ドメイン固有言語を用いた仮想化環境における複数レイヤ連携アクセス制御の実現

太田　萌[1]　品川　高廣[1]

概要：仮想化環境では，複数レイヤにまたがるリソースやセキュリティポリシーの動的かつ安全な制御が求められている．しかし，ハイパーバイザとゲスト OS 間のセマンティックギャップにより，高レベルなアクセス制御の実現が困難である．既存研究はハイパーバイザ改変や低レベルコードに依存しており，柔軟かつ抽象的なポリシー記述が難しい．本研究は，仮想化環境向けに設計したドメイン固有言語を用いて，人間が理解しやすい抽象度でアクセス制御ポリシーを記述できるフレームワークを提案する．アクセス制御ポリシーを eBPF プログラムへ変換し，ゲストとホストが協調して動的にポリシーを実施することで，ハイパーバイザの改変を要せずセマンティックギャップを緩和する．

1. はじめに

仮想化技術は現代のクラウドコンピューティング基盤を支える不可欠な要素である．ハイパーバイザを用いて一台の物理マシン上で複数の仮想マシンを動作させることにより，多様なワークロードを必要に応じて柔軟に実行することを可能にしている．このような仮想化環境は，ハイパーバイザや OS カーネルなど複数のソフトウェア層で構成されており，システム全体のセキュリティを担保するためには，複数レイヤにまたがって動的かつ安全にリソースやセキュリティポリシーを制御する仕組みが必要とされている．一方，eBPF はシステムソフトウェアを改変することなく，安全に検証されたプログラムを多様なフックポイントへ動的に挿入できる汎用基盤として急速に普及している．この eBPF による柔軟なアクセス制御機構を仮想化環境に応用することで，セキュリティや運用管理の効率化を実現する新たなアーキテクチャが期待される．

しかし，仮想化環境でアクセス制御をおこなう場合の課題として，セマンティックギャップの問題がある．これは，ハイパーバイザが管理するリソースと仮想マシン内部のゲスト OS のリソースとは抽象度には大きな差があり，ハイパーバイザからゲスト OS のセマンティックに基づくアクセス制御をおこなうことが難しいという問題である．ゲスト OS の構造を仮定することでハイパーバイザから仮想マシン内部の状態を推測する Virtual Machine Introspection (VMI) という技術は存在するが、その正確性や確実性、一貫性などに課題が残る．一方、eBPF によるアクセス制御は，アクセス制御のロジックを低レイヤのバイトコードを用いてプログラムとして記述する必要があり，ポリシーの抽象度が低いという問題点がある．

Lares [1] は，VMI とゲスト内でのフックを組み合わせ，脅威を判断するアクティブモニタリングを達成している．しかし，メモリ解析をする都合上 OS のバージョンに強く依存することや，複雑なメモリ解析のオーバーヘッドがかかる．Leonardi [2] は，eBPF を用いて CPU アフィニティ要求というゲスト OS のセマンティクスをホスト OS にマッピングする．また，Hyperupcalls [3] は，ゲストから提供された eBPF コードをハイパーバイザ内で実行し，コンテキストスイッチを避けて高速化する．しかし，これらの研究は目的が性能最適化であり，VMM 改変が必要であるほか，ロジックを開発者が低レベルなコードで記述する必要がある．bpfbox [4] は，eBPF を LSM フック等にアタッチすることで，単一 OS 内でのシンプルなプロセス隔離を実現する．しかし，制御範囲が単一 OS 内に限定されており，複数レイヤにまたがるアクセス制御はおこなっていない．

本研究では，人間が分かりやすい抽象度でのセキュリティポリシー記述を可能にする仮想化環境向け複数レイヤ連携アクセス制御のフレームワークを提案する．仮想化環境向けに設計した専用のドメイン固有言語を用いることで、eBPF の実装を抽象化した分かりやすいポリシー記述を可能にしつつ，そのポリシーを適切な eBPF に変換することで柔軟かつ動的なポリシー実施を可能にする．また，ゲスト OS とホスト OS が連携して eBPF によるポリシー実施をおこなうフレームワークを実現することで，セマン

[1] 東京大学大学院情報理工学系研究科

ティックギャップの問題を緩和して様々な抽象度のポリシーを階層的に記述・実施することを可能にする．

2. 設計

本手法では，eBPF によるカーネル内 I/O 制御と，ユーザ空間エージェントおよび vsock による通知と制御を明確に分離する．管理者はポリシーを DSL で記述し，ポリシージェネレータが解析してロジックに変換する．これにより，ゲストおよびホストのカーネル向け eBPF プログラムと，それぞれのユーザ空間エージェント用ポリシー設定を生成する．各エージェントは生成された eBPF プログラムをコンパイルしてカーネル空間にロードする．

ゲストでポリシー違反が発生すると，ゲストカーネル内の eBPF が検知してブロックし，vsock を介してホストエージェントに通知する．ホストエージェントは BPF Map を更新し，ホストカーネル内の eBPF に指示を送って I/O を高速にブロックする．また，必要に応じてハイパーバイザ API を介して VM 制御を行う．

DSL はホストスコープとゲストスコープを区別し，リソースを中心に複数レイヤをまたがるアクセス制御を宣言的に定義できるよう設計する．ゲストスコープでは LSM フックを用い，プロセス名やファイルパスといった高レベルなセマンティクスに基づいて検知条件を定義する．ホストスコープでは，その通知をトリガとして VM ディスクの Read-only 化や特定 IP のブロックなどのアクションを定義する．

3. 実装

提案 DSL を処理するために，パーサとコードジェネレータを実装した．パーサは ANTLR を用いてポリシーから AST を生成し，ジェネレータは AST に基づいて libbpf を利用したゲスト側およびホスト側の eBPF C コードと，各エージェント用ポリシー設定を生成する．eBPF コードは clang/llvm でバイトコードにコンパイルされ，BPF Type Format（BTF）および CO-RE（Compile-Once-Run-Everywhere）を活用することでカーネルバージョン差を吸収する．

ゲストカーネル内の eBPF がポリシー違反を検知すると，イベントデータを BPF RINGBUF 経由でゲストエージェントに送信する．ゲストエージェントは virtio-vsock を介してホストエージェントへ通知し，ホストエージェントは eBPF Map を更新してポリシーを動的に反映する．また，必要に応じて libvirt を用いた VM 制御も実行する．

4. おわりに

本研究では，ドメイン固有言語による柔軟な宣言的ポリシー記述を可能にし，ゲストとホストが連携して eBPF によるアクセス制御を実現する複数レイヤ連携フレームワークを提案した．今後は，対応フックポイントと eBPF アクションの拡充を進め，ユースケースを通じた有効性評価を行う予定である．

謝辞 本研究は、JST、CREST、JPMJCR22M3 の支援を受けたものである。

参考文献

[1] Payne, B. D., Carbone, M., Sharif, M. and Lee, W.: Lares: An Architecture for Secure Active Monitoring Using Virtualization, *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pp. 233–247 (online), DOI: 10.1109/SP.2008.24 (2008).

[2] Leonardi, L., Lettieri, G. and Pellicci, G.: eBPF-based Extensible Paravirtualization, *High Performance Computing. ISC High Performance 2022 International Workshops*, pp. 383–393 (2022).

[3] Amit, N. and Wei, M.: The Design and Implementation of Hyperupcalls, *Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pp. 97–112 (2018).

[4] Findlay, W., Somayaji, A. and Barrera, D.: bpfbox: Simple Precise Process Confinement with eBPF, *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, CCSW'20, p. 91–103 (online), DOI: 10.1145/3411495.3421358 (2020).