本文作者: riusksk (信安之路特约作者 & 个人公众号漏洞战争)

本文汇总了作者这么多年来做漏洞挖掘的学习经验,具有非常大的参考意义,其中的方法不限于某个专业和方向,无论你做什么职业或者从事安全的哪个方向,都是有参考意义的,所以为大家推荐这个文章给大家学习。

### 0x01 学习资料来源

官方文档永远都是最好的一手资料,通常它包含原理,使用方法以久示例代码,对于攻击面分析非常有参考价值,而且如果自写 fuzzer,甚至可能因此少写很多代码。

有时候单纯把系统 API 过一遍都可以挖到一堆,即使是现在的 windows 与 macOS/iOS系统,只要你愿意,依然可以挖到 0day,如果运气好的话,一个 API 挖出 10+CVE 也是有可以的,因为我早已验证过!

## 0x02 如何做好 fuzzing

Fuzzing 三要素: **目标、策略、样本**。假设有人开源一款 fuzzer 或者新型方法论的 paper,很多人的第一印象都会是:漏洞刷光了才发出来的吧。

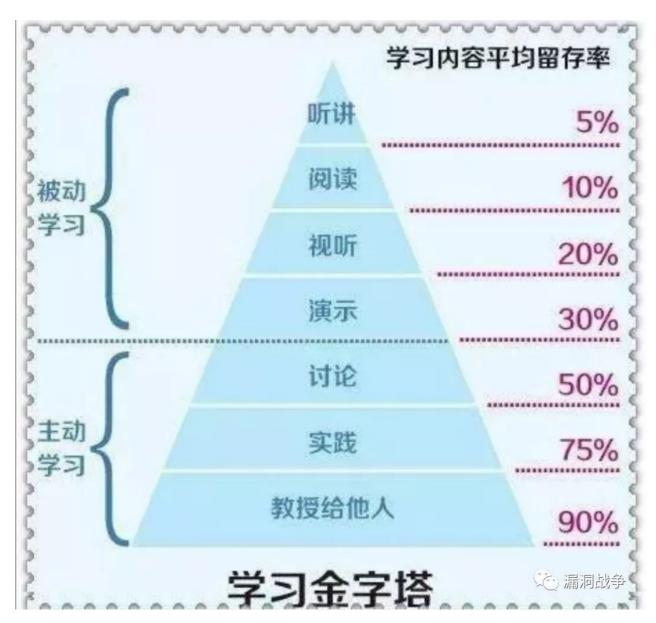
从发布者的角度来看,确实如此,但是从上述三要素来看,刚不一定。

以 afl 刚开源为例,虽然很多开源库已被挖了不少,但后来被移植到很多地方,如 android,还有内核,也有不错的产出; afl 变异策略本身还是暴力为主,一些条件语句有时难进入,所以有人拿比较语句中常量值作为变异值插入,以增加代码路径;有人拿 afl 去 fuzz stagefright,使用相同的测试程序,但因样本不同,导致结果不一样。所以,fuzzer 的效果主要还是依赖这三要素的。

#### 0x03 保持正确的学习方法

保持学习,但有时方法更重要。每个月都会一些安全工具或技术文档公开,以前我都会及时 跟进学习,但很多只是被动学习,遗忘率太高了。

下面一张图就很好的介绍了这一情况,但要求你老教授他人,显然是不现实,或者写作是一种更好的方式。学习也不在量多,关键还是在于主动思考与实践的投入时间的长短。



就像大家都看 project zero 的 buglist,有的人用来搞 pr 研究,有的人用来刷 src,有人动手调试分析,甚至试着写 fuzzer 验证思路......

#### 0x04 细节决定你是否专业

专业与业余的差别有时只体现在细枝末节上。以前刚开始做渗透测试时,可能就是开网站,看见输入框时写句 <script> alert(1)</script> 测试下,还有踩点收集信息这种小事,很随意,自然难逃"随机挖洞"的命运。

正是这些看似小事的东西,很多专业者都搞出了很自动化、工程化的系统出来,从子域收集,ip 提取,字典定制化生成,新业务监控,威胁情报收集,漏洞扫描、告警,甚至自动生成报告,提交至 zdi、hackerOne 及各大 SRC 平台,实现技术套现……

### 0x05 自动化很重要

肉眼挖洞的能力是需要,但不能全部人工,否则产出相当有限,如果你想追求精品漏洞又有很多时间(不代表自动化就出不了精品),不嫌累的话,请随意! 我挖到的 CVE 大多是在睡觉的时候挖的。

# 0x06 推荐学习

安全研究者的自我修养

原创 轻松理解什么是模糊测试

**原创** 信息安全学习很枯燥,很难坚持,一点小小*感*悟分享给你