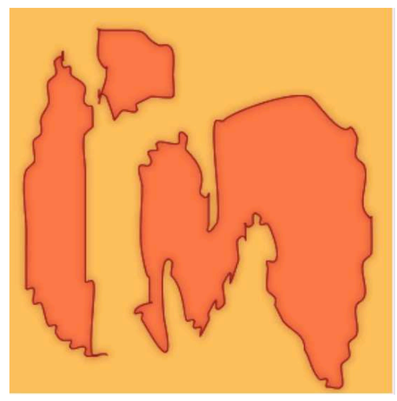
SecHack365

EEMU~セキュアなエミュレーター~

学習駆動コース　坂井ゼミ　木下和巳



背景

・はりぼてOSが動くx86なPCのCPUエミュレーターを作っていく過程で、x86なpcについて色々学びたい
・ROP攻撃を検知するセキュリティー機構を備えるエミュレーター(アンチROP)
・はりぼてOSとは、本「30日でできる！OS自作入門」で作るOS
・エミュレーターは、ハイパーバイザーとも違い、VMとも違い、dockerとも違う

成果物その1

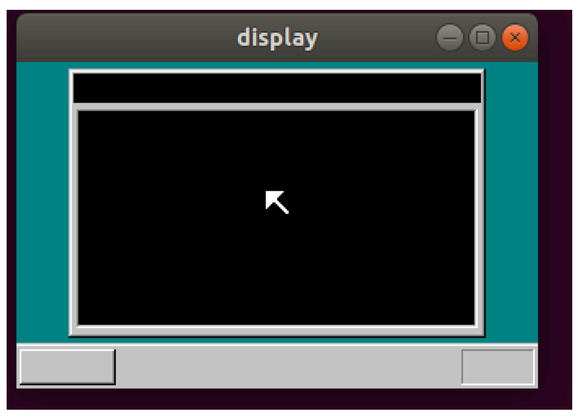
EEMU(セキュアなエミュレータ）

実装したい全機能一覧

・GUI・IO port
・タスクスイッチ・割り込み
→実装済み
・マウス・キーボード・time・アンチROP
→未実装

アンチROP

ROP判定方法は、return命令が実行された後に、
数バイト前にcall命令がないとROPと判断する。
ROPされたと分かれば、何かしらの方法で通報する。
例えば、メールやSlackなどである。




```
void add_rm32_r32(Emulator *emu){
        emu->EIP++;
        ModRM modrm(emu);
        uint32_t rm32 = modrm.GetRM32();
        uint32_t r32 = modrm.GetR32();
        modrm.SetRM32(rm32 + r32);
        emu->update_eflags_add(rm32, r32);
}

void add_r32_rm32(Emulator *emu){
    emu->EIP++;
    ModRM modrm(emu);
    uint32_t r32 = modrm.GetR32();
    uint32_t rm32 = modrm.GetRM32();
    modrm.SetR32(rm32 + r32);
        emu->update_eflags_add(r32, rm32);
}
```

このプロジェクトはこれで完結ではなくて、まだまだこ実装していくつもりです。
今後はこっちのブログで公開していきます
-> http://kazuminkun.hatenablog.com/

成果物その2

はてなブログの記事
タイトル「エミュレーターの作り方（はりぼてOSが動く）」

他のドキュメントとの違い

はりぼてOSを動かすことができる
エミュレーターの作り方について書いてある。他のドキュメントではOSが動かない。

主な目次

・マウス・キーボード
・プロテクトモードに突入するための、16bit実装
・far-jump命令
・セグメンテーション
・Port IO
・割り込み

公開URL

http://kazuminkun.hatenablog.com/entry/2020/02/24/001130

記事の”はじめに”

この記事は、はりぼてOSが動くi386 PCエミュレータの作り方の解説記事です。ただし、「[OtakuAssembly vol.1]」(自作エミュレーターでOSを動かす章)や[自作エミュレータで学ぶx86アーキテクチャ-コンピュータが動く仕組みを徹底理解!]に書いていることは重複して書かないつもりです。なので、この記事を読む前に上記の本で基本的なエミュレーターの作り方を学んでから、本記事を読むのが筆者のおすすめです。従って、この記事は、上記の本には書かれていない、はりぼてOSを動かすための技術やtipsを説明します。読んでいて、分からないなと思うところがあると思うのですが、その分からないまま先を読めば、疑問の答えが書いてあることが多数あります。分からないまま読んでいきましょう。いずれ、答えば出てくるので。

ただし、パソコンの挙動を完全理解しているわけではないので、筆者のエミュレーターは、間違えているかもしれません。もし、間違いや、こうしたほうがいいよ、などはコメント欄に書いてもらえると嬉しいです。

エミュレータを作る上でPCの動作の仕組みをよく理解していなければいけません。
本記事の目的ですが、筆者は、はりぼてOSが動くエミュレータの仕組みや技術を知ったり実際に作ったりすることにより、PCの仕組みを理解する助けになればいいと思ってます。また、他のエミュレーターを作る際の参考にしてもらえば幸いです。実際に、筆者は、低レイヤーに興味があり、実際に作ることによりPCの仕組みを理解したいのでエミュレーターを作り始めました。

NICT 国立研究開発法人 情報通信研究機構