利用wllvm编译linux，可以在每个目录下得到bc文件，还有根目录下的vmlinux。接下来比较由这些bc文件生成的dot文件是否和由vmlinux生成的dot文件相同。

# 1

把所有位于linux-5.4.284的bc文件整理到一个文件夹bcfiles下

#!/bin/bash

# 检查是否提供了两个参数

if [ "$#" -ne 2 ]; then

echo "Usage: $0 <source\_directory> <destination\_directory>"

exit 1

fi

# 从命令行参数获取源目录和目标目录

src\_dir="$1"

dst\_dir="$2"

# 创建目标目录（如果不存在的话）

mkdir -p "$dst\_dir"

# 使用find命令查找所有.bc文件并处理

find "$src\_dir" -type f -name "\*.bc" | while read file; do

# 计算相对于源目录的相对路径

relpath=$(realpath --relative-to="$src\_dir" "$file")

# 构建目标文件路径

dstfile="$dst\_dir/$relpath"

# 确保目标文件所在的目录存在

mkdir -p "$(dirname "$dstfile")"

# 复制文件

cp "$file" "$dstfile"

done

分别处理每个bc文件，以生成dot文件

#!/bin/bash

# 检查是否提供了一个参数

if [ "$#" -ne 1 ]; then

echo "Usage: $0 <source\_directory>"

exit 1

fi

# 从命令行参数获取源目录

src\_dir="$1"

# 使用find命令查找所有.bc文件并处理

find "$src\_dir" -type f -name "\*.bc" | while read file; do

# 获取文件所在目录

dir=$(dirname "$file")

# 切换到文件所在目录

cd "$dir" || { echo "Failed to cd to $dir"; continue; }

# 获取文件名（不包括路径）

base=$(basename "$file" .o.bc)

base01="${base#.}"\_dot

mkdir -p "$base01"

cd "$base01" || { echo "Failed to cd to $base01"; continue; }

# 使用 opt 工具生成 .dot 文件

opt -passes=dot-cfg "$file"

echo "Generated .dot files for $file at $dir/$base01"

done

对每个dot文件进行分析，统计其基本块数

#!/bin/bash

# 检查是否提供了一个参数

if [ "$#" -ne 1 ]; then

echo "Usage: $0 <source\_directory>"

exit 1

fi

# 从命令行参数获取源目录，并确保结尾有斜杠

src\_dir="$1"

src\_dir=${src\_dir%/}/ # 如果结尾没有斜杠，则添加一个

# 初始化总基本块计数器

total\_blocks=0

# 使用 find 命令从 src\_dir 开始递归查找所有 .dot 文件

while IFS= read -r -d '' file; do

if [ -f "$file" ]; then

# 获取相对路径

rel\_file="${file#$src\_dir}"

# 使用 grep 命令查找所有包含 [shape=record,color= 的行

num\_blocks=$(grep -c '\[shape=record,color=' "$file")

# 打印当前文件的基本块数量

echo "File: $rel\_file, Number of basic blocks: $num\_blocks"

# 累加到总基本块计数器

total\_blocks=$((total\_blocks + num\_blocks))

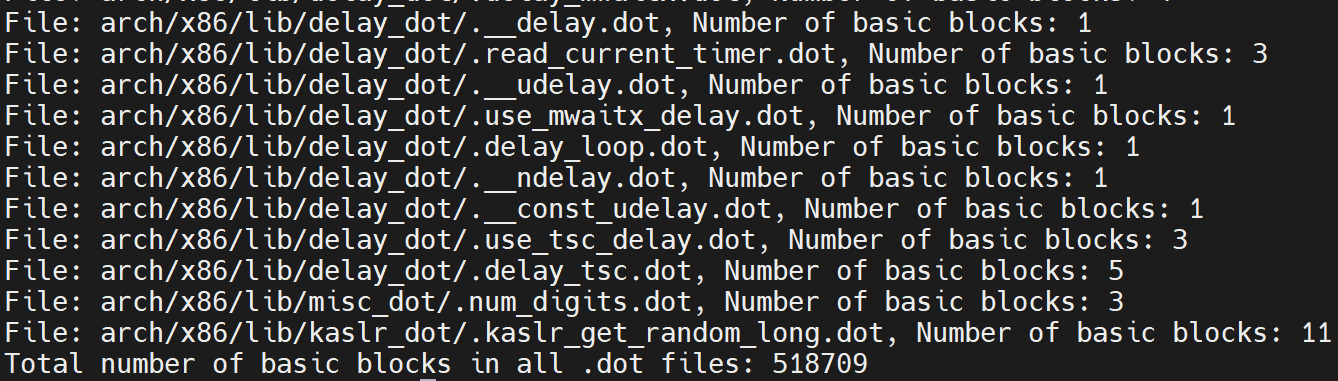
fi

done < <(find "$src\_dir" -type f -name "\*.dot" -print0)

# 打印总的基本块数量

echo "Total number of basic blocks in all .dot files: $total\_blocks"

结果：



共48281个dot文件，对应48281个函数

# 2

extract-bc vmlinux

llvm-dis vmlinux.bc

#!/bin/bash

# 定义文件名

filename="vmlinux.ll"

# 使用grep搜索字符串并计数匹配的行数

count=$(grep -c '; Function Attrs:' "$filename")

# 输出结果

echo "找到 $count 行包含 '; Function Attrs:'"



共47732个函数，比前者少549个

# 3

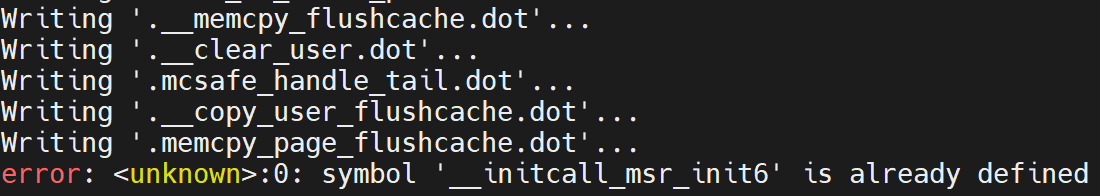
结论：两者不一样，应该按照后者来分析(但是感觉按照前者来分析也可以比较添加配置项后的CFG差异)

不一样的原因：需要理一下编译的过程，看看Makefile

# 4

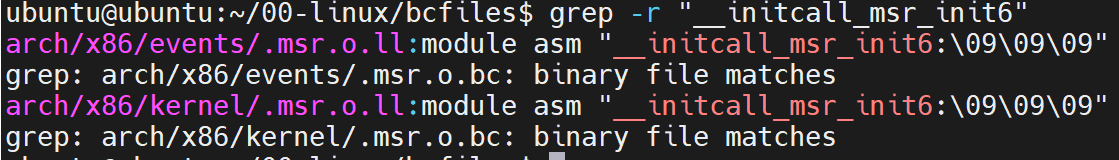
后者获取CFG遇到的问题：

opt -passes=dot-cfg vmlinux.bc



查看vmlinux.ll，发现有两行module asm "\_\_initcall\_msr\_init6:\09\09\09"导致重复定义

原因：在arch/x86/events/msr.o和arch/x86/kernel/msr.o这两个文件里都定义了一次，当把文件链接到一起时，就出现了重复定义的问题



解决方法：把vmlinux.ll中的module asm删掉，但不确定会不会对产生CFG有影响