CTF前置知识杂烩

Web – Web安全

Web安全

> 随着 WEB 2.0、社交网络、微博等等一系列新型的互联网产品的诞生，基于 WEB 环境的互联网应用越来越广泛，企业信息化的过程中各种应用都架设在 WEB 平台上，WEB 业务的迅速发展也引起黑客们的强烈关注，接踵而至的就是 WEB 安全威胁的凸显，黑客利用网站操作系统的漏洞和 WEB 服务程序的漏洞得到 WEB 服务器的控制权限，轻则篡改网页内容，重则窃取重要内部数据，更为严重的则是在网页中植入恶意代码，使得网站访问者受到侵害。
> 在 CTF 竞赛中，WEB 也是占比重很大的一个方向之一，WEB 类的题目种类繁多，知识点细碎，时效性强，能紧跟时下热点漏洞，贴近实战。
> WEB 类的题目包括但不限于：SQL 注入、XSS 跨站脚本、CSRF 跨站请求伪造、文件上传、文件包含、框架安全、PHP 常见漏洞、代码审计等。

可能你读到这里已经听不懂前面再说什么东西了，没关系，随着慢慢的学习，所有的疑问都会迎刃而解

基础前置知识

- 计算机网络基础知识(参考https://blog.csdn.net/day0713/article/details/123209328)
- Web基础知识（参考https://www.it610.com/article/1547470485221085184.htm）
- HTTP协议相关知识(包括但不限于HTTP协议结构与各类Header)
- **PHP**（很重要，后续大部分漏洞原理的讲解都会以PHP语言为例）
- HTML/Javascript：大致了解即可，前期不需要太过深入
- Linux常用命令（先了解什么是Linux，然后去了解Linux下常用的一些命令 参考https://aistudio.baidu.com/aistudio/projectdetail/37491?hmsr=aladdin）
- 对漏洞的模糊概念（https://blog.csdn.net/qq_33409012/article/details/107407992或https://zhuanlan.zhihu.com/p/393635352。不一定要完全理解，但是要对常见安全漏洞有一个大概的认知）
- MySQL数据库语法（SQL）
- Python的基本语法（Python的用处很大，所以一定要认真学会）
- Burp Suite的基本使用方法（比如如何抓包、改包、重发等常用操作）

P.S：PHP、Python、SQL、HTML、Linux命令之类的知识可以在菜鸟教程(https://www.runoob.com/)学习，如果感觉纯文字难以理解也可以去找一些视频教程来学习，主要在于掌握语言的基础语法。
P.P.S：顺便推荐一个渗透测试的随查手册，常见的一些漏洞利用手法与Payload在上面都可以找到资料，不过是纯英文的https://book.hacktricks.xyz/welcome/readme

常用工具

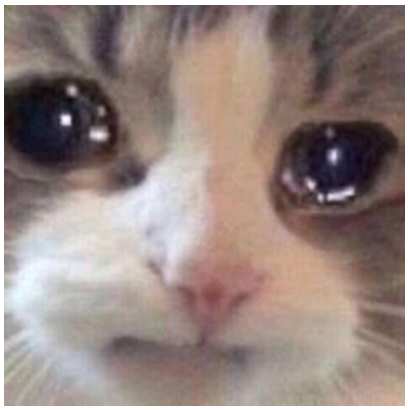
- **一颗好学的心**
- **遇到问题学去查一下 / 下载软件的时候明辨是非 不要一刀9999**
- Java8、python2/3、curl、ncat、phpstudy(Windows)、mamp(Mac)：一切学习的开始
- Vscode(全语言) / Idea(Java) / sublime / goland(Golang) / phpstorm(Php) / pycharm(Python)：语言编译器，可以在编程时给到极大的代码提示
- BurpSuite – 常用于抓改包（链接: https://pan.baidu.com/s/1tmevi474Eycb6tf8su-zyw 提取码: u62s））
- nmap – 信息收集扫描工具
- sqlmap – SQL注入工具
- dirsearch/dirmap – 敏感目录扫描工具
- FireFox/Chrome – 浏览器，推荐使用这两个（二选一）
- HackBar – 浏览器插件
- PHPStudy – 一键搭建PHP+MySQL的环境，适合初学者使用
- 蚁剑AntSword – WebShell管理工具，使用指南：https://www.freebuf.com/articles/web/270488.html
- Docker
- ……

常见名词

1. Exp/Exploit：中文 ”利用”，指利用系统漏洞进行攻击的动作。
2. Poc：全称”Proof of Concept”，中文”概念验证”，常指一段漏洞证明的代码。
3. Payload：中文”有效载荷”，指成功exploit之后，真正在目标系统执行的代码或指令。
4. Shellcode：简单翻译”shell代码”，是Payload的一种，由于其建立正向/反向shell而得名。
5. 渗透测试：渗透测试是通过模拟恶意黑客的攻击方法，来评估计算机网络系统安全的一种评估方法，这个过程包括对系统的任何弱点、技术缺陷或漏洞的主动分析。
6. OWASP TOP10：OWASP项目中的“十大安全漏洞列表”，OWASP Top 10不是官方文档或标准，而只是一个被广泛采用的意识文档，被用来分类网络安全漏洞的严重程度，目前被许多漏洞奖励平台和企业安全团队评估错误报告。这个列表总结了Web应用程序最可能、最常见、最危险的十大漏洞，可以帮助IT公司和开发团队规范应用程序开发流程和测试流程，提高Web产品的安全性。
7. Kali Linux：一款知名的渗透测试系统，集成了大量常用的渗透测试工具，基于Debian开发。
8. WebShell：以asp、php、jsp或者cgi等网页文件形式存在的一种代码执行环境。
9. Shell：命令解释器，可以理解为可以执行命令的地方。
10. 一句话木马：通常指一段较短的含有命令执行函数的代码，黑客通过向服务器中注入这段代码来实现获取目标服务器的权限：

后续补充

考虑到部分同学进度比较快或是有一定基础，所以后续的一些资料也一起发出来：
https://ctf-wiki.org/web/php/php/
Java/Node.js/Golang等语言后续也是需要学习的，不过建议大家入门阶段还是以PHP入手。



Q&A

- 我需要严格按照上面的顺序学习吗?

  显然不用，按照自己的计划慢慢安排就好了，但是切记不能三天打鱼两天晒网
- 学完这些我就能黑天黑地黑教务系统了吗?

  显然是不可以的，安全领域的东西太多太多了
- 有的东西我实在是看不明白咋办?

  如果文章遇到不明白的地方了，例如三元运算符/各种奇奇怪怪的语法/函数，建议停下来，先把不明白的地方弄明白，如果硬着头皮读下去，遇到不会的就直接跳过，那其实和没读没有什么明显的区别，切勿好高骛远，从基础走起
- 要学多久?

  安全是一个学习很久很久的过程，而且每天基本要投入大量的时间去了解之前不会的知识，亦或是偷偷冒出来的全新知识

来自`Ha1c9on`的一些碎碎念

首先！不能光说不练！感觉我看懂了就直接跳过了！这绝对不可以！！！！

php无疑是入门web方向最简单的语言之一，作为脚本语言，调试方便，语法简单。

但是随着技术的不断进步，许多企业已经不再采用php作为首要的编程语言。**如果已经能掌握相关php的语法知识，对代码审计已经有一定基础**(即拿到代码除了一些不认识的函数能较为快速的分析出这都是干啥的，亦或是已经亲手debug过一些常用框架的漏洞：thinkphp/laravel/yii等)，还是建议去多入门Java、Golang等目前来说相对主流的语言

Java 这玩意前期看起来可能会比较痛苦，涉及到很多其独有的语言特性，但是慢慢熟悉以后就会好很多，人们常说，如果掌握了一门语言，其他语言入门也会相对较快，他们的思想其实都差不多，比如学完了php反序列化，如果再去看java的cc反序列化，相对理解起来就会容易很多。

假设你已经做了许许多多的题目，所有的基本知识点都掌握的差不多了，但是在打CTF比赛的时候依然不会做题，不要担心。所有的题目都是一个学习的过程，不能保证所有的类型你都接触过。学习历史的CTF赛题只是为了培养一个学习过程/思路。比如遇到不明白的知识点是不是应该从官方文档去入手，看看能不能利用，亦或者是在本地去慢慢调试，看看是不是有没有注意到的地方。 即使最后真的没做出来某个题目，但是这个思考/搜集信息的过程也是进步的一部分。可能在解题思路出来的那一刻，你会恍然大悟(这个点我注意到了，但是我没太关注 / 这个真巧妙) 之类的，慢慢复盘，才会在拿到一个题目时有事可做。

Pwn – 二进制安全

基础前置知识

- 先参考下面 Re 的基本知识

- Pwn 环境搭建（常用的工具，虚拟机环境或wsl环境 参考大佬博客）
- 基础的 C 语言知识（参考 bilibili）
- IDA 的基本使用（动手 F5 试试，修个结构体啥的）
- 函数调用约定（Google）
- 基本函数调用时的出栈入栈流程（Google）
- 基础的调试技巧 （参考 csdn 和 100个gdb小技巧）
- 熟悉linux常用基础命令(常用的那几个)
- x86汇编基础指令(看懂就行)
- python简单使用(pwntools,zio,LibcSearcher...)
- linux程序保护机制

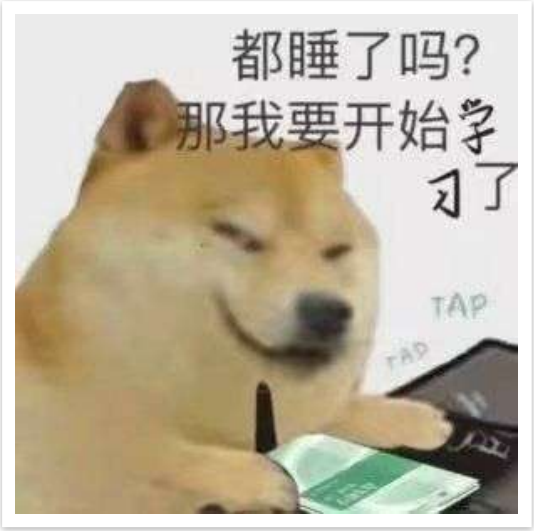
https://ctf-wiki.org/pwn/linux/user-mode/stackoverflow/x86/stack-intro/
https://www.bookstack.cn/read/CTF-All-In-One/SUMMARY.md
https://www.yuque.com/hxfqg9/bin
https://elixir.bootlin.com/glibc/glibc-2.12.1/source

一些碎碎念

- 当学完基础知识之后就可以着手做题了，**不用等全部知识都学会再做题**。一个是Pwn深度还是有的，不怎么学得完，二是容易劝退XD。基础知识的学习是重要且必要的，**忌**一味追求快。学长们都乐意帮你解决疑难，只要**认真学**速度一定不会慢的。
- 学习Pwn还可以一个的方法就是边练边学，以赛督学，发现自己有知识点欠缺就知道学了hhh。
- 初始阶段可以去 buuctf 做点高解题目（基本都是栈，知识点比较少）。
- 当发现自己栈掌握程度比较可以了之后就可以尝试入门堆了，堆是能否进行进一步Pwn学习的一个**分水岭**。入门堆可能确实会比较吃力，尤其是现在 glibc 的**“三年堆题，五年模拟”**现状，还是希望大家找到其中的乐趣，培养兴趣，一步步的进步。
- Pwn 的学习路线陡峭且有意思，在学习完基本的 glibc 之后，可以把目光放在 Kernel、V8、IOT 等和以后工作比较有联系的方向上，复现一些实际生活中的 CVE 会很有成就感。

部分练习平台

- CTFTime 国际赛
- BUUCTF 在线评测
- NSSCTF



Reverse – 逆向工程

浅浅入门

宝宝教程 滴水逆向 随海哥开启逆向之旅(汇编、PE、od调试)
https://www.bilibili.com/video/BV1w54y1y7Di?spm_id_from=333.999.0.0&vd_source=5fae2dd56282a3ba712f67016e54a997

书的话确实要先学习c/c++，之后可以学前九章的<<王爽汇编>>, <<逆向工程核心原理>>和 <<加密与解密>>，可以当做字典来查，按需求学某一部分。至于<<从0到1>>和<<CTF训练营>>可以刷一定题后再读一遍，学习作者的思考方式。
刷题: buuctf前三页、ctftime水比赛
工具: 熟练使用IDA，动调OD和X64DBG

入门建议看的书(只需要一点c语言基础就可以直接看)

逆向工程核心原理
https://share.weiyun.com/2wu8kDlq

来自 qsdz 的学习建议

先把 C/C++ 学到熟练运用，理解指针和结构体的妙用（例如 void* 指针的用处、不定长结构体），理解 C/C++ 变量与内存的关系（即常称的栈与堆究竟在内存中是怎么样的），了解基本的函数调用规则（例如什么是 fastcall、stdcall）、认识编译器 gcc（主要是 gcc 的用法，例如怎么生成调试信息、怎样生成汇编代码、为什么使用不同的参数（-O3、-O0）生成的代码大小不一样）、认识调试器 gdb（主要是会使用基本的下断点、单步调试和查看汇编等的功能）、认识不同语言的差异（比如说C++、Java和Python的差异性）

> ps: 俺学 C/C++ 的时候是使用 notepad++（作者那时候还没有明示自己的政治取向）和 MinGW，希望大家不要盲目遵从网络上或者教学视频中的编程工具，找到自己喜爱的工具比较好。
> 推荐的 IDE 有：Visual Studio、CLion、Sublime
> 推荐的编译器有：Linux 平台下的 GCC、Clang
> 推荐的编辑器有：Visual Studio Code、~~Notepad~~

Crypto – 密码学

密码学本质上是一个数学的深入方向，若想密码学方向发挥良好，无他，第一步先尝试了解一下《数论概论——华章数学译丛》中的各种数论基本概念（仅需了解，无需证明）

> 《数论概论——华章数学译丛》PDF下载
> 链接: https://pan.baidu.com/s/19ZgGCMhQ3ZHCXh7KoKqIlg 提取码: t9m8

譬如小考一个知识点（与有限域开根知识点相关的），如何判断某个数在域 $GF(p)$ 可开根：
高次剩余定理（当满足 $n|(p-1)$ 时即可）
（假如说能有算法基础和数学系基础就更好啦！）

First at all

顾名思义密码学就是研究加密的学科。

古典密码一般凭借脑洞对明文进行替换、位移等操作而使明文变得不可阅读，但这种加密算法只要暴露，也可通过人工暴力尝试进行解密。
现代密码学就要求在加密算法公开的情况下，也能够成功的完成加密解密操作。因此，现代密码学引入秘钥，只要不知道秘钥，就无法对信息进行加密或解密。所以密码学就是要寻找一个在不知道密钥情况下无法破解的算法。
在CTF比赛中，Crypto题目一般会给出加密脚本以及密文，你需要找到最简单且正确的算法来对密文进行还原，当然也有可能是一个在线的加密系统，你需要找出服务器的漏洞，从而攻破服务器！
哦对了，如果你觉得我说的太难懂，不妨去这里看看，我相信你会回来的（笑

数学基础

如果你曾经参加过数学竞赛，获得过不错的成绩或奖项，那么就可以直接跳过这一条，或者你想挑挑刺也可以，哪里说的不好尽管去锤那个Dawn_whisper。密码学是数学的一个应用学科，最早的公钥密码算法RSA就是基于数论而创造的，因此学习密码学需要从数论开始学起，但是密码学并不限于数论，公钥密码往后发展的过程中，也逐步用到了线性代数与抽象代数的内容。其次，最早不是基于数学的块密码，在发展的过程中，也被运用数学的语言来描述，从而更能够更清晰的找到攻击方法。所以，学习密码学会涉及到大量的数学知识，欢迎对数学感兴趣（至少不讨厌）的同学来钻研学习。

编程基础

如果你是Python&C++巨犇，建议直接跳过这一条，当然依旧欢迎挑刺。现代密码学比古典密码复杂许多，它的加密解密算法不是人能够口算或者笔算出来的东西，所以需要编程来解决（电脑比你算的快得多）。密码学由于经常要用到非常非常非常非常非常大的数字，如果你对C和C++有一点了解，一般情况下单就这个数字就会造成溢出，因为它远超C和C++的long long int的上限，因此一般使用python编写程序。而python是一个较接近自然语言的编程语言，非常容易上手，灵活运用搜索引擎以及网上的教程很容易就能学会。

英语基础

8说了，就一句话，看不懂题神仙来了也没辙 qwq。

Summary

密码学不需要没有web、pwn、re那些花里胡哨令人头大的软件（啥也不会瞎说的），只需要一个能开 机的电脑、你聪明的脑袋瓜和勤快的双手就足够啦~

Misc – 安全杂项

图片隐写

https://dummersoul.top/2021/02/09/Misc%E6%80%BB%E7%BB%93%E4%B9%8B%E5%9B%BE%E7%89%87%E9%9A%90%E5%86%99/
https://ctf-wiki.org/misc/picture/introduction/
https://ctf-wiki.org/misc/picture/png/
https://ctf-wiki.org/misc/picture/jpg/
https://ctf-wiki.org/misc/picture/gif/

流量分析

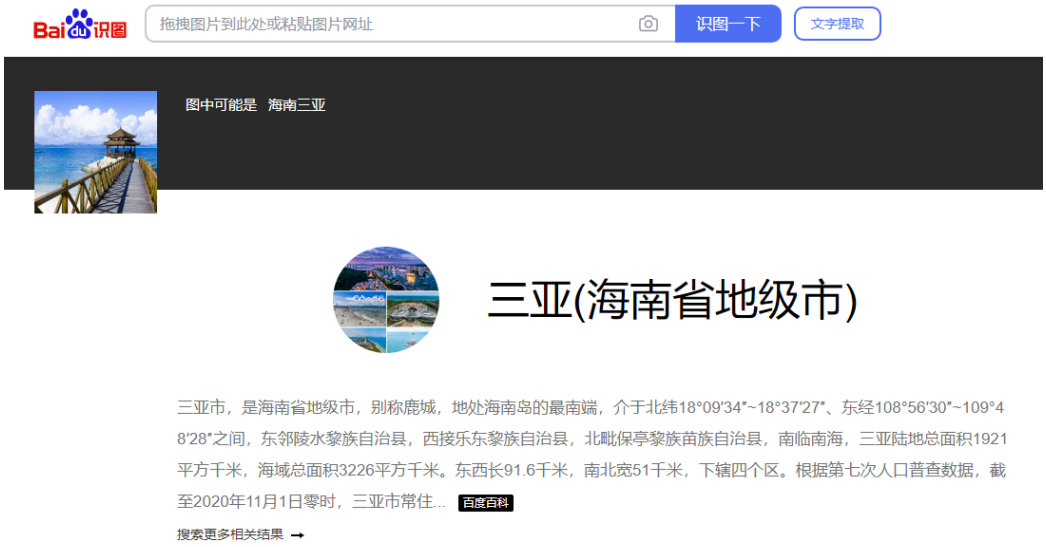
参考文章：

https://dummersoul.top/2021/02/19/Misc%E6%80%BB%E7%BB%93%E4%B9%8B%E6%B5%81%E9%87%8F%E5%88%86%E6%9E%90/
https://xz.aliyun.com/t/1979
https://ctf-wiki.org/misc/traffic/introduction/

OSINT

OSINT， Open Source Intelligence，即开源网络情报，一种情报收集的手段，从各种公开的信息资源中寻找和获取有价值的情报。
这个公开的信息资源主要是指互联网上的资源，有时候也可以是社会工程学。
一般来说互联网上的资源可能是从社交媒体上，又可能是从互联网的记忆中，同时也有可能是从社会资产中（例如政府报告等）。
一般来说需要你擅于使用各种搜索引擎和各种网站的功能，例如百度图片的搜索引擎、谷歌图片的搜索引擎等，又或者是谷歌地图、推特、微博等等。
现在你已经了解基本概念了，我们来做一题进行尝试：请找出以下这张图片的所在地（精确到市）

碰到图片信息的题目，我们第一步一般是用百度识图或者谷歌识图进行搜索，进行识图后我们便可以得到结果：



好了，你已经完全学会1+1了，接下来让我们尝试搜索一下这座桥在哪里吧！

> 来自 qsdz 的亲情建议，最好不要把 Misc 当成主方向（XD