浅谈从CTF到实战

讲师：Atum

内容简介

INTRODUCTION

01 从CTF与实战的挑战

02 CTFer在实战中的入手点

从CTF与实战的挑战

漏洞挖掘对象的放大：

- CTF：规模较小的binary程序
- 实战：最大至一个多层的子系统
- 分析对象规模的差距直接导致了漏洞挖掘的方法完全不同，在CTF中，由于Binary程序较小，所以逆向整个binary是可能的。但是实战中，由于分析规模较大，通常要通过攻击面分析、数据流分析、fuzz等手段进行漏洞分析。

案例：挖掘分析安卓二进制漏洞

- 攻击面分析，寻找潜在的可能被攻击的地方，通常是攻击者可以注入数据的地方，如USB模块(电脑要通过USB与系统交互)等
- 选择攻击面进行进一步数据流、控制流分析。分析输入数据在程序中是如何传递的，在传递过程中寻找是否可能存在漏洞。或者根据对攻击面的了解编写fuzzer进行fuzz。

从CTF与实战的挑战

存在漏洞的目标和系统种类繁多：

- CTF：较为单一的ELF文件，linux系统
- 实战：各种各样的目标文件、各种各样的系统。
- 不同系统不同目标文件的利用套路可能完全不同，如之前讲的ptmalloc的利用姿势在jemalloc中大多数都无法使用，不同目标文件的调试方法和exploit编写方法也有所不同。

案例：利用Adobe Flash AVM中的二进制漏洞

- 将Adobe flash AVM的目标文件flash.ocx IDA中，结合前些年泄漏的AVM plus源代码，定位漏洞触发的位置以及注意观察点。以及定位script层漏洞触发关键代码与native层代码的关系
- 在script层编写Action script触发漏洞，编写中可能需要用到actionscript专有的堆喷、堆风水等技术，在native层在关键触发位置中和观察点下断点，观察内存和变量
- 需要频繁的在Script层和Native层进行切换，需要了解script层的代码与native层代码的对应关系

从CTF与实战的挑战

漏洞利用所牵动的环境变大：

- CTF：从规模较小的binary中寻找可用的机制、gadget来进行exploit
- 实战：漏洞利用可能牵动整个漏洞所在的子系统，所以需要对研究对象的子系统有较深的理解
- 漏洞的利用

案例

- android-libcutils库中整数溢出导致的堆破坏漏洞的发现与利用 by gongguang
- 可以看到，要利用一个整数溢出漏洞需要了解图形子系统的原理以及binder IPC的原理，exploit行数数千行。

CTFer在实战中的入手点

尽管有这么多挑战，但是还是有不少相对简单的入手点，由简入繁

推荐入手点1：智能设备如智能路由器、智能摄像头

- 推荐理由：物联网设备的保护机制相对较为简单，且研究目标多为ELF文件，与CTF较像
- 难点：需要一定的电子知识，通常需要串口调试
- Example：CVE-2015-3036 NetUSB内核栈溢出漏洞

推荐入手点2：Windows下Office系列漏洞

- 推荐理由：Office漏洞非常丰富，漏洞分析中的一些方法与CTF具有一定的相似性，如下断点的方法几乎是相同的，较多且大多数漏洞也都是可以在Native层进行构造exploit，适合CTFer学习。
- 难点：office漏洞需要使用od或者windbg，需要熟悉新的工具，另外windows下的利用套路与linux也不太一样，所以需要熟悉相关套路。
- Example：CVE-2015-1641word 类型混淆漏洞

CTFer在实战中的入手点

推荐入手点3: 阅读各种漏洞分析报告

- 推荐理由：网上各类漏洞分析报告非常多，通过阅读这些漏洞分析报告能够帮助理解真实环境漏洞的相关情况
- 难点：不少漏洞分析报告较为枯燥，需要耐心
- Examples:
  - Adobe在野漏洞：CVE-2016-4117漏洞分析
  - 类型混淆漏洞（CVE-2015-1641）分析
  - One Perfect Bug: Exploiting Type Confusion in Flash
- 推荐一本书：《漏洞战争:软件漏洞分析精要》

无论是实战还是CTF，都需要勇于迎接挑战和永不放弃的geeker精神

The End