九州工業大学 情報工学部 情報・通信工学科 2025 年度 卒業論文 概要

コース	ソフトウェアデザイン	指導教員	光來 健一
学生番号	222C1096	氏　　名	高巣 雄大

論文題目	Arm CCA の Realm VM を用いた安全なシステム監視

1　はじめに

近年，クラウドや IoT，エッジデバイスなどあらゆるシステムがインターネットに接続されるようになっている．その結果，インターネットからシステムへの攻撃も増加しているが，システムへの侵入を完全に防ぐことは難しい．そこで，侵入後に異常を早期検知し被害を最小化するために，侵入検知システム (IDS) による監視が重要となる．監視対象システム内で IDS を動作させると侵入者によって無力化される可能性があるため，IDS を監視対象システムの外に配置して監視する IDS オフロードが用いられる．IDS オフロードを安全に実行する手法の一つとして，Arm TrustZone によって提供されるセキュアワールドで IDS を安全に動作させ，ノーマルワールドで動作するシステムを監視する手法 [1] が提案されている．しかし，セキュアワールドは高い権限を持つため，IDS が攻撃を受けるとシステム全体の制御が奪われる恐れがある．

本研究では，Arm CCA で導入された Realm ワールドを用いて Realm 仮想マシン (VM) 上で IDS を動作させ，より安全に監視を行うことを可能にする CCAmonitor を提案する．

2　CCAmonitor

CCAmonitor は図 1 のように，Arm CCA の Realm ワールド内で動作する Realm VM 上で IDS を実行し，ノーマルワールドで動作するシステムを監視する．Realm ワールドはノーマルワールドから隔離されており，ノーマルワールド内の侵入者から IDS への攻撃を防ぐことができる．セキュアワールドとは異なり，Realm ワールドはデバイスに直接アクセスすることができず，アクセスできるメモリも制限されている．そのため，IDS が攻撃を受けたとしても，システムへの攻撃を行うことは難しい．

CCAmonitor では，Realm VM 上の IDS が Realm マネジメントモニタ (RMM) を介してノーマルワールドのメモリ上にある OS データを取得することで監視を行う．Realm VM は OS データのアドレスを指定して，Realm サービスインタフェース (RSI) を用いて RMM を呼び出す．RMM は

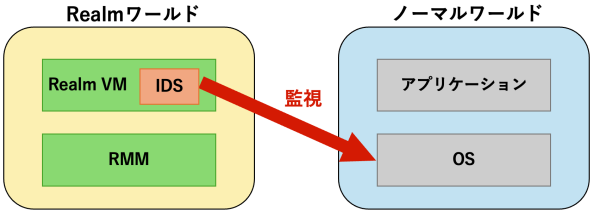

図 1. CCAmonitor のシステム構成

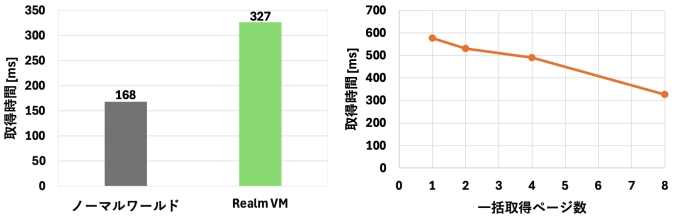

図 2. proc ファイルシステムの情報の取得時間

指定されたアドレスに対応するノーマルワールドのメモリページを一時的にマッピングする際，1 回の呼び出しで複数ページを一括取得してマッピングし，そのデータをコピーして Realm VM に返す．IDS は LLView [2] を用いて開発し，IDS が OS データにアクセスする際に透過的に RMM を呼び出して必要なメモリデータを取得する．

IDS はノーマルワールドの OS の仮想アドレスを用いて OS データを指定するため，RMM がそれを物理アドレスに変換してメモリアクセスを行う．そのために，ノーマルワールドのメモリ上にある 5 段のページテーブルを参照してアドレス変換を行う．ページテーブルのアドレスやページサイズなどの情報は，ノーマルワールドから Realm ワールドに切り替わる際に保存されるレジスタ値から取得する．

3　実験

CCAmonitor を用いてノーマルワールドの proc ファイルシステムによって提供されるシステム情報を取得する実験を行った．実験の結果，Realm VM 上の IDS がシステム情報を取得することができ，ノーマルワールド内で得られる情報と一致することが確認できた．

また，IDS が一括取得するページ数を変更しながら，システム情報を取得するのにかかる時間を測定した．比較のために，ノーマルワールド内で proc ファイルシステムを読み出す時間も計測した．結果は図 2 に示すようになり，Realm VM での取得時間は 8 ページを一括取得した時に最小となり，ノーマルワールド内での取得時間の 1.9 倍であった．

4　まとめ

本研究では，Realm VM を用いて IDS を安全に実行し，ノーマルワールドのメモリデータを取得して監視を行う CCAmonitor を提案した．今後の課題は，IDS がノーマルワールドのメモリを直接マッピングして効率よくアクセスできるようにすることである．

参考文献

[1] M. Guerra et al. Introspection for ARM TrustZone with ITZ Library. QRS 2018.
[2] Y. Ozaki et al. Detecting System Failures with GPUs and LLVM. APSys 2019.