令和 元年度 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	18237202	学生氏名	越智 健人
論文題目	ARM 向けの軽量な仮想マシンを用いた IoT 実機の安全な監視		

1　はじめに

近年，自動車や家電などのあらゆるモノ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 では，インターネットを経由して遠隔で機器の操作やアップデートを行ったり，状態を遠隔から監視したりすることが可能になっている．一方，インターネットに接続することによって，これらの機器がサイバー攻撃を受ける恐れも出てくる．サイバー攻撃への対策の一つとして，侵入検知システム（IDS）を用いて攻撃を検知し，管理者への通知など必要な対応を行うことが考えられる．サーバでは IDS 自体が攻撃を受けるのを防ぐために，IDS を仮想マシン（VM）の外で動作させる IDS オフロード手法が用いられている．しかし，サーバ向けの仮想化システムを性能の低い IoT 機器において利用するのは難しい．そこで，IoT 向けに軽量な仮想化システムである Xvisor が開発されているが，Xvisor における IDS オフロード手法はまだ十分に確立されていない．

本研究では，ARM プロセッサを搭載した IoT 機器の実機上で Xvisor を用いて VM 内のシステムを安全に監視する XvIDS を提案する．

2　IoT 機器における IDS オフロード

IoT 機器は爆発的な増加を続けており，2020 年の現時点において既に 300 億台を超えている．5G 等の通信技術の実用化によってさらに応用が広がると考えられるが，サイバー攻撃のリスクについても考慮する必要がある．IoT 機器のメーカーはサイバー攻撃への対策を行っているものの，現状では IoT 機器の多くに脆弱性が存在している可能性が高い．実際の例として，ネットワークに接続できる自動車のシステムを乗っ取り，ステアリング等への介入に実験で成功している．また，大量の IoT 機器を乗っ取り，サービス妨害攻撃の発信元として利用するといった実害も出ている．こうした攻撃に対応するには，サーバと同様に IDS を用いてシステムの監視を行う必要がある．

システムの監視を安全に行えるようにするために，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．図 1 に示すように，IDS オフロードは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，IDS をその外側で動かす．この手法を用いることで，攻撃を受けて監視対象システム内に侵入された場合でも VM 内で IDS は動作していないため，攻撃者は IDS を無効化することができない．一般に，サーバでは Xen や KVM 等の汎用性の高い仮想化システムが用いられており，高機能な反面オーバヘッドが大きいため高い処理能力を持つハードウェアを必要とする．しかし，小型化・低価格化を求められる IoT 機器ではこのようなハードウェアを用いることができない場合が多いため，サーバ向けの仮想化システムを導入することは難しい．

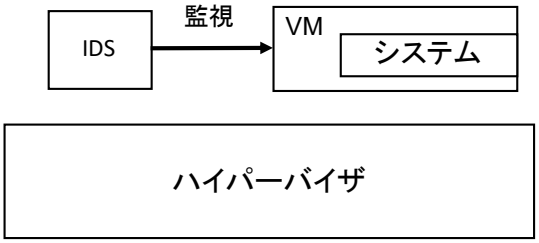


図 1　IDS オフロード

そこで，IoT 向けの軽量な仮想化システムである Xvisor[1] が開発されている．Xvisor は組込み機器でよく用いられる ARM プロセッサを主な対象としている．仮想化に必要な最小限の機能のみを提供するため，性能の低い IoT 機器でも動作する．また，必要な機能はすべてハイパーバイザと呼ばれる基盤ソフトウェア内に実装されているため，機能の一部をハイパーバイザ外部で実現しているサーバ向けの仮想化システムに比べてオーバヘッドが小さい．Xvisor を用いることで IoT 機器に IDS オフロード手法を適用することも可能になるが，その手法はまだ十分には確立されていない．先行研究 [3] では Xvisor の VM のメモリからの簡単な情報取得しかできておらず，エミュレータでの実行にとどまっている．

3　XvIDS

本研究では，ARM プロセッサを搭載した IoT 機器の実機上で Xvisor を用いて VM のメモリ，ディスク，ネットワークを安全に監視する XvIDS を提案する．サーバ向けの仮想化システムでは IDS を監視対象とは別の VM 上で動作させて IDS オフロードを実現することが多かったが，XvIDS では図 2 に示すようにハイパーバイザ内で IDS を動作させる．これにより，IDS のオーバヘッドを小さく抑えることができ，監視対象システムの性能への影響を最小化することができる．

XvIDS はハイパーバイザ内のコマンドマネージャのコマンドとして IDS の機能を提供する．コマンドマネージャは telnet プロトコルを用いてネットワーク経由でアクセスすることができる．リモートホストからログインして IDS コマンドを実行することで侵入検知を行い，その結果をリモートホストに返す．

3.1　メモリ上の OS データの監視

オフロードした IDS は VM のメモリを解析して OS データを取得することにより，VM 内のシステムの状態を監視する．その際に，IDS は 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物理アドレスに変換する．この変換には

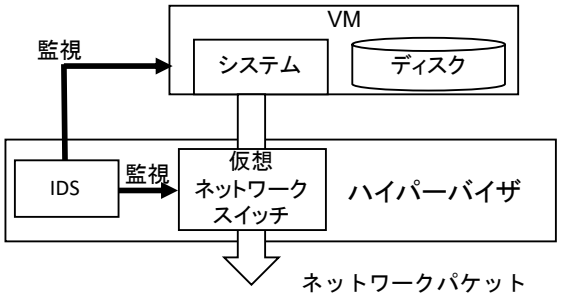
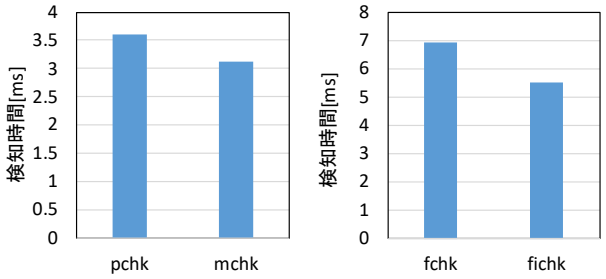


図 2　XvIDS における IDS オフロード



(a) プロセスとカーネルモジュール　　(b) ファイル

図 3　IDS の検知時間

VM のメモリ上にあるページテーブルと呼ばれるアドレス変換表を用いる．64 ビット ARM の場合，TTBR1 と呼ばれる CPU レジスタが指すメモリアドレスに 4 段のページテーブルが格納されている．変換元の仮想アドレスに基づいてページテーブルを検索していき，物理アドレスへと変換を行う．なお，Linux カーネルの仮想アドレスの上位 16 ビットはすべて 1 になっており，アドレス変換時には利用しない．

XvIDS ではこのアドレス変換を自動化することで，より効率的に IDS を開発することを可能にしている．そのために，LLView フレームワーク [2] を Xvisor に適用し，ハイパーバイザ内に実装された IDS のコンパイル時に自動的にアドレス変換のためのプログラムを組み込む．その結果，IDS が VM 内の OS データを読み込む際に自動でアドレス変換が行われるようになる．これにより，IDS の開発者は VM 内の OS のソースコードを用いて OS の機能を開発するように IDS を開発することが可能となる．

3.2　RAM ディスク上のファイルの監視

オフロードした IDS は VM のディスクを解析してファイルやディレクトリの監視を行う．Xvisor の VM はメモリ上に作成した RAM ディスクのみを用いており，仮想ディスク上のファイルやディレクトリはすべて起動時に SD カードから RAM ディスクに読み込まれる．そのため，VM のメモリ上の OS データを取得することでディスクの解析を行うことができる．IDS は監視対象のファイル等のパス名に従ってディレクトリをたどり，指定されたファイルやディレクトリの所在を探す．そして，ファイルの所有者などの属性やファイルの中身を調べることで侵入検知を行う．

3.3　ネットワークパケットの監視

オフロードした IDS は VM が送受信するパケットのヘッダを解析してネットワーク監視を行う．Xvisor ではすべてのパケットがハイパーバイザ内の仮想ネットワークスイッチを経由して送受信されている．IDS は仮想ネットワークスイッチの中でパケットを取得し，パケットに含まれるイーサネットヘッダ，IP ヘッダ，TCP ヘッダなどの解析を行う．仮想ネットワークスイッチにおいて IDS が攻撃を検知した場合には，通信元の IP アドレス等の情報を記録しておき，IDS コマンドの実行時に記録しておいた攻撃の情報を返す．

4　実験

XvIDS を用いて外部から VM の監視が行えることを確認するために，作成した IDS コマンドを用いて実験を行った．実験には 64 ビット ARM プロセッサと 1GB のメモリを搭載した Raspberry Pi 3 Model B の実機を使用した．この機器では Xvisor 0.2.11 を動作させ，VM には仮想 CPU を 1 個，メモリを 96MB 割り当てた．VM には BusyBox 1.27.2 をインストールし，OS として Linux 4.15.0 を動作させた．

まず，VM の起動後に，不正なプログラムを実行し，不正なカーネルモジュールをロードした．この VM に対して IDS コマンドを実行したところ，pchk コマンドは不正なプロセスを，mchk コマンドは不正なカーネルモジュールを検知することができた．その検知時間は図 3(a) のようになり，十分に短い時間で検知することができた．

次に，VM の起動後に VM のディスク上にいくつかの不正なファイルを作成した．この VM に対して IDS コマンドを実行したところ，fchk コマンドは不正なファイルの存在を，fichk コマンドは不正な文字列を含むファイルを検知することができた．その検知時間は図 3(b) のようになり，不正なプロセスやカーネルモジュールよりも検知に時間がかかることが分かった．

最後に，VM の 2001 番ポートを使うバックドアを模したサーバを設置し，ネットワーク上のホストから通信を行った．この VM に対して IDS コマンドとして net コマンドを実行したところ，検知した攻撃の種類と通信元の IP アドレスおよび MAC アドレスを取得することができた．

5　まとめ

本研究では，軽量な仮想化システムである Xvisor を Raspberry Pi の実機上で動作させ，ハイパーバイザの中から VM 内のシステムを監視する XvIDS を提案した．作成した IDS コマンドにより，VM のメモリ上の OS データ，RAM ディスク上のファイル，ネットワークパケットの監視が可能になった．今後の課題は，パケット内のデータ部の解析など，より詳細な監視を行えるようにすることである．

参考文献

[1] A. Patel et al. Embedded Hypervisor Xvisor: A Comparative Analysis. *PDP*, 2015.

[2] Y. Ozaki, H. Yamamoto, S. Kanemoto, and K. Kourai. Detecting System Failures with GPUs and LLVM. *APSys*, 2019.

[3] 森本晃穂. 軽量な仮想マシンを用いた IoT 機器の安全な監視. **九州工業大学卒業論文**, 2019.