弱性を発見し、開発者に報告することを実現した。

7. 採択理由

社会のコンピュータへの依存度が高まる中で、セキュリティは非常に重要な分野であるが、最近では Spectre や Meltdown などのマイクロアーキテクチャに対する脆弱性が話題となり、ハードウェアのセキュリティに注目が集まっている。

そのような中で、ファジングを活用して演算器などのバグや脆弱性を発見し、サイドチャネル攻撃などへの耐性を検査するという提案は未踏性もあり、かつ完成した時の社会への有益性も高く、そして実現可能性もあると考え採択した。

8. 開発目標

本プロジェクトの目標は、「RISC-V プロセッサのバグ・脆弱性を効率的に発見」でき、「RISC-V プロセッサのためのテストケースを自動で生成」できるツールの開発を目指すことである。

RISC-V プロセッサのバグ・脆弱性を効率よく発見できることはプロセッサのセキュリティを高めることができるため重要であり、テストケースを自動で生成できることはプロセッサ開発のコストを削減できるため重要である。

その目標を達するために、本プロジェクトでは RISC-V 向けのファザーの開発を行った。ファザーは大別して以下の 3 つの操作を繰り返し行う（図 1）。

(1) 命令列を生成する
(2) その命令列を入力として、ターゲットのプロセッサを実行する
(3) 実行時の情報をフィードバックする

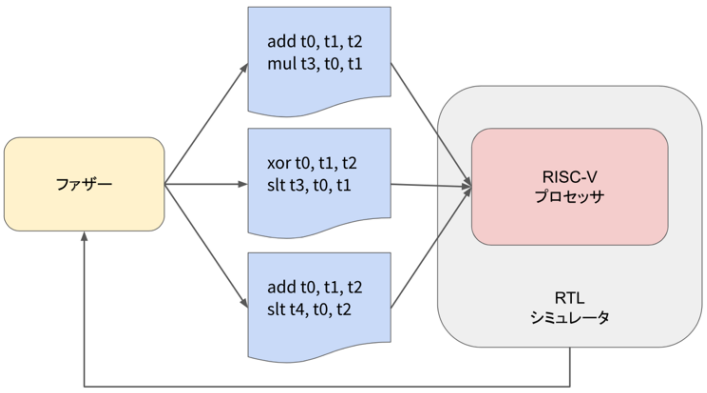

図 1：RISC-V 向けのファザー

この一連の操作を繰り返すなかで、プロセッサのバグ・脆弱性が発生する命令列の作成や有用なテストケースの作成を目指す。

既存のプロセッサ検査手法は、ハードウェアに関する特殊な専門知識やハードウェアデザインを別の言語で書き直す必要があり、手動で行わなければならない作業が多く、とても大変な作業である。また、大規模なハードウェアデザインに対して効率よく検査を行うことが難しいといった課題がある。そこで、一般的に簡単に実行でき、スケールしやすいファザーを使うことで、既存ツールの問題点を解決できる。

9. 進捗概要

本プロダクトは MicroFuzz と名付けた。MicroFuzz の特徴は 2 つあり、

(1) 既存のツールでは発見できないバグ・脆弱性を多く発見でき、
(2) 手動で作ることが難しいテストケースを自動で生成できる

点である。(1)で示した既存のツールでは発見できないバグ・脆弱性を多く発見できることを示すために、評価実験を行った。その結果、MicroFuzz が既存のツールに比べて、意図的にバグを挿入したプロセッサから多くのバグ・脆弱性を発見できることを示した。

オープンソースで開発されている RISC-V プロセッサに対してテストでは、既存ツールでは発見できていない未知のバグ・脆弱性を発見し、プロセッサ開発者にバグ・脆弱性を報告することができた。

また、今まではプロセッサのマイクロアーキテクチャやアセンブリ命令を熟知している人間が時間をかけて作成していたプロセッサのテストケースを、自動で生成できることを示した。

なお、当初の実施計画との相違点は 3 点ある。

1 点目は、バグ・脆弱性を手動で意図的に挿入したプロセッサを用いて評価を行うために、意図的にバグを挿入したプロセッサの開発を行った点、2 点目は意図的にバグを挿入したプロセッサに対する評価を行い、既存ツールに比べてどのような有用性があるかを評価した点、3 点目は情報フロー解析を実装しなかった点である。

結果として、高速な動作と、的確なバグの発見を両立できたものと考えている。

10. プロジェクト評価

当初想定されていた結果は十分に達成できたと言える。

特に、実際にバグを見つけ出して報告を行った点については、実際に既存のテストツールを用いて既にテストをパスした RISC-V プロセッサからも、MicroFuzz によって未発見のバグ・脆弱性を発見することができたなど、明確な成果が出ている。

成果報告会においては、比較的難易度の高い本プロジェクトの内容を丁寧に解説し、プロジェクトで達成したことについて、的確なプレゼンテーションを行えたものと考えている。

なお、成果報告会後もソフトウェアのアップデートに取り組み、実際に新たなバグを見つけ出すとともに、それらを成果報告書に反映したことなど、プロジェクトの完成度を高める努力については大変評価できる。

11. 今後の課題

今後の課題は大きく分けて４つあり、

(1) MicroFuzz を論文として公開
(2) 手動でバグを挿入したプロセッサをオープンソースとして公開
(3) 発見したバグ・脆弱性を自動で分類・分析
(4) MicroFuzz の高速化

することである。

プロジェクトの成果は学術的にも新規性があるため、MicroFuzz を論文として公開するほか、(2)～(4)についても継続的に取り組みを行う予定である。