平成 30 年度 卒業論文概要			
所　　属	機械情報工学科	指導教員	光來 健一
学生番号	13237069	学生氏名	森本 晃穂
論文題目	軽量な仮想マシンを用いた IoT 機器の安全な監視		

1　はじめに

近年，インターネットにつながっていなかった機器がインターネットに接続されるモノのインターネット（IoT）が急速に普及している．IoT により，機器を遠隔から操作したり，機器の状態を遠隔から監視したりすることができるようになり，機器同士を通信により連携させることも可能になる．一方で，十分にセキュリティ対策が行われていない IoT 機器に対してインターネット経由での攻撃が増加している．そのため，IoT 機器への攻撃を検知するために侵入検知システム（IDS）が必要とされている．しかし，IoT 機器の監視対象システム内で IDS を動作させると，攻撃者に侵入された際に IDS を無効化される恐れがある．この問題を解決するために，サーバにおいては仮想化システムを用いて監視対象システムを仮想マシン（VM）内で動作させ，VM の外から安全に監視を行う IDS オフロードと呼ばれる手法が用いられている．しかし，Xen や KVM 等のサーバ向けの仮想化システムはオーバヘッドが大きいため，IoT 機器では利用するのが難しい．そこで，IoT 向けに軽量な仮想化システムである Xvisor [1] が開発されている．しかし，現在のところ，Xvisor における VM の外からの監視方法は未確立である．

本研究では，Xvisor のハイパーバイザの中から VM 内のシステムを監視するための IDS オフロード手法を提案する．

2　IoT 機器の監視

IoT 機器は急激に増加しており，2020 年には 200 億台を超えるとも言われている．一方で，IoT 機器はサーバほどにはセキュリティ対策がなされておらず，多くの機器に脆弱性が存在する．その上，IoT 機器はインターネットに接続されるため，世界中の攻撃者からネットワーク経由での攻撃を受ける可能性がある．例えば，2016 年には Mirai というマルウェアが IoT 機器を用いた大規模なサービス妨害攻撃を行い，様々なサービスが停止させられた．そのため，IDS を用いて IoT 機器内のシステムへの侵入を検知し，管理者に通知することが必要とされている．しかし，監視対象システム内で IDS を動作させて監視を行うのは安全ではない．監視対象システムに侵入された後，攻撃者に IDS の動作を停止される恐れがあるためである．

この問題を解決するために，サーバにおいては IDS オフロードと呼ばれる手法が用いられている．IDS オフロードは図 1 のように，仮想化システムを用いて監視対象システムを VM 内で動かし，VM の外から安全に監視を行う手法である．VM 内の監視対象システムに侵入を許した場合でも，VM の内部では IDS が動作していないため，攻撃者に IDS を無効化される恐れはない．しかし，サーバ向けに用いられている Xen や KVM 等の仮想化システムは汎用性が高く，豊富な機能を提供しているためオーバヘッドが大きい．そのため，これらの仮想化システムを IoT 機器で動作させて IDS オフロードを行うのは難しい．

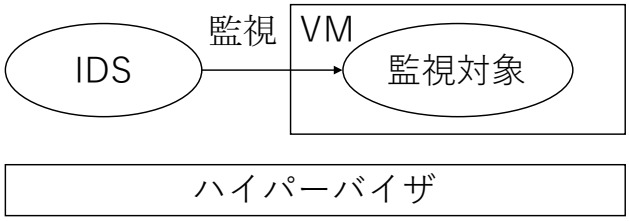

図 1　IDS オフロード

そこで，IoT 向けに Xvisor [1] と呼ばれる軽量な仮想化システムが開発されている．Xvisor ではハイパーバイザ内にすべての機能が実装されているため，一部の機能がハイパーバイザの外側のコンポーネントで実装されている従来の仮想化システムと比べて，機能間の通信オーバヘッドが小さい．また，IoT 機器を仮想化するのに必要最小限の機能だけを提供しているため，メモリ消費量が少ないという特徴を持つ．そのため，搭載されるメモリ量が比較的少ない IoT 機器でも動作させることができる．Xvisor を用いることで，IoT 機器内のシステムを VM を用いて動作させ，IDS オフロードを行うことが可能となる．しかし，Xvisor において VM の外からシステムの監視を行う手法はまだ確立されていない．

3　提案

本研究では，Xvisor のハイパーバイザの中から VM 内のシステムを監視するための IDS オフロード手法を提案する．従来の仮想化システムでは，IDS は特権を持った別の VM などにオフロードされることが多かった．提案手法では，図 2 のようにハイパーバイザ内で IDS を動作させることにより，VM の情報に小さなオーバヘッドでアクセスすることができ，監視性能を向上させることができる．さらに，提案手法ではハイパーバイザ上で一つの VM だけを動作させることを想定する．第一に，IDS オフロードを行うためにシステムを仮想化しており，VM は一つだけで十分なためである．第二に，Xvisor はサーバと比べると性能の低い IoT 機器で利用され，VM を二つ以上動作させるのは難しいためである．動作する VM を一つに限定することにより，IDS を特定の VM 向けにカスタマイズすることができ，監視性能を向上させることができる．

3.1　コマンドマネージャ

提案手法では，ハイパーバイザ内のコマンドマネージャで利用可能なコマンドとして IDS が提供される．Xvisor ではすべ

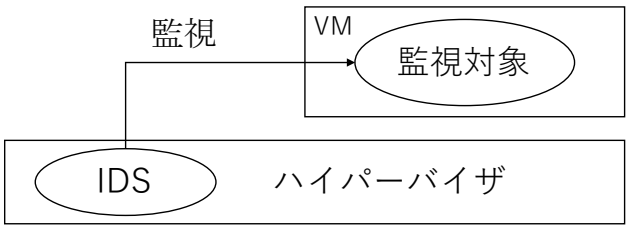

図 2　ハイパーバイザへの IDS オフロード

ての機能がハイパーバイザ内に実装されているため，コマンドマネージャがネットワークやシリアルポート経由でコマンドを実行する機能を提供している．例えば，仮想 I/O 関連のコマンド，デバイスドライバ関連のコマンド，ネットワーク関連のコマンド，ファイルシステム関連のコマンドなどが利用可能である．提案手法では IDS のためのコマンドを追加し，コマンド文字列が “ids” であった場合には続くサブコマンドの文字列に応じた VM の監視処理を実行する．

3.2　アドレス変換

ハイパーバイザ内の IDS は VM のメモリを解析して OS のデータを取得することにより VM 内のシステムの監視を行う．そのために，IDS は VM 内の OS データの仮想アドレスをハイパーバイザがアクセス可能なホストの物理アドレスに変換する必要がある．まず，IDS は OS データの仮想アドレスを VM の物理アドレスに変換する．このアドレス変換は VM のメモリ上に置かれているページテーブルと呼ばれるアドレス変換表を用いて行う．次に，VM の物理アドレスをホストの物理アドレスに変換する．VM の物理メモリは仮想化されており，VM にはホストの物理メモリの一部が割り当てられているためである．Xvisor ではホストの物理メモリの連続領域が VM に割り当てられているため，このアドレス変換は単純な加減算で行うことができる．

3.3　アドレス変換の自動化

VM 内の OS データにアクセスするために必要なアドレス変換を自動化できるようにするために，LLView [2] を Xvisor に移植した．LLView は IDS プログラムをコンパイルして生成された中間表現を変換し，メモリからデータを読み込む load 命令の直前にアドレス変換を行うプログラムを挿入する．また，IDS プログラム内で参照されている OS の変数を対応するアドレスに置き換える．LLView を用いることで，VM 内の OS のソースコードを用いて，OS の一部であるかのように IDS プログラムを作成することができる．

3.4　IDS の例

VM 内の OS バージョン等の取得，システムコールテーブルの取得，プロセスリストの取得を行うコマンドを作成した．

- **ids banner コマンド:** Linux カーネル内の linux_banner 変数からバナー文字列を取得する．取得した Linux のバージョンや Linux カーネルをコンパイルしたコンパイラのバージョンなどの情報を用いることで，特定のバージョンの Linux カーネルの脆弱性や特定のバージョンのコンパイラの問題を把握することができる．
- **ids syscall コマンド:** Linux カーネル内の sys_call_table 変数からシステムコールテーブルを取得する．このテーブルにはシステムコール発行時に呼び出される関数のアドレスが格納されており，テーブルの改ざんを検知することができる．
- **ids plist コマンド:** Linux カーネル内の init_task 変数からプロセスリストをたどり，task_struct 構造体に格納されたプロセスの ID と名前を取得する．プロセス名を調べることにより，不正なプロセスの実行を検知することができる．

```
XVisor# ids banner guest0
Linux version 4.9.0 (kourai@ccry
pt) (gcc version 5.4.0 20160609
```

図 3　取得されたバージョン情報等

```
XVisor# ids plist guest0
0: swapper
1: init
2: kthreadd
3: ksoftirqd/0
```

図 4　取得されたプロセスリスト

4　実験

提案手法により VM の監視が行えることを確かめる実験を行った．実験には，Intel Xeon X5675 の CPU，36GB のメモリを搭載した PC を使用した．ハイパーバイザとして Xvisor 0.2.10 を用い，ARM 用の QEMU 2.5.0 を用いて VersatilePB エミュレータ上で実行した．VM には仮想 CPU を 1 個，メモリを 96MB 割り当て，Linux 4.9.0 を動作させた．

まず，ids banner コマンドを実行して Linux のバージョン情報等を取得して表示させた．その結果，図 3 のように Linux のバージョンが 4.9.0 であり，gcc 5.4.0 を用いてコンパイルしたことが確認できた．次に，ids syscall コマンドを実行してシステムコールテーブルを取得したところ，システムコール関数のアドレスが 400 個表示された．最後に，ids plist コマンドを用いてプロセスリストを取得させたところ，図 4 のように表示された．このプロセスリストと VM 内で表示させたプロセスリストは一致していることが確認できた．

5　まとめ

本研究では，IoT 機器において軽量な仮想化システムである Xvisor を動作させ，ハイパーバイザの中から VM 内のシステムを監視するための IDS オフロード手法を提案した．提案手法を用いて 3 つの IDS コマンドを作成し，VM 内のシステムの情報を取得することが可能になった．今後の課題は，提案手法を IoT 機器に用いられている Raspberry Pi に実際に適用することである．また，VM のディスクやネットワークの監視を行えるようにすることも必要である．

参考文献

[1] A. Patel et al., Embedded Hypervisor Xvisor: A Comparative Analysis, PDP 2015.
[2] 植木あずさ, LLVM の中間表現を用いた IDS オフロードの開発支援, 九州工業大学卒業論文, 2015.