VM上でのマルウェア動的解析のためのVMとホストでの取得可能ハードウェアイベント数の比較

本田 健人[1]　穐山 空道[1,a)]

概要：マルウェアの動的解析の手法として、キャッシュミスのようなハードウェアイベント数の計測が提案されている。また、マルウェアを実行する際に、ホストがマルウェアに汚染されないために VM でマルウェアを動かし、VM 上で動的解析することも提案されている。しかし VM 上のマルウェアが発生させるハードウェアイベント数をホストから計測すると、VM 内のその他のプロセスの影響などにより誤差が生じる。そこで本研究では、様々なハードウェアイベントのホストから見た発生数を、プログラムをホスト上で実行した場合と VM 上で実行した場合で比較する。調査の結果、cache-misses、cache-references、branches、L1-dcache-loads、L1-dcache-loads-misses、dTLB-loads、branch-loads、cycles、instructions では誤差が少ないが、その他のイベントでは誤差が大きく対策が必要なことが分かった。

1. はじめに

マルウェアの被害は年々大きくなっている。最近の例だと、KADOKAWA グループが大規模なランサムウェア攻撃を受けた事例がある。この攻撃で、25 万人以上の個人情報が漏洩した [1]。このような被害を出さないためにもマルウェア検知の研究を進めるべきである。

あるプログラムがマルウェアかどうか特定するにあたり、ホストにマルウェアの影響が及ぶことは避けなくてはならない。そこでマルウェアと疑われるプログラムを VM で実行することが提案されている。ホストと異なる OS の VM（例えば、ホストを Windows、VM を Linux にする）にすると、ホストに対する影響を少なくできる。

マルウェアの検知にはハードウェアレベルの痕跡が有効だと提案されている。例えば、良性のプログラムとマルウェアではキャッシュミスなどのハードウェアイベントの数に違いが出る。この方法は syscall 命令を直接呼んでシステムコールを実行することでラッパー関数のフックによる検知を回避するマルウェアにも有効である。

本研究では VM でマルウェアの動的検知を行うため、同一プログラムをホストと VM で実行した際に取得できるハードウェアイベント数を比較する。VM 上でプログラムを実行すると、ホストから見れば取得したイベントの発生源が VM 内のどのプロセスか簡単には分からない。そこで本調査では、ホスト上と VM 上で同一プログラムを実行した際に取得できるイベント数の差が少ないハードウェアイベントを「VM 上でのハードウェアイベントを使った動的解析に適している」と見なす。

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

2. 調査の目的と方法

2.1 調査目的

同一プログラムをホストと VM で実行した際に取得できるハードウェアイベント数の違いを各ハードウェアイベントごとに知ることを目的とする。また、この差が大きいハードウェアイベントに関し、イベントを発生させたプロセスの内訳を分析することで差の原因を探る。

2.2 調査方法

ホストの OS は Ubuntu 22.04 で VM の OS は Ubuntu 23.04 である。実験の際はマルウェアの代わりとして、SPEC CPU 2017 の lbm というプログラムを実行する。この以降では本プログラムを「ベンチマーク」と呼ぶ。このベンチマークは格子ボルツマン法をシミュレートしている。

ハードウェアイベントの取得には Linux の perf ツールを用いる。具体的にはホスト上でベンチマークを実行する際には perf record を、VM でベンチマークを実行する際には perf kvm をホスト上で実行する。VM でベンチマークを実行する際に VM 内で perf を実行しない理由は、(1) ハードウェアイベント数の計測は特別なレジスタを用いて行われており、このレジスタを仮想化しない限り VM 内で perf が実行できないこと、(2) 仮に VM 内で perf が実行できたとしても VM 内で見える値はマルウェアによっ

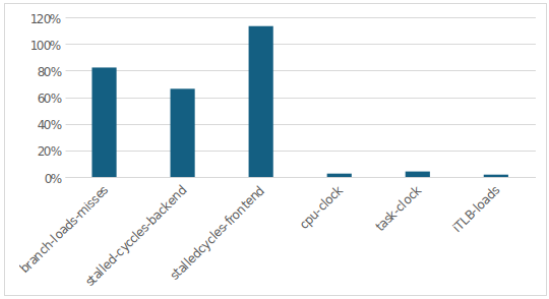

図 1: 調査 (1) の結果が 90% から 110% に収まらなかったハードウェアイベント

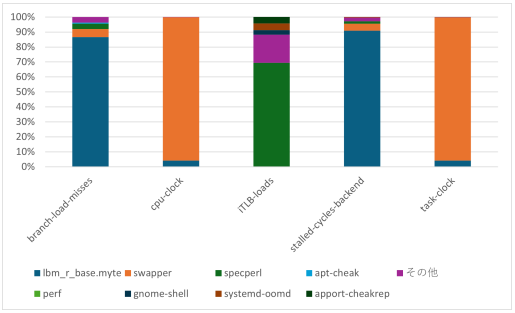

図 2: ベンチマークのプロセスの割合が 90%以下の各ハードウェアイベントのプロセスの割合

て改ざんされている可能性があること、である。

本稿で調査するハードウェアイベントを以下に示す。

- cache-misses, cache-references
- branches, branch-loads, branch-loads-misses
- L1-dcache-loads, L1-dcache-loads-misses
- dTLB-loads, iTLB-loads
- cycles, stalled-cycles-backend, stalled-cycles-frontend
- instructions
- cpu-clock, task-clock

これらのイベントに対し、以下の計測を行う。計測を行う際は結果が均一になるように計測のたびに PC を再起動する。結果は 3 回の測定の平均をとる。

- ホストのシステム全体で各ハードウェアイベントごとのイベント数を計測し、その値を H とする。
- ホストでベンチマークのプロセスのみハードウェアイベント数を計測し、その値を P とする。
- VM のシステム全体で各ハードウェアイベントごとにイベント数を計測し、その値を V とする。

計測した値を用い、以下の二点を調査する。

(1) VM のシステム全体のハードウェアイベント数に対するベンチマークのハードウェアイベント数の割合。これは $\frac{P}{V} \times 100$ で求められる。

(2) ベンチマークが発生させたハードウェアイベント数の、システム全体のイベント数に対する割合。これは $\frac{P}{H} \times 100$ で求められる。

3. 調査結果

図 1 に、調査方法 (1) の結果のうち 90% から 110% に収まらなかったハードウェアイベントを示す。特に cpu-clock, task-clock, iTLB-loads で割合が低いことが分かった。

図 2 に調査方法 (2) の結果のうちベンチマークが発生させたイベント数の割合が 90% 以下であったイベントについて、全イベント数のうちの各プロセスの割合を示す。図は各ハードウェアイベントの総数に対するプロセスのうち、割合が大きかったプロセスの上位 4 つを示す。この結果より、 iTLB-loads 以外のハードウェアイベントで swapper が上位プロセスに含まれていることが分かった。特に、task-clock と cpu-clock のハードウェアイベントの総数の swapper の割合はそれぞれ 95.81%、95.81% であった。また SPEC CPU Benchmark を制御するためのツールである specperl が全てのハードウェアイベントで上位に入っており、特に iTLB-loads は 69.44%含まれていた。

調査の結果、VM 上でマルウェアの動的解析をする際の入力として適しているハードウェアイベントは cache-misses、cache-references、branches、L1-dcache-loads、L1-dcache-loads-misses、dTLB-loads、branch-loads、cycles、instructions であることが分かった。

4. 関連研究と今後の課題

関連研究としては perf を使ったランサムウェアの早期発見がある [2]。本研究では特徴重要度アルゴリズムを使用してランサムウェアの早期発見に貢献できるハードウェアイベントを特定している。

ハードウェアパフォーマンスカウンターの値を画像化し、従来の CNN を再構築することで良性のプログラムか悪性のプログラムかを判断した研究がある [3]。この研究では、ハードウェアパフォーマンスカウンタの 100 ミリのスナップショットを使うことで、平均 96.8%の MCC スコアとほぼゼロの偽陽性率を達成した。

今後の課題は、動的解析の入力に適していないと今回の調査で結論づけたハードウェアイベントを動的解析に利用可能にするために、他のプロセスによるノイズの影響を軽減することである。

参考文献

[1] KADOKAWA: ランサムウェア攻撃による情報漏洩に関するお知らせ, `https://www.kadokawa.co.jp/topics/12088/` (2024).

[2] Anand, P. M., Charan, P. V. S. and Shukla, S. K.: HiPeR - Early Detection of a Ransomware Attack using Hardware Performance Counters, *Digital Threats: Research and Practice*, pp. 1–24 (2023).

[3] Ganfure, G. O., Wu, C.-F., Chang, Y.-H. and Shih, W.-K.: DeepWare: Imaging Performance Counters With Deep Learning to Detect Ransomware, *IEEE Transactions on Computers*, pp. 600–613 (2023).