TERM 2018 年秋学期
RDMA を用いた，遠隔ベアメタルマシンデバッグのための論理メモリの参照

Arch B3 石川 達敬 (tatsu)*　　Adviser: 松谷 健史 (macchan)†
Adviser: 空閑 洋平 (sora)‡

1　概要

遠隔ベアメタルマシンの物理メモリの監視アーキテクチャを確立，外部リソースで動作するメモリ監視環境の実現を目指す

2　背景と目的

OS デバッグやセキュリティフォレンジックでは，物理・仮想メモリを解析していく際に，仮想マシンをたて，実行環境をサンドボックス化し，デバッグしていく手法が一般的であった．解析の際に用いられる環境及びツールとして，Xen，あるいは QEMU+KVM があげられる．また，関連ソフトウェアとして，libvmi[1] や google/rekall などもあげられる．この現状の課題として，カーネルパニック時の解析における物理的なメモリ解析手段が存在しないという点があげられる．カーネルパニック時には，OS が提供していた，マシン内部で利用できる API およびシンボル，機構などが使えなくなるためである．具体的には，MMU(Memory Management Unit)，システムコール，カーネルシンボルなどである．これらが使えないことにより，VM として仮想化したマシンを，QEMU などのハイパーバイザーを通じて解析するしか方法が存在しなかった．

この課題を解決するために，本研究では，外部リソース，具体的には，外部電源，外部プロセッサで動作するマシンからリモートホストへの物理メモリの解析手法を確立することで，対象ホストのメモリ監視アーキテクチャを構築することを目指す．

3　と x86-64 Linux におけるページウォーク

x86-64 Linux(以後 Linux) では，ページング機構を通して，プロセス中で仮想アドレスを物理アドレスへと変換している．この行程には，後述するテーブルをたどることで，最終的に物理アドレスを算出している．この，仮想アドレスから物理アドレスを算出する行程のことをページウォークと呼ぶ．ページウォークは，CR3 レジスタの値を起点としている．CR3 レジスタとは，ページング機構の最上位テーブルである PML4 の物理メモリを保持するレジスタである．この CR3 の値は，プロセスごとに設定されており，この値と，テーブルの中身によって，プロセスは，アドレス空間の独立を保証されている．変換テーブルは，四段階存在する．PML4(Page Map Level 4), Page Directory Pointer, Page Directory, Page Table である．プロセス中における仮想アドレスは，最大で 48bit 長で表されるが，この値はそれぞれのテーブルのインデックスを表しており，47-39bit が PML4, 38-30bit が Page Direcotry Pointer, 29-21bit が Page Directory, 20-12bit が Page Table である．11-0bit には，オフセットが格納されている．

4　本研究のアプローチ

本研究におけるアプローチの概要として，前述のページウォークのロジックを解析していく．四段階のテーブルの全エントリを本研究のプログラムで保

*慶應義塾大学 環境情報学部
†慶應義塾大学大学院 政策・メディア研究科特任講師
‡慶應義塾大学大学院 政策・メディア研究科特任講師

持し，仮想アドレス空間を復元していく．

5　環境

環境は，監視プロセスが動くホスト，監視多少のホスト共に，x86-64 Linux とする．

6　実装

実装および実験をしていく上での前提条件として，CR3 レジスタの値を監視対象ホストから通知する方式とした．CR3 はレジスタであるため，現在の値をメモリから参照できないからである．

実装の流れとして，まず，通知された CR3 の値を引数として受け取る．この値を起点に，全てのエントリの値を取得し，配列に保持する．値を取得する際は，PCIe-Eth-Bridge の手続きを呼び出す．最終的に，四段階目である Page Table の値を取得したのち，そこから得られる物理アドレスの値を取得する．得られた値が，各階層で，どのインデックスを通過してきたのかを保持した構造体に値を保持しているため，最後に出力する際，各インデックスを対応分左にビットシフトさせることで，仮想アドレスを復元する．

7　評価

評価を，カーネルパニック時の対象プロセ卯 s の論理メモリ参照の可否とする．状態を正しく取得できているのかを確認するための手段として，現在の UNIX TIME を保持する変数を扱うプロセスを監視する．評価手順として，このプロセスが起動している最中に，カーネルパニックを意図的に発生させる．その上で本研究の実装である監視プログラムを実行する．監視されるプロセスは，現在の UNIX TIME を出力すると同時に，監視ホストへ，CR3 の値を通知するために，CR3 の値を出力しているプロセスでもある．結果として，変数の値を正しく取得することができた．

8　参考文献

参考文献

[1] libvmi.com `http://libvmi.com/`

[2] 青柳 隆宏 (2013) . はじめての OS コードリーディング　UNIX V6 で学ぶカーネルのしくみ

[3] Daniel P. Bovet(2007) . 詳解 Linux カーネル第 3 版