Ubuntuパッケージ内のベクトル命令の使用傾向調査によるDownfall攻撃への影響推定

原田 陽平[1]　穐山 空道[1,a)]

概要：本研究は、Ubuntu パッケージのバイナリにおけるベクトル命令の使用状況を調査することを目的としている。ベクトル命令は複数のデータを同時に処理することで高速なデータ処理を可能にする一方、Downfall 攻撃と呼ばれるベクトル命令使用時に特有な攻撃も発見されておりセキュリティ上の懸念がある。Ubuntu の LTS (Long Term Support) である 5 つのバージョンのパッケージを対象にベクトル命令の使用状況を調査した結果、全パッケージ中の 75% 以上にベクトル命令が含まれていること、またベクトル命令の利用はバージョンが新しくなるにつれ増加していることが分かった。

1. はじめに

ベクトル命令（SIMD 命令）はパフォーマンスの向上に必要不可欠である。特に、AVX (Advanced Vector Extensions) や SSE (Streaming SIMD Extensions) といった命令セットは計算負荷の高いアプリケーションなどで多くのデータを同時に処理できるため、パフォーマンス向上に貢献している。これにより、より高速で複雑な処理が可能となり、ソフトウェアの最適化が進んでいる。

AVX や SSE などの命令により、大規模なデータを同時に処理できる一方で、その効率化はセキュリティリスクを伴う可能性がある。Downfall 攻撃 [1] はベクトル命令を含むプログラムに対し、そのプログラムが利用中のベクトルレジスタの値の読み書きを実現する。したがって、ベクトルレシスタの使用状況を理解することは、パフォーマンス向上とともに安全なシステム設計のために重要である。

そこで本研究では、Ubuntu パッケージのバイナリにおけるベクトル命令の使用状況を調査し、セキュリティリスクを評価する。特に、幅広い分野で採用されている Ubuntu パッケージを調査対象とする。

2. 調査

2.1 調査目的と調査対象

本調査の目的は、Ubuntu パッケージのベクトル命令の有無をバージョンごとに調査することである。バージョンごとに調査することで、最新のバージョンと過去のバージョンのベクトル命令の使用状況を比較する。

[1] 立命館大学 情報理工学部
[a)] s-akym@fc.ritsumei.ac.jp

調査対象として、Ubuntu の LTS (Long Term Support) バージョンである Ubuntu 14.04、Ubuntu 16.04、Ubuntu 18.04、Ubuntu 20.04、Ubuntu 22.04 を選定する。LTS バージョンは通常バージョンに比べ利用者が多く、攻撃の影響も大きいと考えられるためである。

2.2 調査手法

本研究では以下の手順に従いパッケージごとのベクトル命令使用状況を調査する。

(1) 特定のバージョンの Ubuntu パッケージを取得する
(2) ダウンロードしたパッケージを展開したファイル群からバイナリファイルのみを特定して抽出する。
(3) 各バイナリファイルに objdump を適用しアセンブリコードを生成する。
(4) 生成されたアセンブリコードにベクトル命令が含まれているかを調査する。ベクトル命令は種類が膨大なため、ベクトル命令を直接発見する代わりに xmm レジスタがバイナリ内で利用されているかを調査する。xmm レジスタとは SSE 命令や AVX 命令で用いる 128 bit 長のレジスタであり、xmm レジスタが使われていればベクトル命令も使われていると言える。アセンブリコードから grep コマンドを用いて xmm という文字列を検索することで xmm レジスタが使われているかどうかを判断する。なお ymm レジスタや zmm レジスタのみが使われているケースは今後調査する。

3. 調査結果

本稿ではバージョンごとのベクトル命令使用率の変移と同じパッケージ間でのベクトル命令の有無の変化を評価す

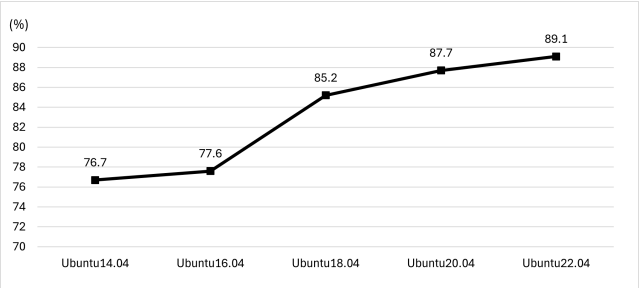

図 1: ベクトル命令が含まれているパッケージの割合

表 1: 同一パッケージのベクトル命令使用有無の変化

バージョン	無 → 有	有 → 無	共通パッケージ数
14.04 → 16.04	70	1	8812
16.04 → 18.04	621	65	8671
18.04 → 20.04	242	34	8798
20.04 → 22.04	112	54	9550

る。本調査ではパッケージ内に 1 つでもベクトル命令が含まれているバイナリがある場合、ベクトル命令を含むパッケージとする。

(1) バージョンごとのベクトル命令使用率：調査の結果を **図 1** に示す。図の横軸は、Ubuntu パッケージのバージョンであり、縦軸はベクトル命令が含まれているパッケージの割合である。バージョンが上がるにつれて増加傾向である。特に、Ubuntu 16.04 と Ubuntu 18.04 の間では他と比べ大きく変化している。

(2) 同じパッケージ間でのベクトル命令の有無：本調査は隣接するバージョン間で同じパッケージのベクトル命令の使用有無変化を調査する。ここで同じパッケージとは、バージョン番号を除いたパッケージ名が同一のパッケージペアのことである。 **表 1** は、隣接するバージョン間での同一パッケージのベクトル命令の有無の変化を示す。無 → 有はバージョンが上がる際にベクトル命令を含んでいなかった同一パッケージがベクトル命令を含むようになったパッケージ数、有 → 無はベクトル命令を含まなくなったパッケージ数である。また共通パッケージ数とは比較されている Ubuntu バージョン間で共通に存在したパッケージ数である。表より、どのバージョンもベクトル命令が追加されたパッケージ数が削除されたパッケージ数より多くなっている。また、(1) と同様に、Ubuntu 16.04 と Ubuntu 18.04 の間ではベクトル命令を含むパッケージの増加数に他と大きな差がある。

調査結果より、Ubuntu パッケージのうち 75% 以上がベクトル命令を含んでいること、またベクトル命令を含むパッケージの割合は年々増加していることが分かった。

このような結果が得られた理由の解明は今後の課題だが、一つの可能性としてコンパイラの最適化能力向上が考えられる。今回発見されたベクトル命令は例えばマニュアルを参照する info コマンドのような性能要求が緩いソフトウェアでも使われている。これらのベクトル命令はプログラマが意図して挿入したのではなくコンパイラの最適化により生成された可能性があり、これが真ならばコンパイラの最適化能力が向上して自動的にベクトル命令を生成できるケースが増えたことが今回の調査結果の要因な可能性がある。

4. 関連研究

関連研究として、2 つの Downfall への緩和策がある。1 つ目はコンパイラと OS のデータリークを軽減させるために、情報を収集する特定の命令を禁止できることを使う [1]。コンパイラでは SIMD 命令を SIMD レジスタを使用しない同等の命令に書き換えることでアプリケーションが直接データを漏らさないようにする。同様に、OS では SIMD レジスタを使用する SIMD 命令を無効にして、任意のメモリとレジスタのリークを防ぐ。また、gather 命令のあとに lfence 命令を追加することで gather がデータを一時的に転送しないようにすることで攻撃を緩和できる。2 つ目は攻撃者と被攻撃者がプログラムを同じ物理コアで実行する必要があるという Downfall の特性を利用する [2]。守りたいプログラムと同じ物理コアで、他のプログラムの実行ができないようにする。

5. 今後の課題

一点目は、バイナリだけでなくソースコードにも同様にベクトル命令の有無を確認することである。ベクトル命令が本来不要なパッケージに対し、コンパイラが自動的にベクトル命令を追加している場合は、不要な命令を削除することで、リスクを減らすことができる。このため、バイナリとソースコード間でのベクトル命令の有無を比較することで、ソースコードに手動でベクトル命令が追加されたのか、あるいはコンパイラによって自動的に挿入されたのかを確認できる。二点目は、Ubuntu に加えて Debian など他の Linux ディストリビューションにも調査対象を広げることである。ディストリビューション間でベクトル命令の使用量に違いがある場合、用途に応じたディストリビューションの選択できる。

謝辞 本研究は JST，さきがけ，JPMJPR22P1 の支援を受けたものである．

参考文献

[1] Moghimi, D.: Downfall: Exploiting Speculative Data Gathering, *32nd USENIX Security Symposium (USENIX Security 23)* (2023).

[2] 荒木悠生，高前田伸也，穐山空道：Downfall 等の攻撃に対するマイクロコードアップデートを用いない緩和手法，*Cross-Disciplinary Workshop on Computing Systems, Infrastructures, and Programming (xSIG), Poster* (2024).