OSX 被动 Fuzz 框架

译者：银雁冰

原文链接：https://github.com/SilverMoonSecurity/PassiveFuzzFrameworkOSX

原文作者：SilverMoonSecurity

OSX 被动 Fuzz 框架

1. 这是什么？

本框架基于内核模式中的被动 inline hook 机制，主要用于 fuzzing OSX 内核漏洞。

简单来说，这是一个典型的内核驱动，它 inline-hook 了与 IOKit 框架和内核服务相关的导入 API。

如果内核发生了 crash，你可以收集内核 dump 文件来重现漏洞。

你可以关注我的 twitter：@Flyic (of moony li) 来获得更多详细信息。

我们将会在东京的 PacSec 2016(10.26/10.27) 大会上，进行“Active fuzzing as complementary for passive fuzzing”的主题演讲，这之后，才会公布相关源代码。

https://pacsec.jp/speakers.html

这个被动 fuzzing 框架是基于 fG! 所写的针对 OSX 平台的

“the_flying_circus”Rootkit。在此特别感谢 fG!

（A Mountain Lion rootkit for Phrack #69! Copyright (c) fG!, 2012, 2013 - reverser@put.as - http://reverse.put.as All rights reserved.)

2. 运行要求

原则上，这个被动 fuzzing 框架支持 Mac Pro/Air 上的普遍 OSX 版本。

根据我们的经验，从 10.11 到 10.11.6 的内核版本对被动 fuzzing 几乎没有干扰。

这个框架已在 Mac Pro，10.11.6，KDK_10.11.6_15G31.kdk 上测试。

3. 如何使用？

重要提示：

运行内核驱动会导致内核突然崩溃，使得你失去所有的数据。请自行承担使用这个内核驱动的风险。

3.1 快速开始

如果你只是为了好玩而尝试这个被动 fuzz，请按照这样快速操作：

a. 加载快速被动 fuzz 的驱动

```
sh-3.2# chown -R root:wheel ./quick-pasive_kernel_fuzz.kext
sh-3.2# kextutil ./quick-pasive_kernel_fuzz.kext
```

b. quick-pasive_kernel_fuzz 会在内核模块中出现

```
sh-3.2# kextstat
```

然后你就会在内核模块列表中看到这个驱动

3.2 使用 ThunderBolt 线进行调试的完整调试

这个开始引导适用于任何运行 Mac OSX 系统的机器（例如：MacPro, MacAir, Mac Mini）

这个方案要求需要另外一台被调试用的 OSX 机器和额外的 ThunderBolt 线。

ⅰ. 在被调试的 OSX 机器上

a. 准备 KDK 和 nvram

Ⅰ. 下载 KDK_10.11.6_15G31.kdk（以这个 KDK 为例）并且安装在你的 Mac 电脑上。

Ⅱ. 将 kernel.development 拷贝到系统目录文件夹并且同步内核缓存

```
sh-3.2#cp -fr
/Library/Developer/KDKs/KDK_10.11.6_15G31.kdk/System/Library/Kernels/ker
nel.development* /System/Library/Kernels/
sh-3.2# kextcache -invalid /
sh-3.2# reboot
```

III. 设置用以调试的启动参数

```
sh-3.2# nvram boot-args="debug=0x566 kdp_match_name=firewire
fwkdp=0x8000 pmuflags=1 kext-dev-mode=1  -v"
sh-3.2# reboot
```

b. 加载用来被动 fuzzing 的内核驱动

```
sh-3.2# chown -R root:wheel ./pasive_kernel_fuzz.kext
sh-3.2# kextutil ./pasive_kernel_fuzz.kext
```

c. 你的 Mac 可能会在等待进一步调试时发生内核崩溃

ⅱ. 在 OSX 主机上：

重要提示：

请保持 ThunderBolt 始终连接在两台机子上，因为即插即用(PnP)机制不支持已崩溃的内核。

a. 准备 KDK

1. 下载 KDK_10.11.6_15G31.kdk（以这个 KDK 为例）并且安装在你的 Mac 上。

这个步骤不是必须的，但是我们强烈建议这样做。在接下来的步骤中调用 lldb，lldb 会在调试主机和被调试机间匹配*.dSYM 符号文件。否则，在你的调试过程中，符号信息不会显示。

b. 调试已崩溃的目标机

Ⅰ. 启动 fwkdp 服务

```
flyic-pro:pasive_kernel_fuzz.kext root1$ fwkdp
```

Ⅱ.lldb 调试

```
sh-3.2# cd
/Library/Developer/KDKs/KDK_10.11.6_15G31.kdk/System/Library/Kernels

sh-3.2# lldb ./kernel.development

(lldb) kdp-remote localhost

The debugger would wait until the target machine  crashes, and then you can type any command for debugging including collect core dump file.
```

3.3 使用有线局域网进行完整调试

因为只有老式的 OSX 机器支持有线局域网（例如：旧的 MacMini），所以这种类型的调试并不普遍。

等待被完善。