Sprofiler: 攻撃対象領域削減のためのコンテナネイティブなシステムコールフィルタルール生成機構の実装と評価

20G451 飯國 隆志 （最所研究室）

1 はじめに

コンテナ型仮想化技術は，OS 層の仮想化技術であり，アプリケーションの実行環境に用いられている．仮想マシンなどに比べ，仮想環境のディスク容量が小さく，起動が高速である．そのため，スケーラビリティ及びポータビリティの改善，高速なリリースサイクルを実現できる．しかし，コンテナ型仮想化技術を導入する上でセキュリティが課題となっている．本研究では，コンテナ型仮想化技術のセキュリティを向上させるシステムコールフィルタルールの生成機構の開発を行う．

2 脅威モデル

コンテナ型仮想化技術は，コンテナとホストがカーネルを共有することにコンテナ内で発行されたシステムコールがホストや全ホストに波及する可能性がある．これを利用し，コンテナ内からコンテナ外の権限を不正に利用する権限昇格攻撃のようなセキュリティリスクがある．このような権限昇格攻撃に対処する手法の 1 つとして，コンテナ内で発行されるシステムコールを制限し，攻撃対象領域を狭めることが有効である．Linux には，Seccomp というシステムフィルタ機構がある．コンテナにおいては，Seccomp のシステムコールフィルタルール (seccomp profile) によって，特定のシステムコールに対し，発行を許可，不許可，監査ログへの記録といったアクションを定義できる．

3 従来の手法と課題

seccomp profile は，コンテナランタイムとコンテナ上で実行されるアプリケーションによって発行されるシステムコールが許可されていなければ，コンテナが起動できない．プロダクション環境で利用される大量のコンテナごとにシステムコールフィルタルール (seccomp profile) を手動で作成し，保守するのは専門知識を有する技術者でも現実的には不可能である．そのため，システムコールフィルタルールの作成を支援する機構が必要である．従来の手法として，動的なシステムコール解析手法と，静的なシステムコール解析手法が挙げられる．

動的なシステムコール解析は解析対象のアプリケーションが実行中に使用したシステムコールのみを抽出する．解析中に使用されなかったシステムコールは抽出しないため，攻撃対象領域の削減効果は高い．その反面，アプリケーション内の分岐を網羅した seccomp profile を生成する場合には，解析のために複数回異なる条件でアプリケーションを実行する必要がある．

静的なシステムコール解析手法は，解析対象のアプリケーションを動かすことなく発行されるシステムコールを解析する手法である．アプリケーションを実行することなく，動的なシステムコール解析手法と比べ，システムコール解析が容易に可能となる．しかし，静的なシステムコール解析手法は，コンテナランタイムが発行する不要なシステムコールを許可してしまう．

この 2 つの手法は，システムコールのカバレッジと攻撃対象領域の削減効果という要素においてトレードオフの関係にある．これらの問題を解消し，コンテナに必要十分な seccomp profile を作成することが課題となる．本研究では，コンテナに適した seccomp profile を生成するシステムである Sprofiler の開発を行う．

4 Sprofiler

Sprofiler は，アプリケーションの実行形式ファイルの静的解析 (Static Analyzer) と，コンテナから発行されるシステムコールの動的解析 (Dynamic Analyzer) を組み合わせて seccomp profile を生成する．Static Analyzer は，静的なシステムコール解析手法によってアプリケーションから発行されるシステムコールを抽出する．バイナリ解析を行い，シンボルテーブルから関数名を抽出し，

関数名から呼び出されるシステムコールを特定し，seccomp profile を出力する．Dynamic Analyzer はコンテナを一度実行し，コンテナランタイムが発行するシステムコールを抽出する．コンテナ内でシステムコールが発行されたら，記録し，コンテナ停止時に seccomp profile を出力する．Static Analyzer と Dynamic Analyzer によって生成された seccomp profile を合成し，1 つの seccomp profile として使用する．

5　評価

Sprofiler によって生成された seccomp profile の攻撃対象領域を評価する．攻撃対象領域はシステムコールの許可数で評価することとする．本評価では，コンテナイメージごとに生成されたルールと Docker や Podman といった既存のコンテナ型仮想化ソフトウェアで提供されている default seccomp profile と比較する．

本評価では 4 種類のコンテナイメージを用いて評価する．基本的なアプリケーションとして，“Hello world” を実行するアプリケーションを C 言語と Go 言語で実装したものを用いる．また，本番環境で使用されるようなアプリケーションとして，CoreDNS，etcd を用いる．CoreDNS，etcd は，Kubernetes といったコンテナオーケストレーションシステム内部で利用されている．etcd は，Kubernetes クラスタのメタデータを格納する Key-Value Store である．CoreDNS は，Kubernetes 内の DNS として用いられている．

評価結果を図 1 に示す．Sprofiler が生成した seccomp profile は，Docker や Podman といったコンテナ型仮想化ソフトウェアの default seccomp profile よりも 200 種類以上のシステムコールの発行を制限するルールを生成した．

また，過去発見されたシステムコールに起因する権限昇格脆弱性を防ぐことができているかを評価した結果を表 1 に示す．本評価には，攻撃対象領域の評価に用いたコンテナイメージ及び seccomp profile と同様のものを用いる．評価結果としては，対象のコンテナに対し，今回検証した脆弱性はすべて低減することができた．

6　関連研究

sysfilter [1], confine [2] は，本研究と同様にシステムコールの静的解析を行い，システムコールフィルタルールの生成を行っている．Sprofiler と異なり，コンテナランタイム側の発行するシステムコールは解析していない．

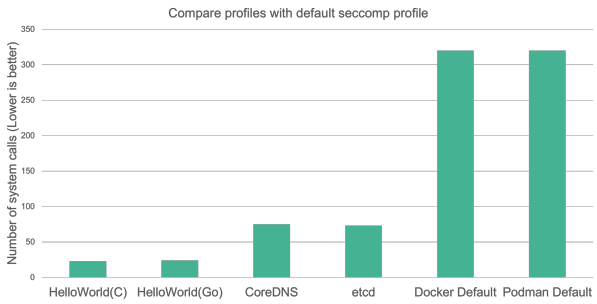

図 1: 許可するシステムコール数の比較

表 1: Vulnerabilities prevented by Sprofiler

CVE ID	Hello(C)	Hello(Go)	CoreDNS	etcd
2014-4699	✓	✓	✓	✓
2014-9529	✓	✓	✓	✓
2015-3290	✓	✓	✓	✓
2016-0728	✓	✓	✓	✓
2022-0185	✓	✓	✓	✓

DockerSlim [3] は動的なシステムコール解析手法を用い，seccomp profile を生成する．Sprofiler と比べ，アプリケーション内の分岐を一度の解析で網羅することができないため，必要十分な seccomp profile を生成するのは難しい．

7　おわりに

本稿ではコンテナ型仮想化技術の堅牢化を目的としたのシステムコールフィルタ機構のセキュリティポリシー自動生成システムである Sprofiler の設計と実装及び評価について述べた．Sprofiler が生成した seccomp profile は既存の seccomp profile よりも大幅に攻撃対象領域を減少させることに成功した．

参考文献

[1] N. DeMarinis, K. Williams-King, D. Jin, R. Fonseca, and V. P. Kemerlis, “sysfilter: Automated system call filtering for commodity software,” in *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*. San Sebastian: USENIX Association, Oct. 2020, pp. 459–474.

[2] S. Ghavamnia, T. Palit, A. Benameur, and M. Polychronakis, “Confine: Automated system call policy generation for container attack surface reduction,” in *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*. USENIX Association, Oct. 2020, pp. 443–458.

[3] docker-slim Organization. (2021, May) Docker-slim. [Online]. Available: https://github.com/docker-slim/docker-slim