AFL-FUZZ 启用 llvm

坏虾 / 2017-11-09 14:52:00 / 浏览数 4815 安全技术 技术讨论 顶(0) 踩(0)

Fuzzing技术被证明是当前鉴别软件安全问题方面最强大测试技术。
当前大多数远程代码执行和特权提升等比较严重的漏洞都是使用Fuzzing技术挖掘的。
然而Fuzzing技术仍然存在着覆盖率低的缺陷。
而许多的代码漏洞需要更大的路径覆盖率才能触发，而不是通过纯粹的随机尝试。
而AFL-FUZZ 是一款采取遗传算法生成用例的FUZZ工具。可以有效的解决这些问题。
为了提升Fuzzing的效率，AFL-FUZZ可以采用LLVM来使用`afl-fast-clang` & `afl-fast-clang++`去替换`afl-gcc`进行插桩。而且当使用`afl-fast-clang`来编译的时候可以使用`__AFL_LOOP`
`__AFL_LOOP`可以一次调用，发送多条模糊测试用例。

下面介绍一下如何启用`LLVM`模式。

1、http://releases.llvm.org/download.html#3.5.2 下载所需要的源码包。

```
cfe-3.5.2■clang-tools-extra-3.5.2■ compiler-rt-3.5.2 ■llvm-3.5.2
xz -d ■■ ■5■■■
tar xvf  ■■■■■■■■■■5■tar■■
```

2、整合源码

```
mv cfe-3.5.2.src clang
mv clang llvm-3.5.2.src/tools

mv clang-tools-extra-3.5.2.src extra
mv extra/ llvm-3.5.2.src/tools/clang/

mv compiler-rt-3.5.2.src compiler-rt
mv compiler-rt llvm-3.5.2.src/projects/
```

3、编译安装

```
mkdir build-3.5
cd build-3.5/
../llvm-3.5.2.src/configure --enable-optimized --enable-targets=host-only
make -j 4
make install
```

4、编译安装afl-fuzz的llvm模块

```
cd afl-2.50b/
cd llvm_mode/
make
make install
```

然后进入`afl-2.50b`目录。重新`make install`激活安装成功的`afl-fast-clang`

OK，这样就可以使用`afl-fast-clang`来进行插桩编译了。
如：

```
   SET(CMAKE_CXX_COMPILER "afl-clang-fast++")

while (__AFL_LOOP(1000))
  {
      XXXXXX
 }
```

最后使用AFL-FUZZ进行模糊测试，会发现效率提升了很多。

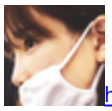
点击收藏 | 0 关注 | 0
上一篇：同盾科技&freebuf 联合出品... 下一篇：NB-IoT的“NB”小漫画
1. 4 条回复


坏虾 2017-11-09 15:14:27

有段写错了。。大家注意下。

mv compiler-rt-3.5.2.src compiler-rt
mv compiler-rt llvm-3.5.2.src/projects/

0 回复Ta


hades 2017-11-09 16:34:40

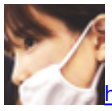
@坏虾 已修正ing

1 回复Ta


坏虾 2017-11-09 16:47:54

@hades 写完不让修改，真头疼。

0 回复Ta


hades 2017-11-09 17:01:11

@坏虾 下版本更新

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板