编号：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验 | 一 | 二 | 三 | 四 | 五 | 六 | 总评 | 教师签名 |
| 成绩 |  |  |  |  |  |  |  |  |

武汉大学国家网络安全学院

课程实验(设计)报告

题 目： 网络对抗演练综合实验

专业(班)： 信安6班

学 号： 2021302181156

姓 名： 赵伯俣

课程名称： 网络对抗演练

任课教师： 陈治宏

2024年 6 月 16日

# 一 实验内容总结

在《网络对抗演练》这门课程中，共完成了4个实验，实验内容和实验心得如下。

## （一）Web安全

### 1 实验内容

（1）Web漏洞利用：

* 使用XPath成功登陆和注入获取XML文件内容。
* 通过xpath成功登陆XML/XPath Injection
* 通过xpath成功注入获取xml文件中所有的内容
* 实施反序列化漏洞攻击，将数据写入session中，并在session中修改cookie输出flag。
* 通过index.php将反序列化写入session中，并完成字符串逃逸，在session.php中更改cookie成功输出flag

（2）进行SSRF攻击，获取内网地址，并渗透内网服务器以获取flag。

* 172.72.23.21 - SSRF 获取本地信息
* 172.72.23.22 - 代码注入
* 172.72.23.23 - SQL 注入
* 172.72.23.24 - 命令执行
* 172.72.23.25 - XML 实体注入
* 172.72.23.26 - CVE-2017-12615
* 172.72.23.27 - Redis 未授权

（3）信息收集：

* 对目标单位“武汉大学”进行全面的信息收集，包括网络侦察、端口扫描、服务识别等，将结果汇总输出。

（4）系统提权：

* 对Ubuntu系统进行提权，以获取root权限。

### 2 实验报告

实验报告详情，包括实验关键步骤、过程结果、实验数据分析等可见《2组-WEB安全》。

### 3 实验心得体会

在本次实验过程中，我掌握了XPath在Web应用中的应用，包括成功登陆和注入获取XML文件内容，通过反序列化漏洞攻击和SSRF攻击，学习到如何利用Web漏洞进行系统渗透和内网攻击；在信息收集方面，通过对 武汉大学的全面信息收集，学习到使用各种信息收集工具和技术，并能够将收集到的信息进行整理和汇总；在系统提权方面，我掌握了针对Ubuntu系统的提权方法。

总的来说，虽然在实验开始前并没有对web安全有比较深层次的研究，但是在实验操作的过程中以及小组共同解决问题时让我对网络安全中的web安全领域有了更深入的理解，还提高了我的实践能力和问题解决能力。

## （二）逆向工程

### 1 实验内容

（1）day1

* 将下面的实验代码进行修改，并使用编译软件进行编译：

将代码中的key数组修改为学号

数组长度与组员数量相同

* 对编译后的程序使用UPX Tool进行加壳
* 对加壳后的程序使用ESP定律法进行脱壳
* 对脱壳后的程序使用IDA进行分析，并对IDA反编译后的伪C代码进行注释、变量重命名，截图，需要操作的函数为：main函数、加密函数
* 编写python脚本，求解出flag的内容

（2）day2

* 使用Directory Monitor通过文件监控分析恶意代码创建了哪些文件？
* 使用RegShot分析恶意代码修改了哪个注册表？修改该注册表的作用？
* 分析恶意代码预留的后门账号？
* 分析Wireshark恶意代码从公网下载了什么内容，具体的链接是什么？
* 分析Wireshark恶意代码向公网上传了什么内容，目标的域名是什么？
* 使用Netstat分析恶意代码监听了什么端口？端口访问返回的内容是什么？

（3）day3

* 实验题目1：
  + 完成逆向题的qiantang apk题目，实现购买成功的功能实现效果
  + 对Android解压，进行文件格式的基本分析
  + 静态分析程序
  + 结合不同逆向工具，分析程序核心算法逻辑
  + 利用程序逻辑逆向分析支付功能的具体实践
  + 实现具体功能的patch，得到购买成功的效果
* 实验题目2：
* 完成逆向题的seh.apk题目，实现购买成功的功能实现效果
* 对Android解压，进行文件格式的基本分析
* 静态分析程序
* 结合不同逆向工具，分析程序核心算法逻辑
* 利用程序逻辑逆向支付功能的具体实践
* 实现具体功能的patch，得到购买成功的效果
* 选做题
  + 完成逆向题的RE\_Cirno题目，拿到flag字符串
  + binwalk分析提取程序
  + 静态分析程序
  + 结合不同逆向工具，分析程序核心算法逻辑
  + 利用程序逻辑逆向写出求解算法程序
  + 利用逆向求解算法拿到flag

（4）day4

* 完成目标程序rps.apk的逆向分析, 拿到flag字符串
* 完成逆向题的**wzry.apk**题目，拿到勒索软件的锁屏密码

（5）day5

* 完成目标程序bang.zip中 signed.apk的逆向分析
* signed.apk题目的分析
* dex文件结构的学习
* 软件加固的识别
* 基础的软件脱壳
* 完成目标程序**jstz.apk**的逆向分析
* unidbg的学习
* unidbg环境的搭建
* unidbg代码的编写
* jstz.apk题目的分析
* 完成目标程序getflag.apk的逆向分析。
* getflag**.apk**题目的分析
* 自定义算法的学习
* 自定义算法的逆向
* hook代码的编写

### 2 实验报告

实验报告详情，包括实验关键步骤、过程结果、实验数据分析等可见《2组-逆向工程》。

### 3 实验心得体会

本周实验的前四天难度并不大，只要根据上课讲的内容即可成功复现出每个任务的结果，第五天的任务二难度比较大，因为先前并没有接触Java，所以在完成该部分的任务时会有些吃力，并且在配置环境的过程中也出现了不小的障碍。在完成后面三天的任务时比较有趣，因为在尝试了各种工具之后使用了各种方法最后成功夺取到flag时会有不小的成就感。

在本周的实验中学习到了基本的Android程序逆向分析方法，结合多种逆向框架进行基础的逆向分析，hook。在实际操作的过程中也能够比较熟练的使用jadx、ida等逆向工具。

## （三）PWN-二进制

### 1 实验内容

（1）day1

完成目标程序bof的利用。

（2）day2

完成64位目标程序axb\_2019\_brop64的利用。

完成64位目标程序easylib的利用。

完成静态编译程序PicoCTF\_2018\_can-you-gets-me的利用。

（3）day3

完成32位目标程序axb\_2019\_fmt32的利用。

完成目标程序pb的利用。

利用IDA插件keypatch完成对AWD文件夹下format、gets、pwn的程序修补。

（4）day4

完成64位目标程序roarctf\_2019\_realloc\_magic的利用。

（5）day5

完成64位目标程序paper的堆利用。

完成64位目标程序pwn的堆利用。

### 2 实验报告

实验报告详情，包括实验关键步骤、过程结果、实验数据分析等可见《2组-PWN二进制》。

### 3 实验心得体会

在本周的试验任务中详细完成了利用pwn对可执行程序分析漏洞并编写自动化脚本的过程中学到了栈溢出、格式化字符串漏洞和堆溢出等漏洞形式，并在完成题目的过程中完善了对于相应的理论知识的理解，也学习到了多种获取目标主机shell的方法。

本周实验的任务因为之前没有基础，所以在实验的过程中出现了不少的问题，在互联网上搜索资料解决问题的同时是我学习到了很多额外的方法以及学会了很多在调试自动化脚本上的技巧。

在本周的实验中最感兴趣的是格式化字符串漏洞，因为在编程的过程中确实会出现使用printf等函数不规范的现象，虽然程序运行的结果没有变化，但是却存在着非常危险的漏洞，这一点警示着我在以后的编程时要时刻按照规则进行编写，不能因为偷懒而造成程序在安全层面的漏洞

## （四）漏洞挖掘

### 1 实验内容

（1）漏洞挖掘环境搭建

* 搭建windows 内核调试环境
* 宿主机上搭建windbg调试环境，并基于第一个实验中的windows虚拟机搭建内核调试环境
* 安装Bindiff
* 安装Ubuntu Fuzz和编译环境
* 在Ubuntu中安装afl-fuzz工具
* 编译upx fuzz目标程序
* 宿主机安装VisualStudio2022，并安装Win10 SDK开发环境

（2）漏洞挖掘实战

* 使用AFL-FUZZ 挖掘最新版本UPX漏洞，学会安装使用该漏洞挖掘工具并进行漏洞挖掘，记录全流程操作和细节。

（3）CVE-2021-1682

* 使用第一天搭建的windows 虚拟机镜像分析CVE-2021-1682,写出poc，漏洞的形成原因，poc执行中的约束条件及漏洞调试的报告。

（4）固件漏洞挖掘环境搭建

* 完成Binwalk工具安装，通过分析固件获取隐藏在固件中的密钥(截图固件中的登陆密钥)。
* 完成静态编译的ARM架构的Hello World程序，使用qemu-arm-static、模拟运行该程序。(截图打印个人学号- Hello \*学号\*)

（5）嵌入式交叉编译

* 完成完成交叉编译工具链生成(生成MIPS大小端工具链) - 截图运行结果
* 完成静态编译Busybox工具(MIPS大小端编译-LSB、MSB)
* LD\_PRELOAD劫持随机数生成(改成自定义生成函数-格式学号+hook+ok) - 截图运行效果

（6）路由器漏洞实战

* 完成DIR-645溢出漏洞，通过栈溢出漏洞利用，获取shell - 截图运行结果
* 完成TOTOLINK逻辑漏洞，通过逻辑漏洞，获取后台管理界面 - 截图后台界面

### 2 实验报告

实验报告详情，包括实验关键步骤、过程结果、实验数据分析等可见《2组-漏洞挖掘》。

### 3 实验心得体会

在本周的实验中，我负责binwalk固件分析，交叉编译工具链生成和D-Link DIR-645几个部分的任务，本周的实验在试验任务上的难度相比于前几周的任务并不算大，但是在配置实验环境时需要消耗掉不少的时间，在机型binwalk固件分析时由于事先并不知道在Kali虚拟机中自带有binwalk工具，所以在Windows和Ubuntu系统中都配置了相应的环境，在完成交叉编译工具链生成任务时小端工具链部分的复现比较顺利，但是在大端工具链部分出现了很多问题，最后在老师后面的演示指导下成功编译出了大端工具链，在编译途中出现了虚拟机死机与编译时间过长等问题，最后借助网络资料克服。

在完成D-Link DIR-645溢出漏洞分析时，大部分的时间都在配置固件仿真环境，我们尝试了在Ubuntu22.04环境下安装firmware-analysis-toolkit工具，并且尝试了该工具的v0.2，v0.3以及plus等各个版本，但是都会出现死循环或者程序报错问题，在Kali环境下使用该工具也会出现相同报错，并且在虚拟机的共享文件夹中执行该工具会出现网络连接失败的错误。尝试使用firmware-analysis-plus作者提供的Ubuntu镜像文件也会出现报错。最后在查阅资料时发现可以使用AffityOS虚拟机成功实现硬件仿真。最后的漏洞部分比较简单，成功获取到了目标路由器的shell。

# 二 实验评语

实验报告的考评依据实验内容完整度、实验步骤清晰度、实验结果与分析正确性、实验心得与思考的全面性、实验报告文档的规范性等五个维度综合考评，建议分值和标准如下。

|  |  |
| --- | --- |
| 85-100 | * 实验内容完整或者有超出课程实验大纲的内容； * 实验步骤详尽，能够体现完整的实验过程； * 实验结果正确且实验数据分析得当； * 实验心得与思考全面并且有自己的独立思考； * 实验报告文档规范、排版整齐。 |
| 75-84 | * 实验内容较为完整； * 实验步骤较为详尽，能够体现实验过程； * 实验结果正确且实验数据分析较为得当； * 实验心得与思考全面； * 实验报告文档规范、排版较为整齐。 |
| 60-74 | * 实验内容有缺失； * 实验步骤不够详尽，不能够体现完整的实验过程； * 实验结果部分正确； * 实验心得与思考无或者不够深入； * 实验报告文档规范性有待增强。 |
| 60以下 | * 实验内容严重缺失、实验态度不够端正 * 实验步骤不够详尽，不能够体现完整的实验过程； * 实验结果部分正确； * 实验心得与思考无或者不够深入； * 实验报告文档规范性有待增强。 |

|  |  |
| --- | --- |
| 任课教师评语 |  |