moectf2023 web 入门指南

Klutton

知识和资源的获取

前言

ctf知识的学习与课内一板一眼式的教书不同，如果你想要获得成长的能力，那么你 **必须** 获取 **自主获取知识** 的能力，如果你是一个初学者，这是一个脱离以前填鸭式应试教育的学习模式，你需要不得不逐步适应新的学习方式， **这样的能力不管在哪个方向，甚至在别的竞赛学科中，都是必要的**

这样的学习方式是受益终生的，不是吗?

从公开资源获取知识和资源

这个目录表是有难易顺序的，因为每一个途径都需要一定的经验和知识才能掌握

- 搜索引擎
  - 优先选择**bing**和**google**
  - 不会就先搜，搜索引擎的速度肯定比管理员回消息快
  - 在上面两引擎信息不足情况下考虑其他引擎
- 人工智能
  - 随着时代的发展，从2023年（笔者确信）开始，无论国内外，语言模型的发展使得我们可以大概地与人工智能沟通获取知识，人工智能不怕累、不怕麻烦，值得重复问一些简单问题
  - 如果不知道怎么弄，请接着看下面的内容
- GitHub等开源社区
  - 搜索关键词，可能有热心的开源作者汇集的一些*某某大全形式*的攻击载荷，忘了就去翻翻嘛
  - 例子：
    - PayloadsAlltheThings —— payload大全
    - vulhub的docker（需要会使用搭配相应漏洞知识）
  - 如果觉得你访问的仓库有价值，不妨给仓库一个一个star

从私下获取资源

有些知识在互联网上传播是有风险的，例如*怎么上GitHub，我老是上不去*，不妨私下问问你的室友如何解决 ~~或者你可以陪我跑跑出吃吃饭的时候聊一聊~~

例如：chatgpt不允许国内访问；openai的账号注册需要外国手机号码

那你可以找朋友问一问，借一借

从私下获取知识

那就和问老师问题一样嘛（而且老师还比同学多出来一个教人的义务），主要几点：

- 不要问“在吗？”一类的字眼，言语谦逊地完整提出问题再等待答复
- 被提问者不一定了解具体的细节，要及时补充要求的具体内容

- 如果被问者也不知道，请原谅他
- 如果认为受益匪浅，不妨给出物质上的答谢

综上，遇到不会的怎么办

- 把你认为是特征内容的复制下来
- 扔给上面提到的资源

如何练习

一些练习场

- buuoj
- bugku （可以看三哈师傅的网站了解更多平台）https://www.su-sanha.cn/platforms/

需要知道的：**靶场分普通的ctf靶场，有的是awd或者awdp，有的则是给你一个虚拟机自己从头到尾拿下系统权限（vulnhub）最前者是最基础的，如果想要了解完整真实的渗透流程，可以去尝试vulnhub这类网站**

学会查看题解write up（wp）

在平时的练习中，肯定会遇到不会的题，千万不要死磕，不会就看wp，下一道更好！ 在搜索引擎或者开源社区查找wp，在看wp的过程中，补充并且掌握自己的不会的知识点

如何配置环境

一定要有耐心！ 一套环境不是十分钟二十分钟就能配完的，对于新手而言，几天甚至一两周都有可能，反复地确认自己的步骤、教程的日期、教程使用的系统版本是否正确；

尽量选择官方网站的配置方式：寻找document字样去看文档，这是最稳的方法，但是比较费力； 可以去看私人的文章，但不一定能成功

熟能生巧

可以自己尝试配置一套lamp环境，或者配一台虚拟机，运行一个docker

一些知识清单

编码

在网络传输中，数据被编码成为二进制内容经过网络传输之后解码，由于在这个过程中应用了不同的规则，只有使用了相应的编码才能正确地解析内容，有很多编码形式需要你了解（正经和不正经的），在深度了解编码规则后，你甚至可以自创独属于你的编码规则！

可供参考的资料：https://www.cnblogs.com/ruoli-s/p/14206145.html

编程语言

当提到计算机，不可避免地需要了解编程，事实上，已经存在相当多主流的编程语言，有一些适用于提供和处理web服务，例如python, java, php, golang, rust，你需要具备对于这些代码最少有审计能力（看得懂在做什么）

工具

够用：

- postman
- burp
- sqlmap

计网知识

- 标准的OSI七层模型（其实了解tcp足矣）
- 了解 "协议"（例如http，https，ftp，gopher，也需要了解php伪协议）

HTTP

- 请求方法（get/post）
- 请求内容（json/raw/binary）
- 请求头（content-type/xff/referer/cookie）

服务框架

有现成的网络框架用于高效并发处理请求，所以你需要了解函数是干嘛用的，怎么处理请求的

- python: Flask, Django, Tornado
- java: Spring Boot

用户凭证

http是无状态协议，因此需要储存处理用户信息，你需要了解：

- cookie
- session
- jwt

有时会涉及到用户信息伪造，例如flask框架下的session伪造，通过获取（或者弱口令爆破）secret_key来伪造一个session，通常需要自己写脚本（或者获取GitHub现成的脚本改一改使用）

数据储存

实际上，大部分持久化信息是储存在数据库的，有的出题人为了简化流程也会直接把信息储存在内存里

数据库管理系统（DBMS）

你需要知道去哪儿了解不同DBMS的sql语法，以便完成相应的挑战：

- sql injection注入：查询脚本由于过滤不严导致查询语句可控，用户得以访问数据库
  - 有回显
  - 盲注
- 文件读写：通过数据库进行文件的读写，可以配合起来getshell

（sql注入可以尝试sqlmap一把梭，前提是你找对了注入点）

反序列化

反序列化一直是一个离不开的话题，大体来说，序列化就是把内存里处理好的数据变成二进制内容（通常）用来传输或储存；反序列化就是按照序列化的规则的把内容读取到内存中参与程序运行；如果反序列化不得当，被恶意构造的内容就可能导致机器被"骇入"

- xxe（xml外部实体攻击 libxml 2.8.0以下）
- php反序列化（包括了解其中的魔术方法）
- pickle反序列化（可能会有r指令过滤不过涉及出题不多）
- java反序列化（pop链）

CMS

上网查吧（杂

内网渗透

代理工具

配置socks代理，继续进行内网渗透

- nps
- frp
- proxifier
- proxychains

扫描工具

发现内网资产

- nmap
- fscan

结束

祝你能玩得愉快，学得愉快!

**666c61673d6257396c5933526d6533637a62454e7662575666564739666257396c513152475831637959
6c396a61474673624756755a3055684958303d**