eBPFプログラムを用いたVM内の安全な情報取得方式

堀 恭介[1]　光来 健一[1]

1. はじめに

Amazon EC2などのIaaS型クラウドは仮想マシン（VM）を提供しており，ユーザがVM内のシステムを自由に管理することが可能である．クラウド側はVM内のシステムの負荷や状態を監視することにより，VMのオートスケールやセキュリティ向上に活用することができる．クラウドによるVMの監視手法として，エージェント方式が一般的に用いられている．この方式はVM内にエージェントと呼ばれるソフトウェアを組み込み，クラウド側はエージェントと通信して情報を取得する．例えば，Amazon CloudWatchエージェントはログやメトリクスの収集，ログやトレースの分析などを行うために用いられている．

エージェント方式の問題点は，ユーザがエージェントのインストールやバージョンアップなどの保守作業を行う必要があることである．この作業を怠るとエージェントがVM内のシステムの脆弱性となる可能性がある．この問題を解決するために，イントロスペクション方式が用いられている．この方式はVMの外部のクラウド側からVMのメモリや仮想ディスクなどに直接アクセスし，OSのデータ構造やファイルシステムを解析することで情報を取得する．しかし，OSのデータ構造はOSのバージョンに大きく依存するため，ユーザが管理しているVM内のシステムに適用するのは容易ではない．また，最近のCPUによって提供されているAMD SEVなどによってメモリが暗号化されたVMに対しては利用することができない．

本稿では，クラウド側からVMにeBPFプログラムを動的に送り込んで実行し，VM内のシステムを監視するシステムTeleBPFを提案する．

2. TeleBPF

TeleBPFは**図 1**のようにeBPFアプリケーションとTeleBPFプロキシで構成され，VMにeBPFプログラムを送り込んで実行する．eBPFはBerkeleyパケットフィルタを拡張して，より複雑なプログラムを実行できるようにしたLinuxの機能である．eBPFアプリケーションがeBPFプログラムをOSにロードすることにより，OS内の様々な情報を取得することができる．TeleBPFではクラウド側で既存のeBPFアプリケーションを実行し，VM内のTeleBPFプロキシを経由してeBPFプログラムをVM内のOSにロードする．eBPFプログラムが取得した情報はTeleBPFプロキシ経由でクラウド側のeBPFアプリケーションに返される．このように，TeleBPFにおいてはeBPFプログラムがエージェントの役割を果たす．

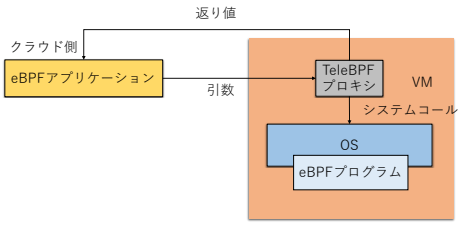

図 1　TeleBPFのシステム構成

eBPFアプリケーションがeBPFプログラムを透過的にVM内に送り込んでアクセスできるようにするために，TeleBPFではeBPFシステムコールの転送を行う．eBPFシステムコールはeBPFプログラムのOS内へのロードやデータを共有するためのマップの作成や読み書きなどを行うために，eBPFアプリケーションによって用いられる．TeleBPFはeBPFアプリケーションが実行したeBPFシステムコールをフックし，その引数をシリアライズしてVM内のTeleBPFプロキシに転送する．例えば，eBPFプログラムをロードする際にはその名前や種類，バイトコードなどがeBPFシステムコールの引数で指定される．TeleBPFプロキシは引数をデシリアライズしてeBPFシステムコールを実行し，その返り値をクラウド側に転送してeBPFア

[1] 九州工業大学
Kyushu Institute of Technology

```
BPF_ARRAY(counters, u64, 2);

RAW_TRACEPOINT_PROBE(sched_switch) {
    int zero = 1;
    u64 *val = counters.lookup(&zero);
    if (val) lock_xadd(val, 1);
    return 0;
}
```

図 2 Raw Tracepoints を用いた eBPF プログラム

プリケーションの実行に利用する．

TeleBPF は eBPF アプリケーションが実行する eBPF イベント制御システムコールについてもフックして転送を行う．eBPF イベント制御システムコールはシステム内のイベントに eBPF プログラムを関連づけ，イベントが発生した時に指定した eBPF プログラムが実行されるようにするために用いられる．例えば，OS 内の指定された関数が実行された時に eBPF プログラムを呼び出すようにすることができる．eBPF システムコールの場合と同様に，TeleBPF は eBPF イベント制御システムコールの種類と引数を TeleBPF プロキシに転送してシステムコールを実行し，その返り値を eBPF アプリケーションに転送する．

3. 実験

TeleBPF を用いて eBPF アプリケーションの挙動を確認する実験を行った．この実験では，**図 2** のような Raw Tracepoint を使う eBPF プログラムをロードし，カウンタ値をマップから定期的に読み出す eBPF アプリケーションを実行した．その結果，VM 内でコンテキストスイッチが行われるたびにカウンタ値が増加し，コンテキストスイッチの回数を取得できることが確認できた．

次に，様々なイベントに eBPF プログラムを関連づける eBPF アプリケーションを作成し，TeleBPF を用いて実行した結果を**表 1** に示す．カーネル内やプロセス内の関数呼び出し時と実行終了時，カーネル内やプロセス内に用意されているフックポイント実行時などに eBPF プログラムを実行することができた．kfuncs 以降のイベントは用いた Linux 5.4 のカーネルがサポートしていなかったため，動作が確認できなかった．

表 1 イベントの対応状況

イベント	動作
kprobes	○
kretprobes	○
Tracepoints	○
uprobes	○
uretprobes	○
USDT probes	○
Raw Tracepoints	○
system call tracepoints	○
kfuncs	カーネルが未対応
kretfuncs	カーネルが未対応
LSM Probes	カーネルが未対応
BPF ITERATORS	カーネルが未対応

4. まとめ

本稿では，eBPF プログラムを動的に VM 内の OS に送り込み，VM 内の情報を安全に取得するシステム TeleBPF を提案した．TeleBPF は eBPF アプリケーションによる eBPF 関連システムコールの呼び出しをフックし，そのシステムコールを VM 内の TeleBPF プロキシに転送する．そして，TeleBPF プロキシが代わりにシステムコールの実行を行い，その返り値をクラウド側の eBPF アプリケーションに転送する．

今後の課題は，より多くの BPF 関連システムコールやシステムファイルに対応し，様々な既存の eBPF アプリケーションを実行できるようにすることである．また，TeleBPF プロキシにアクセスする際のセキュリティの強化も課題である．認証や通信暗号化などを行うことによりアクセスを制限できるようにする必要がある．

謝辞 本研究の一部は，JST，CREST，JPMJCR21M4 の支援を受けたものである．また，本研究の一部は，国立研究開発法人情報通信研究機構の委託研究（05501）による成果を含む．