网络安全教学平台的实现

专业：计算机科学与技术　　年级:2015 级　　学生：卓长瑜　指导教师：池灵达

摘　要

随着互联网的高速发展，网络安全问题也不断显露出来，网络空间已被视为海、陆、空、天之外的国家第五主权空间，网络主权是实现网络安全的一个大前提，而网络安全又为网络主权奠定了一个良好的基础。网络安全所面临的巨大挑战，当前的网络系统不够健全和完善，而现在越来越多的人依赖网络去完成大部分的事情，与此同时又没有较高的网络安全意识，导致网络安全问题层出不穷。对于我国网络安全问题的维护是一个漫长的过程，需要所有人去重视并且学习相关网络安全知识。只有全社会真正意识到网络安全的重要性，才能形成网络安全的氛围，从而达到解决网络信息安全所存在的安全隐患的目的。

结合上述内容，设计和开发一套网络安全教学平台是非常有意义的一件事情。之所以采用这种类似竞赛的方式的教学平台，是因为这种模式有利于调动大家学习的积极性，这种模式还能让使用者深刻理解网络安全问题与程序设计密切相关，让学习者感受到网络安全面临的严峻挑战，让学习者能够从平台学习到网络安全基础技术。

本课题以网络安全技术为核心，使用Python语言和Flask框架实现与用户交互的网络安全教学平台，同时采用Docker技术将在线练习独立封装，保障了系统的安全性。

在线练习是本课题的核心内容，以WEB基础技术、PHP漏洞基础、SQL漏洞基础、文件上传漏洞基础四个模块进行题库资源开发，每个模块配以多道题目，每道题目有其目的和意义，让学习者循序渐进的掌握网络安全基础技术。

关键词：信息安全　　在线练习　　Docker　　Flask

目　　录

1 绪论

1.1 研究背景

网络设计的初衷是为了更加方便的交流沟通、进行信息数据的共享。但随着互联网的高速发展和普及，网民规模的逐渐扩大，网络安全问题逐渐显露出来，2015 年下半年到 2016 年上半年的一年间，我国因垃圾短信、诈骗信息、个人信息等泄露遭受了高达 915 亿元的经济损失。互联网已经深入到社会的方方面面，如网上购物和网络金融，还有涉及到民生的行业，都存在着互联网的影子，网络安全的重要性不言而喻。网络空间安全领域逐渐被视为除海、陆、空、天之后的“第 5 空间”，成为世界关注的焦点和热点。西方国家已经将网络空间安全提升到国家战略高度，我国同样十分重视网络空间安全，网络空间安全专业已经被国家教育部列为一级学科，国家正在储备强大可持续的网络空间安全人才。

1.2 研究的意义和目的

在网络空间大事件中，总有知名企业发生数据泄露等事件，如 Facebook 数据泄露事件，Google+数据泄露事件等等，无时无刻不提醒着大家网络安全的重要性，一次数据泄露将会给企业带来巨大的信誉损失，即使及时修复了相关漏洞，也难以挽回在用户中的形象。由此可见，网络空间安全相关的研究是非常有意义的，网络空间安全领域的深入研究有时能为企业避免很多不必要的损失，企业也需要高校提供网络空间安全人才。在网络空间安全领域，基础知识是共通的，建制一个网络安全教学平台，通过练习的方式帮助学生学习网络安全的基础知识，帮助高校培养网络空间安全人才，让学生有厚实的基础知识来面对网络空间安全领域的严峻挑战是一件非常有意义的事情。

本课题通过研究网络空间安全现状，建制一个网络安全教学平台，帮助学生学习网络空间安全领域的相关知识，通过网络安全教学平台，让学生了解网络空间安全领域面临的各种安全挑战，是本课题的主要目的。

1.3 研究的主要内容

本课题主要是通过 Python 语言，在 Flask[1]框架下进行开发，同时利用 Docker[2]容器，保障服务器的安全，避免因题目而导致服务器权限的丢失。可以将内容主要分成以下两部分：Web 系统的开发、Docker 容器题库的开发。Web 系

统的开发主要是实现用户管理、公告管理、题目管理、得分榜管理、提交记录管理、页面管理、网站配置等内容，Docker 容器题库的开发主要是针对在线练习，题库主要划分为 WEB 基础[3]题库，PHP 基础题库[4]，SQL 注入基础[5]题库，文件上传漏洞[6]基础题库等，每种类型的题库下有多道题目，每道题目考察不同的知识点。同时还要研究 Nginx 分发策略，通过 Nginx 将 Web 访问数据分发至 Web 系统，将在线练习访问数据分发至 Docker 容器内部的练习题中，还需要编写脚本管理 Docker 容器，每个 Docker 容器都需要为其设置固定的 IP 地址与 Nginx 分发的 IP 地址对应，需要自定义 Docker 容器的网卡配置等，如何将 Web 系统和 Docker 容器衔接起来是本课题需要解决的问题。

2 环境搭建

2.1 系统环境搭建

本项目编写的代码主要是在服务端运行，这里选用 Linux 作为运行环境，Linux 有众多发行版本，Ubuntu 是一个较为流行的 Linux 版本，由于没有服务器用于搭建本项目环境，因此本项目使用虚拟机拟建一个服务器环境，这里选择 Ubuntu18.04LTS 版本，首先从官网下载 Ubuntu18.04 版本的系统镜像，使用 VMware Workstation 软件进行虚拟机安装，创建 Ubuntu 虚拟机镜像，待系统安装完成之后，配置 Ubuntu 的源和设置系统时间等，为后续操作提供保障，总的来说，需要安装的和设置的内容如下：

（1）安装 VMware Workstation；

（2）下载 Ubuntu18.04LST 镜像；

（3）安装 Ubuntu 系统镜像；

（4）配置 Ubuntu 源和时间；

2.2 PYTHON 环境搭建

无论是 Python 开发环境还是 Python 运行环境都需要安装依赖库，首先确定 Ubuntu 中的 Python 版本（现在大部分 Linux 系统默认带有 Python 环境），使用 Python --version 命令查看 Python 版本，Python2 和 Python3 有较大的变动，本项目需要运行在 Python3 环境下，因此若得到的 Python 版本为 2.x，则需要通过 sudo apt-get install python3 命令安装 Python3 环境，在安装了 Python3 环境后，还需要安装一些依赖包，由于依赖包内容较多，已将依赖包写至 requirements.txt 文件中，只需使用以下命令即可进行安装：

```
python3 -m pip install -r requirements.txt
```

2.3 DOCKER 环境搭建

Docker 容器官方提供了安装脚本没直接运行即可安装最新的 Docker 环境，这里为了稳定性，选择使用 Ubuntu 提供的 Docker 镜像版本，虽然不是最新版本，但是足够满足本项目的运行了，安装方法如下

使用 sudo apt install docker.io 命令安装 Docker 环境，再使用 sudo systemctl start docker 将 Docker 容器启动，最后使用 sudo systemctl enable

docker 命令，将 Docker 添加至开机启动项，使用 docekr -v 查看是否安装成功，再通过 docker run hello-world 命令检测 Docker 运行环境是否正常，至此 Docker 环境搭建完成。

3 系统分析

3.1 可行性分析

项目可行性分析是在选择是否做这项目前的重要工作，从技术、经济、和实用性等多个方面进行分析。从安全层面来分析的话，现在已经存在很多虚拟化技术可以实现系统的隔离，平台可以选择 Docker[7]等技术来保证服务端的安全[8]，用户数据处理与练习页面并无太大关联，用户攻击我们提供的练习网站[9]，数据只会出现在 Docker 容器内部[10]，无法通过攻击我们提供的练习页面来达到控制整台服务器的效果，因此安全方面，可行性很高。从经济层面来说，本项目虽然对服务器配置有一定的要求，但是仍在一个可控的范围内，每个练习网站大概占用 100MB 的内存空间，故而从经济层面也固有一定的可行性。从实用性层面分析的话，本平台提供了较多的基础案例，能够帮助学生快速建立对信息安全的一个意识，后续还可针对各个不同的方向提供更多的案例来帮助学生深入的了解信息安全领域的知识，是一个实用性很高的一个项目。

本项目不论从技术、经济和实用性等各个方面都具有一定的可行性，且实现成本相对较低，实现难度适中，是一个可行的项目。

3.2 需求分析

需求分析是对软件进行细致的调研和分析，准确理解项目的功能、性能、可靠性等要求，确认系统要做什么的过程，本设计是为了帮助学生在平台上完成自主学习信息安全技术，学生可以在平台上注册、登录并且进行安全实验，可以进行各种 WEB 相关的实验，如 WEB 基础实验、PHP 基础实验、SQL 注入实验和文件上传漏洞实验等，用户还可以通过平台查看大家的答题情况以及排行榜信息，用户可以查看到公告等信息，管理员可以通过平台创建题目和发布公告，可以对用户进行管理，对排行榜上的信息进行查询，可以隐藏某些用户，管理员还可以通过后台修改首页页面，可将需求细分为以下几个模块。

3.2.1 用户管理

用户的注册登录基本上是所有系统所必备的一个需求，用户管理包括用户的注册、登录、管理等相关功能，用户的注册基本是每个网站必备的功能，本系统

要求注册用户必须输入用户名、密码、邮箱信息，后台管理员可以对用户进行增删查改等操作。

3.2.2 公告管理

公告管理是本系统必不可少的一个功能，该功能主要负责广播通知用户系统信息，如题目的改动，以及推送任何你想要的信息，普通用户只能接受到公告系统所推送的信息，管理员需要有发布公告、删除公告等功能，该功能是本系统的核心功能点，也是用户能够及时获取系统相关信息的保障。

3.2.3 题目管理

题目的管理是本系统的核心功能，该功能负责题目的显示、添加、编辑、删除等等，普通用户只能通过网站查看题目列表，通过颜色的区别，显示出已做过的题目，每道题目有对应的 flag 以及分值，管理员在后台可以对题目进行增加，每道题目都有其归属的类型，显示给用户的题目以类型划分，管理员还可以在后台对题目进行隐藏/显示等操作。

3.2.4 得分榜管理

得分榜管理是对用户得分情况的管理，该功能负责将所有用户的所得分值进行降序排列展示，将得分最多的用户展示在得分榜的顶端，有利于调动用户的做题积极性，该榜显示得分最高的前 10 名用户。

3.2.5 提交记录管理

提交记录管理是对所有用户的所有提交记录进行统一的管理，在前台用户可以提交自己所得到的答案，后台管理员可以查看到所有用户的所有正确、错误的答案提交，同时还将生成一个题目状态的统计数据，即对那种题型解决的人数较多，那道题得分率最高等进行数据可视化，通过柱状图和饼图将数据清晰的展示给管理，方便管理员对题目难度类型进行调控。

3.2.6 页面管理

页面管理主要是用来想用户展示一些东西在首页，如一些标语，一些鼓励的话，或者一张管理员觉得很酷的图片，只要是正常 WEB 能够展示的，都可以在页面管理中添加，页面管理功能仅限于管理员进行设置，其他用户只能查看。

3.2.7 网站配置

网站配置主要是对网站的基本功能进行配置，如创建用户时是否需要邮箱、是否允许用户注册、网站数据的导入/导出、网站数据清空、网站图标的设置等等，该功能主要负责对网站进行一些小的更改，不需要管理员修改代码，只需要在后台进行设置即可。

3.2.8 在线练习

在线练习主要是在服务器提供的练习页面进行训练，该内容是本设计的核心，用户可以通过在线练习系统进行实操训练，增强自己的安全意识，掌握网络安全相关的基础技术，内容主要以 Web 漏洞为主。

3.3 研究过程

研究重点：首先确定开发本系统所使用的语言的框架，通过网络和书籍查询网站开发、信息安全案例开发等相关知识，在确认是使用 Python 语言进行网站设计之后，再研究 Python 相关的 Web 开发框架，最后确认使用 Flask 轻量级 Web 框架进行开发，针对需求进行 Web 开发后，还需要寻找一个方案用来存放赛题和进行赛题的制作，这里选用的 Docker 轻量级虚拟化容器技术，对整个题目环境进行模拟，为了更好的隔离各个题目，避免出现做完一道题拿到其他题目的答案，所以需要为每一个题目都准备一个 Docker 容器，一道赛题大概要占用 100M 内存，还需要编写脚本对赛题进行管理控制，每道赛题需要单独处理其运行环境、数据库等问题，为了提高赛题的可移植性，还要针对 Docker 容器技术进行深入学习，了解其移植方法等，同时还需要对 Nginx 进行深入了解，了解其分发策略，将赛题网址分发到各个 Docker 容器中，同时 Docker 容器的网络需要自定义后才可对其进行管理，不然分配的 IP 地址将会是随机的，因此容器的 IP 地址如何与 Nginx 挂钩是研究的重点。

3.3.1 研究过程中的关键点

（1） 使用 Flask 框架进行网站开发。
（2） 使用 Docker 容器进行赛题搭建。
（3） 使用 Bash 脚本进行赛题管理。
（4） 使用 Nginx 进行赛题分发。

4 系统设计与实现

4.1 系统架构

本系统使用 Nginx 分发用户访问请求，当用户访问的网址是 Web 主站点时，Nginx 将请求分发至信息安全教学平台，当用户访问的是练习站点的时候，根据用户访问的 URL，将请求转发至各个 Docker 容器，Docker 容器内部存在搭建好的可攻击的站点或程序，大部分 Docker 容器采用 LNMP 结构，因此只需要将 URL 转发至 Docker 容器内部即可访问，信息安全教学平台采用 Python 语言+Flask 框架开发，数据库采用轻量级的 SQLite 设计，方便站点的移植工作，大致的系统架构如下图 0-1 所示：

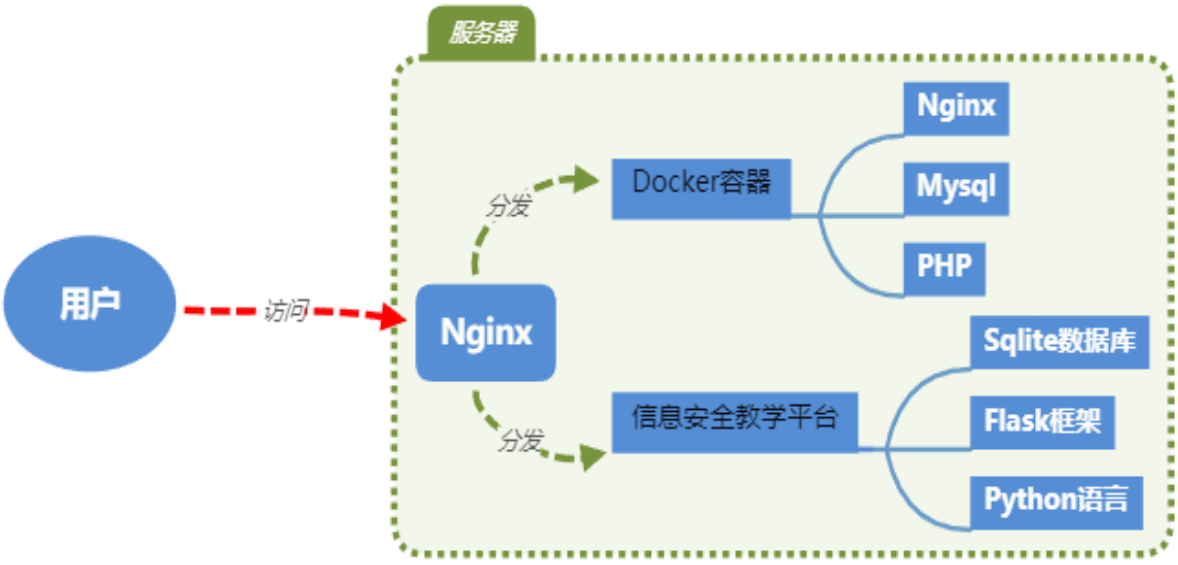

图 0-1 系统架构图

4.2 网络安全教学平台

4.2.1 数据库设计与实现

本课题的核心内容是题目资源开发，平台的设计只是为了管理题目资源，因此数据库部分只将主要数据库表结构展示出来，核心的表有 7 个，用户表、公告表、题目信息、提交记录表、页面管理表、网站配置表，还有个 Flag 表依附于

题目信息，独立出来是为了 Flag 分值计算的拓展。

表 0-1 用户数据表 users

字段名	数据类型	长度	主键	描述
id	int	无	是	用户编号
name	varchar	128	否	用户名
password	varchar	128	否	登录密码
email	varchar	128	否	邮箱
type	varchar	80	否	用户类型
website	varchar	128	否	用户网站
affiliation	varchar	128	否	座右铭
country	varchar	32	否	国家
hidden	boolean	无	否	是否隐藏
banned	boolean	无	否	是否封禁
verified	boolean	无	否	是否验证
created	datetime	无	否	创建时间

表 0-2 公告信息表 notifications

字段名	数据类型	长度	主键	描述
id	int	无	是	公告编号
title	text	无	否	公告标题
content	text	无	否	公告内容
date	datetime	无	否	发布时间

表 0-3 题目信息表 challenges

字段名	数据类型	长度	主键	描述
id	int	无	是	题目编号
name	varchar	80	否	题目标题
description	text	无	否	题目描述
value	int	无	否	题目分值
category	varchar	80	否	题目大类
state	varchar	80	否	题目状态

表 0-4 提交记录表 submissions

字段名	数据类型	长度	主键	描述
id	int	无	是	记录编号
chellenge_id	int	80	否	题目编号
user_id	text	无	否	用户编号
ip	varchar	46	否	IP 地址
provided	text	无	否	提交内容
type	varchar	32	否	提交结果
date	datetime	无	否	提交时间

表 0-5 页面管理表 pages

字段名	数据类型	长度	主键	描述
id	int	无	是	页面编号
title	varcahr	80	否	页面标题
route	varchar	128	否	页面路由
content	text	无	否	页面内容
drift	boolean	无	否	是否下拉
hidden	boolean	32	否	是否隐藏
auth_requeired	boolean	无	否	是否需要登录

表 0-6 网站配置表 config

字段名	数据类型	长度	主键	描述
id	int	无	是	配置编号
key	text	无	否	配置项
value	text	无	否	配置项值

表 0-7 题目答案表 flag

字段名	数据类型	长度	主键	描述
id	int	无	是	页面编号

challenge_id	int	无	否	页面标题
content	text	无	否	Flag 值

4.2.2 平台设计与实现

平台是基于 Python+Flask 框架实现的，使用 Flask 框架的 register_blueprin 实现页面分发，将各个页面的路由分发部分模块化，各个模块负责处理页面访问数据，对页面所传递参数进行解析判断，执行对应的页面数据传递、路由分发策略，该业务层调用服务层的接口来实现数据库的信息交互，服务层负责对业务层传递的数据进行过滤、筛选判断等操作，严格控制传入数据的内容，对带有攻击性的数据进行处理，将处理过后的数据传递给数据交互层，数据交互层负责直接与数据库进行交互，所交互的数据都是经过服务层处理过的可信数据。

所有页面分发策略都放在 CTFd 目录下，Flask 框架的蓝图注册分发代码则在 CTFd 模块的__init__.py 中实现，所有服务层代码均在 CTFd/schemas 下，所有数据交互层的代码均在 CTFd/models 下，其他一些小工具则存放在 CTFd/utils 下，项目主要结构基本如上所述。

最终完成平台如下图所示：

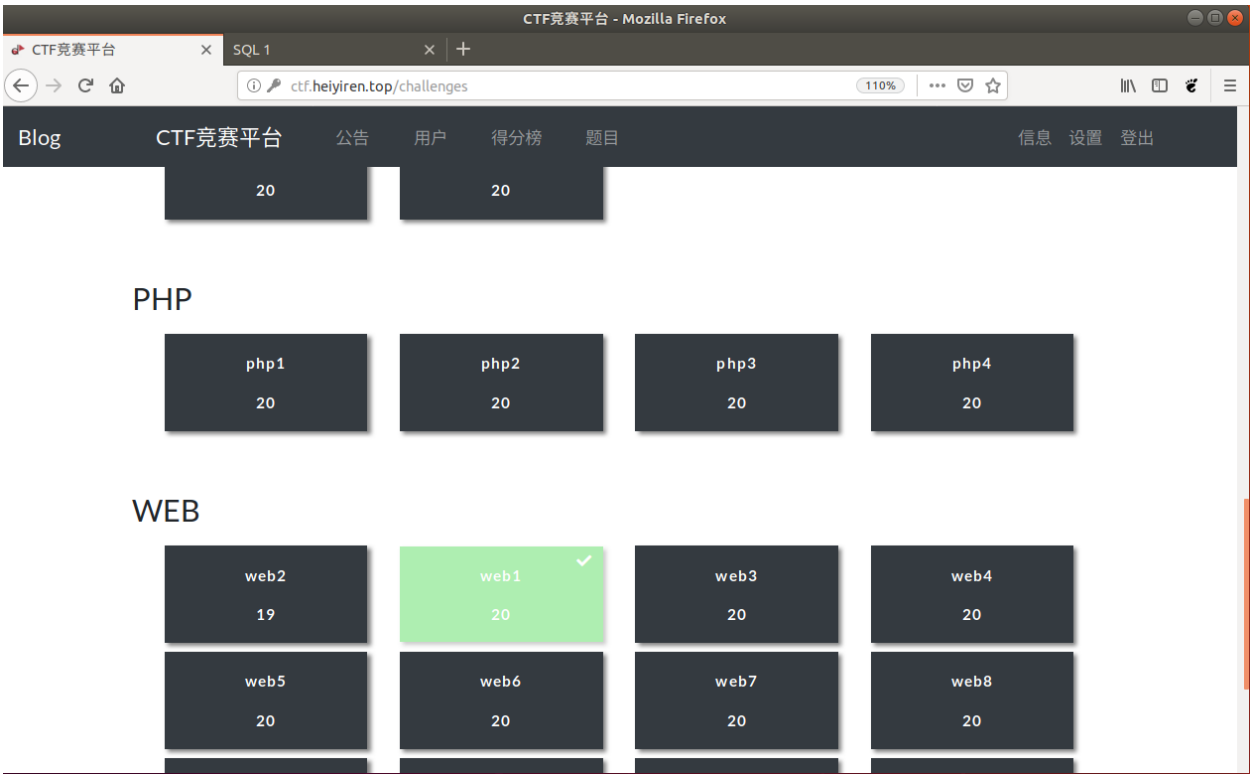

图 0-2 平台效果图

4.3 题目资源开发

题目资源开发是本设计的核心之一，循序渐进的题目才能使学生打下厚实的基础，本设计以 WEB 基础知识、PHP 漏洞基础、SQL 注入基础和文件上传漏洞基础四大类型进行题目资源开发，每种类型有数道题目可供学生练习，每道题目有唯一的答案，只有学生掌握了题目所需基础知识，并且完成了练习部分，才能得到题目的答案，当学生得到答案后提交给系统，系统会将得分累加的学生总分值上。

4.3.1 WEB 基础知识资源开发

Web 类型试题 1：浏览器是进行 WEB 应用程序通信的基础，掌握浏览器的调试开发功能对于网站代码分析有着很大的帮助，因此设计 Web1 用于让学生掌握浏览器的元素审查功能，无论是前端开发还是前端代码审计，都非常依赖于浏览器的元素审查功能，本题将答案藏于 Html 源代码的注释当中，只有在掌握 HTML 语言，知道如何利用浏览器进行元素审查，才可拿到题目的 Flag 值。

Web 类型试题 2：元素审查是 WEB 浏览器的基本功能，有时需要对 WEB 前端代码进行修改，Web2 中通过 maxlength 限制了输入框的大小，需要用户深入理解浏览器调试功能，能够随心所欲的修改前端代码，将 maxlength 的限制去除，才可获得 flag。

Web3：HTTP 协议是进行 WEB 通信的基础，Web3 设计于让学生深刻的理解 WEB 参数传递过程，主要考验用户对 GET 协议的理解，本设计会将代码显示给用户，用户根据代码提示，通过 GET 方法传递指定参数给服务器，服务器才会返回 flag 值。

Web 类型试题 4：HTTP 协议除了 GET 方式传参外，还可以通过 POST 方式传参，通过浏览器无法直接传递 POST 参数，需要安装对应的插件才进行 POST 通信，本题设计于让用户了解 HTTP 协议的 POST 参数传递方式，以及如何利用浏览器插件进行 POST 参数传递，用户只有想服务器 POST 指定参数才可获得 Flag。

Web 类型试题 5：PHP 是一种弱类型语言，由于该弱类型特性，导致参数传递时无法确认其参数类型，本题设计于利用 PHP 类型特性，绕过某些条件判断语句，达到修改代码流程的目的。

Web 类型试题 6：浏览器有时也容易被攻击，如无限 alert 等，当然现在的浏览器以及对恶意 alert 有了一定的检测，可以绕过，本题在恶意 alert 之后还考察查看 HTML 编码格式，将 flag 转换成 html 编码隐藏在注释当中。

Web 类型试题 7：本题设计为了解域名解析原理，通过将 ctf.baidu.com 解析到本服务器 IP 地址，服务器将该域名的解析揭露转发至 web7 应用容器内部才可获得 flag，意图在于让使用者明白 hosts 文件与 DNS 解析的优先级。

Web 类型试题 8：对于服务器中返回的 WEB 数据，同样的 URL 每次返回的结果都有可能变化，网页自身也可通过不断的 reload 让自己产生变化，学会利用浏览器特性或其他工具获取网页某状态页面数据是目的。

Web 类型试题 9：JSFK 是一个 JavaScript 混淆工具，其混淆后的 JavaScript 代码只剩下特殊字符，无任何可读性，本题设计于了解 JavaScript 代码混淆方式，从混淆后的代码中获取重要信息。

Web 类型试题 10：在进行 HTTP 请求是，其实有 HTTP 请求头，头部可以包含一些重要信息，本题设计于了解 HTTP 请求头结构。

Web 类型试题 11：网站后门是黑客在入侵网站时所留下的窗口，本题设计于掌握网页扫描工具的使用，获取到网站后门地址，再通过 BURP 爆破出后门密码，从而获得 flag。

Web 类型试题 12：HTTP 头中有一些数据非常重要，本题设计于让学习者明白 HTTP 头中的 X-Forwarded-For 结构，服务器可以从该头中获取 IP 用户的 IP 地址，用户可以通过构造虚假的 X-Forwarded-For 来绕过服务器某些检测，达到欺骗服务器的目的。

Web 类型试题 13：JavaScript 中支持一种 eval 的方法来实现动态解析 JavaScript 代码，通过该方法可以对 JavaScript 代码进行一些处理，本题设计于让学习者掌握 URL 编码格式，并且掌握 JavaScript 动态解析的方法。

Web 类型试题 14：本题设计于让使用者明白短密码的不安全性，让使用者学会如何通过爆破等方式直接获取网站密码，学会使用 BURP 等 WEB 攻击工具。

Web 类型试题 15：掌握一门脚本语言十分有利于学习网络安全技术，本题设计于让使用者通过脚本语言获得。

4.3.2 SQL 注入漏洞资源开发

SQL 注入是最常见的 WEB 攻击方式之一，通过在参数中传入预先设计好的 SQL 代码，将原有的查询结果替换，获取到数据库的数据或者修改数据中的数据，本类型设计了 8 道 SQL 相关的题型，逐渐增加 SQL 注入的难度，意在让学习者掌握 SQL 注入的攻防部分，由于 SQL 注入所包含的内容相对较广，存在多种注入方式，

如宽字节注入、延迟注入、二次注入等知识，本设计主要设计于演示基础的 SQL 注入方式，以及其攻防方式。

SQL 注入 1：本题设计为最基础的 SQL 常规注入，设计于让学习者明白 SQL 注入的利用方式，通过联合查询等方式获取到表中的 flag 数值。

SQL 注入 2：本题设计为数据水平越权查询，即存在用户 A 和用户 B 两个用户，用户 A 在未经允许的情况下可以获得用户 B 的所有信息，是最常见的数据泄露方式，该题同样可以通过常规注入方式来拿到数据库中的数据。

SQL 注入 3：本题设计同样也是数据泄露，但是对传入的 ID 参数进行简单的处理（加了引号），使得用户构造攻击参数时变得复杂，提高了 SQL 注入的难度。

SQL 注入 4：本题设计于针对用户登录行为，登录要求输入账号密码，需要用户通过某些特殊攻击语句，达到万能密码登录的效果，常见的攻击语句’or 1=’1。

SQL 注入 5：本题在 SQL 注入 4 的基础上进行难度提升，将参数用括号包裹，在用户不知道我们采用括号的情况下，能防止部分攻击探测行为。

SQL 注入 6：本题设计于 SQL 注入非查询语句的情况，在无返回结果的 SQL 语句查询中，可以利用 SQL 语句中的 Sleep 方法来让页面有不同的相应时间，通过脚本语言分析响应时间从而穷举出网站数据库中的数据内容。

SQL 注入 7：本题设计于让使用明白 SQL 注入的多种方式，传输传递并非只能通过 GET 和 POST 传递，还可以通过 HTTP 头中的某些数据来传递参数，因为服务器中可能通过代码读取 HTTP 头的参数作为 SQL 语句中的参数带入数据库查询语句，$_SERVER['REMOTE_ADDR']是常见的服务器获取 IP 地址的方式，通过构造带有攻击行为的 REMOTE_ADDR 头数据，达到从服务器获取数据的目的。

SQL 注入 8：为了长期存储用户数据，服务器可能会将数据存放在浏览器的 cookie 当中，当服务器从浏览器得到 cookie 数据时，若未加校验，会导致带有攻击参数的 cookie 数据被带入 sql 语句中执行，本题设计于让用户了解 Cookie 注入方式，通过 Cookie 注入从服务器获取数据。

4.3.3 PHP 基础知识资源开发

PHP 是目前 WEB 领域最常见的开发语言，由于其开源、免费、高效等特性，受到了广大开发者的喜爱，同时该语言也存在很多问题，对类型的处理以及某些函数容易受到攻击，如 MD5 函数对数组的计算以及正则表达式的回溯等等，本类型主要用来让用户了解常见的 PHP 漏洞，以及引发对 PHP 安全的一些思考。

PHP 类型题目 1：该设计于让用户掌握 PHP 中最经典的一句话木马函数 eval，通过参数传递修改 eval 函数的参数，达到执行任意代码的目的。

PHP 类型题目 2：该设计是为了让用户掌握 PHP 中正则表达式的使用以及全局变量 global 的作用，通过输出全局变量，获取所有变量中的数据。

PHP 类型题目 3：该设计是为了让用户掌握 PHP 中较为常见的文件包含漏洞，通过构造特定的数据，让 PHP 读取到我们指定的文件信息，如 PHP 源代码等，该漏洞因开发者未对 readfile 的函数进行参数校验引发。

PHP 类型题目 4：很多网站因备份文件泄露源代码，本题设计多个考点，一是要求通过扫描工具发现网站备份文件，二是通过备份文件，阅读代码后利用 PHP 中 MD5 比较漏洞绕过数据校验，从而获取到 flag。

4.3.4 文件上传漏洞资源开发

文件上传 1：该设计为对数据进行任何校验，可直接上传一句话密码，获得服务器权限，通过菜刀等工具或直接通过一句话木马获得 Flag 数据。

文件上传 2：该设计在后端对文件类型进行简单判断，$_FILES['uploaded']['type']获取上传文件类型，只能上传 png 和 jpg 文件，但可以构造虚假数据绕过。

文件上传 3：该设计通过对文件后缀名进行识别，判断文件后缀名是否为 png、jpeg、jpg 中的一个，只有该 3 种类型文件能通过判断，可以通过截断上传等方式进行绕过。

文件上传 4：该设计对前端文件后缀名进行识别，判断文件后缀名是否为 png、jpeg、jpg 中的一个，只有该 3 种类型文件能通过判断，可以通过 BURP 抓包修改绕过前端检测。

文件上传 5：该设计与文件上传 2 类似，增加了 gif 图像格式支持，同时降低难度，将代码显示给用户参考。

文件上传 6：该设计使用黑名单机制，对 asp、aspx、php、jsp 等脚本格式进行拦截，并将上传的文件进行文件名随机化处理，大大增加漏洞利用难度。

文件上传 7：该设计同样使用黑名单机制，对.pht、php4、php5、aspx、php、jsp 等 20 多种脚本格式进行拦截，黑名单中增加对大小写的处理，大大增加漏洞利用难度。

文件上传 8：该设计在文件上传 7 的基础上增加了生成文件名随机化处理。

文件上传 9：该设计在文件上传 8 的基础上增加了对上传文件名转化为小写的代码，防止了通过大小写切换绕过黑名单策略。

5 系统测试

5.1 WEB 基础资源测试

由于已开发资源数量较多，测试时只能采取抽样测试，这里选择 WEB5 和 WEB11 进行测试，WEB5 是 PHP 数字解析漏洞，访问页面可以得到网站源代码：

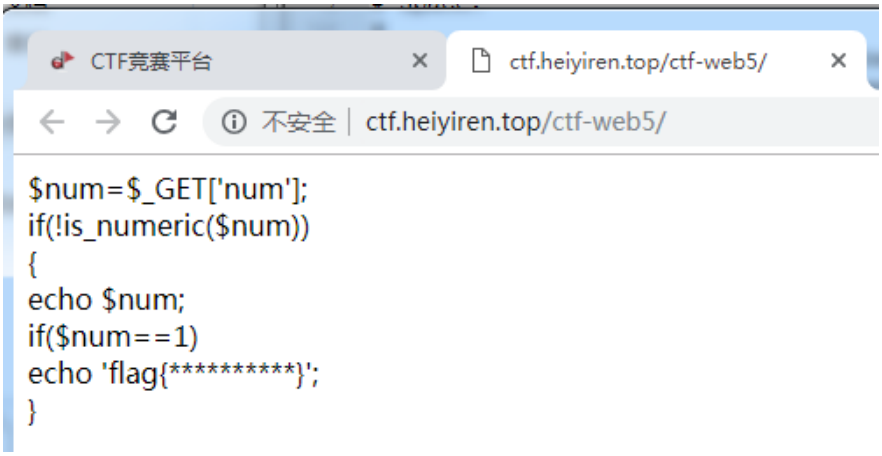

图 0-1 网站代码分析

通过分析网站源代码，可发现 PHP 的双等号判断存在漏洞，当我们传递参数 num=1a 的时候，is_numeric 会将 1a 进行判断，得到 False 返回值，当进行$num==1 判断的时候，$num 会被转化成整数型 1，因此绕过了判断，从而得到 flag，测试过程如图 0-2 所示：

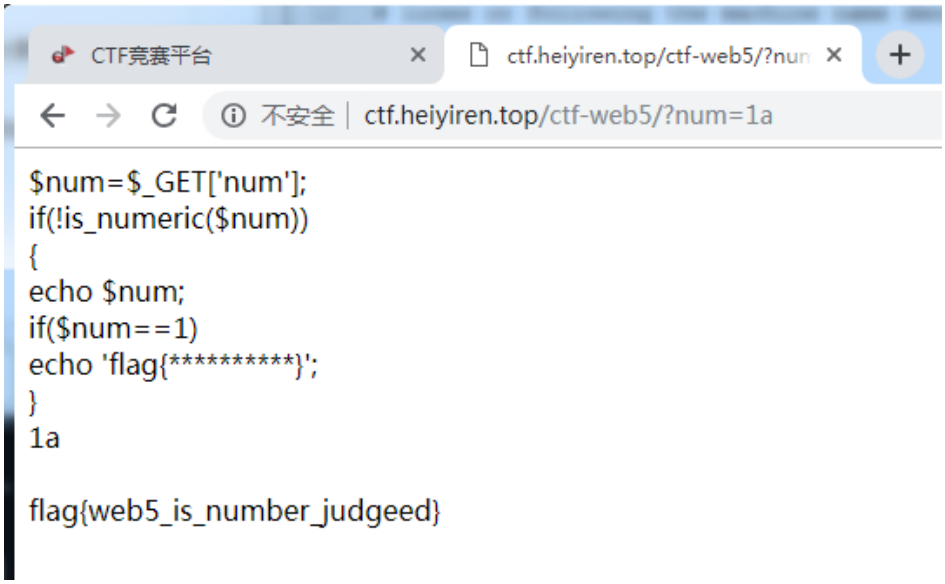

图 0-2 获取 Flag 值

WEB11 是使用对网站进行爆破，首先是通过御剑扫描工具，扫描到网站存在后台木马：

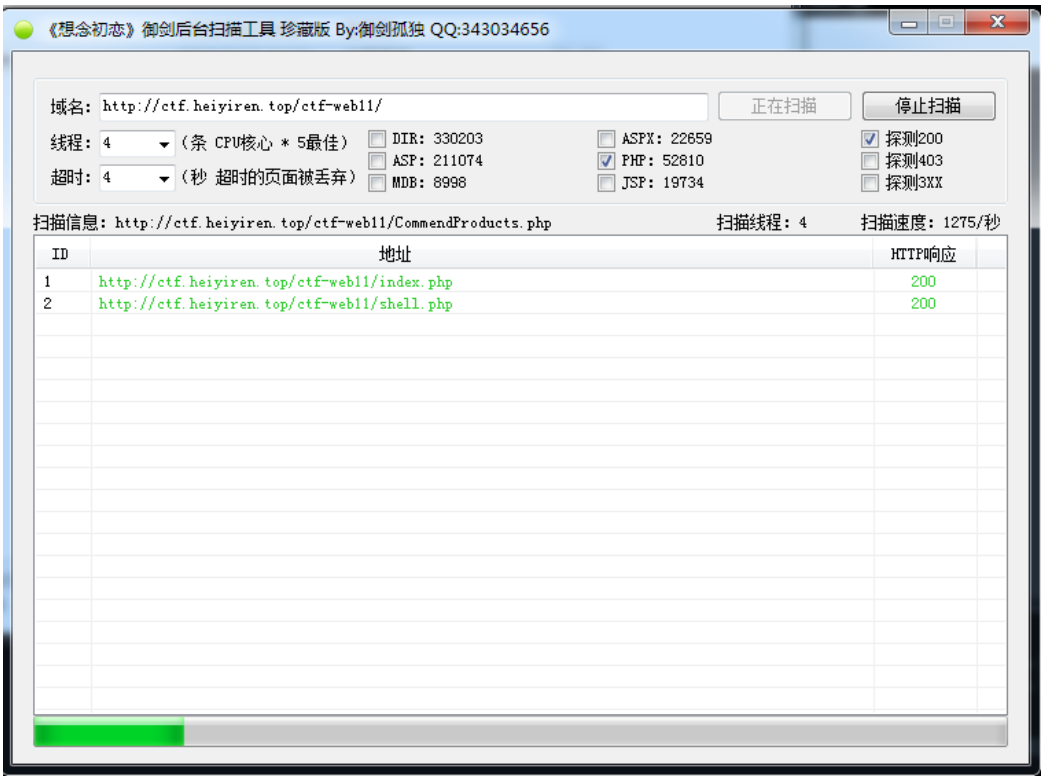

图 0-3 御剑扫描工具

发现网站有 shell.php 文件，进行访问测试，发现后台页面需要密码：

图 0-4 网站后台页面

输入任意字符后会有提示，即要求输入正确的密码才能进入控制页面，这里选择使用 BURP 进行爆破：

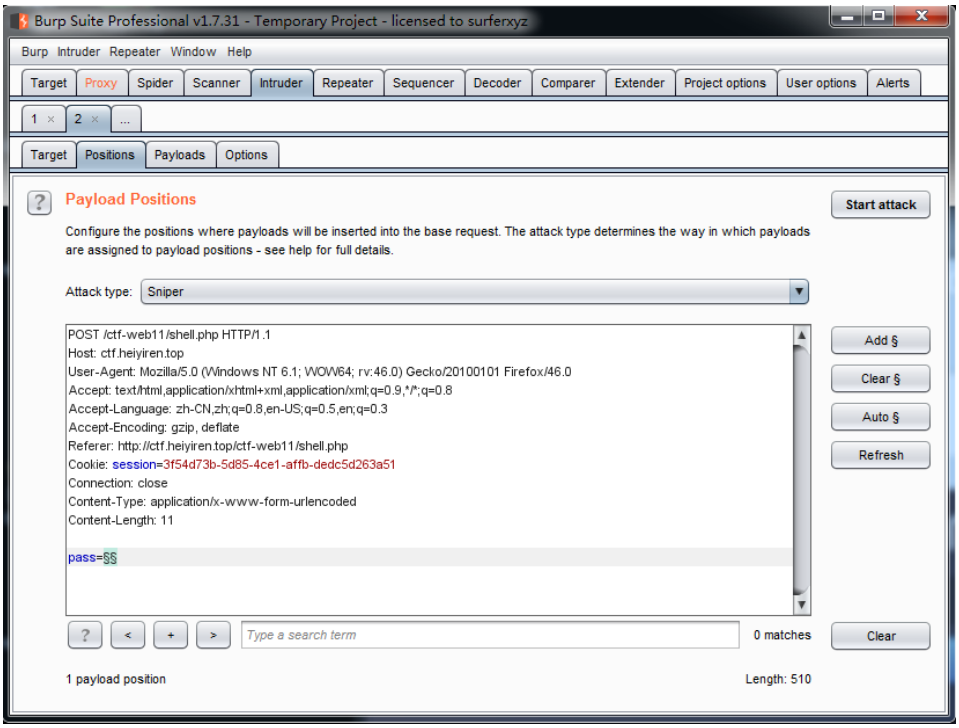

图 0-5 Burp 密码爆破

通过设置爆破字典，来进行密码暴力破解：

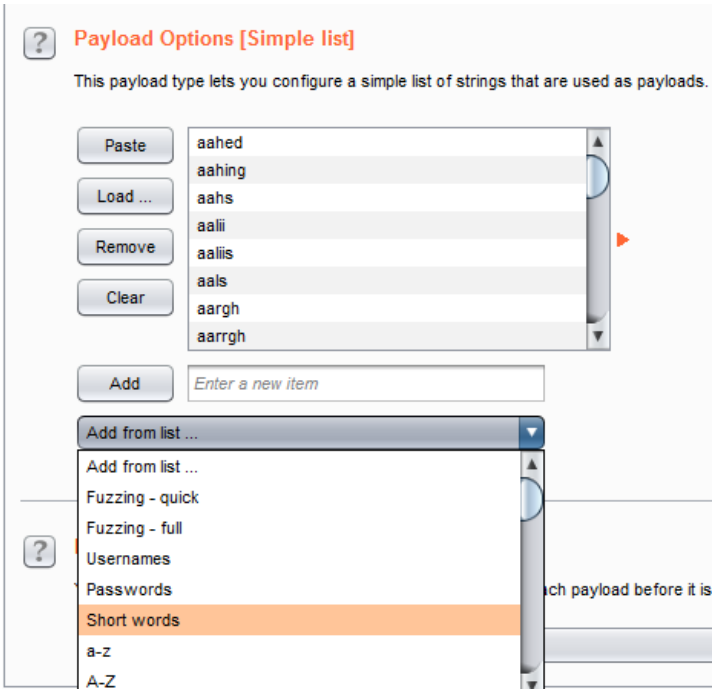

图 0-6 Burp 字典选择

通过返回结果长度比较，判断出正确的密码：

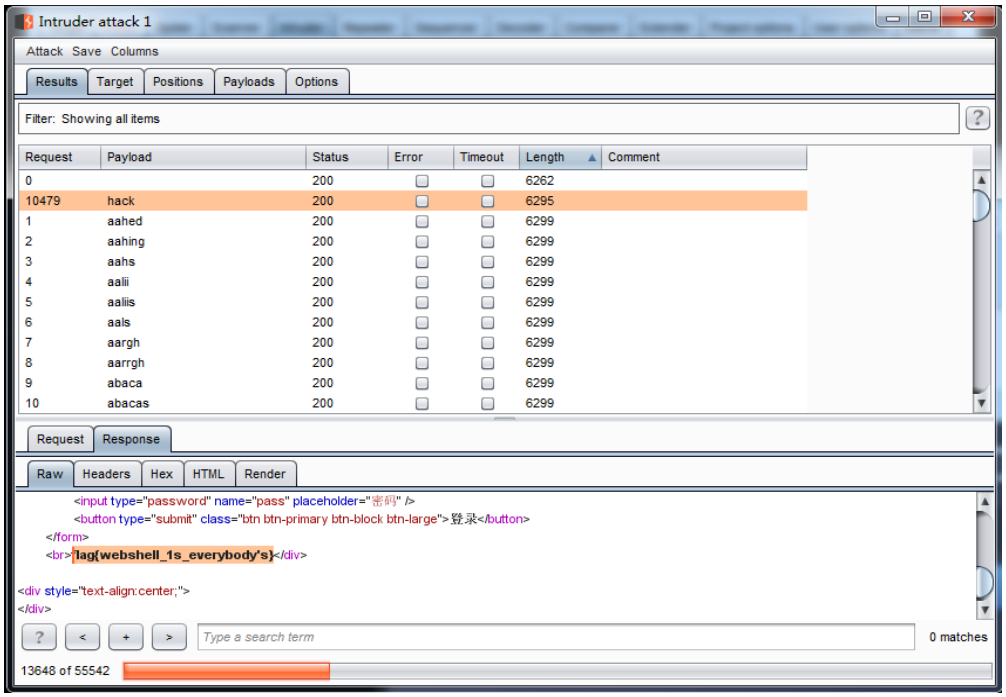

图 0-7 Burp 得到结果

得到密码为 hack,并且得到 flag 值，这里本身是假设存在后台，实际存在不存在后台对本次操作没有太大影响，所以直接给出 flag 值。

5.2 SQL 注入漏洞资源测试

这里对 SQL1 进行抽测，SQL 是基础的 SQL 注入试题，要求对 SQL 数据库有一定的理解才能对其进行攻击，首先使用 1’ and 1=2 UNION SELECT 1,2,group_concat(table_name),4 from information_schema.tables where table_schema=database()# 语句进行攻击，获取到数据库的所有表名:

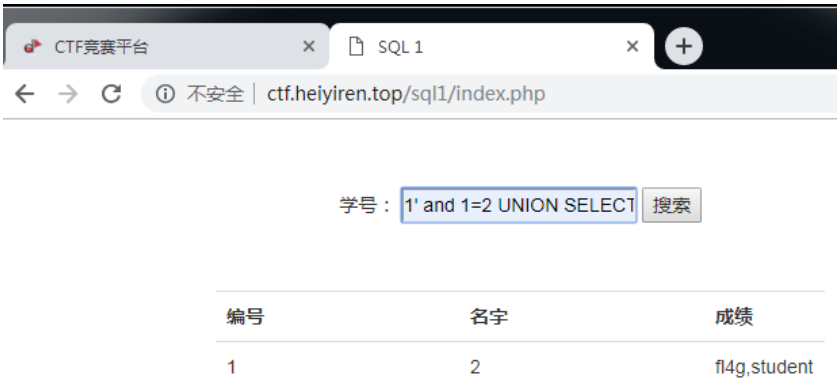

图 0-8 注入获得所有表名

再通过 SQL 攻击语句 1’ and 1=2 UNION SELECT 1,2,group_concat(column_name),4 from information_schema.columns where table_name=0x666C3467# 获得 fl4g 表下的所有字段名：

图 0-9 获得所有字段名

最后通过 SQL 攻击语句 1’ and 1=2 UNION SELECT 1,2,f1ag,4 from fl4g# 获得 fl4g 表下的 f1ag 字段的第一条数据的值：

图 0-10 获得 Flag 值

5.3 PHP 基础知识资源测试

PHP 在渗透过程中最基础的是一句话木马的编写，所以本次 PHP 题型的抽测选择对 PHP1 进行抽测，其通过命令注入，可以实现与一句话木马类似的效果，首先打开目标站点，可以发现有如下代码展示：

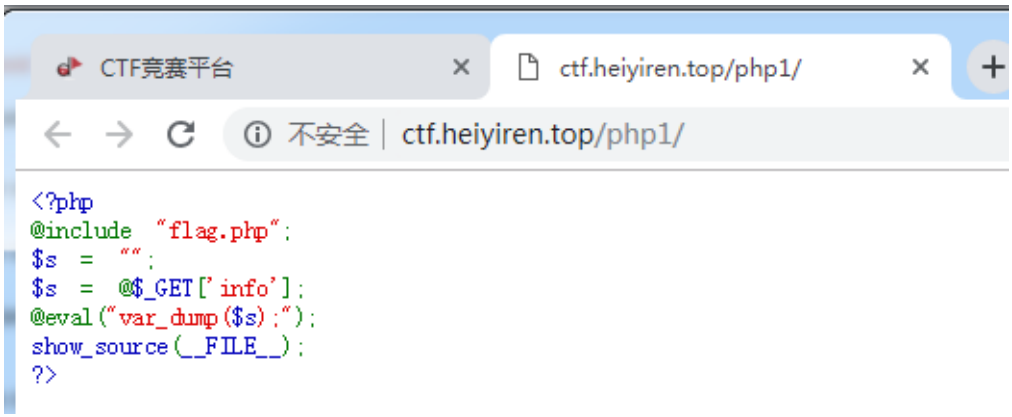

图 0-11 PHP1 代码展示

分析后发现可对其进行命令注入，构造参数 info=1);readfile('flag.php' 截断其命令，将 flag.php 文件读取出来，但由于 php 文件的标签性质，只能通过浏览器的查看源代码可以看见，如下图所示：

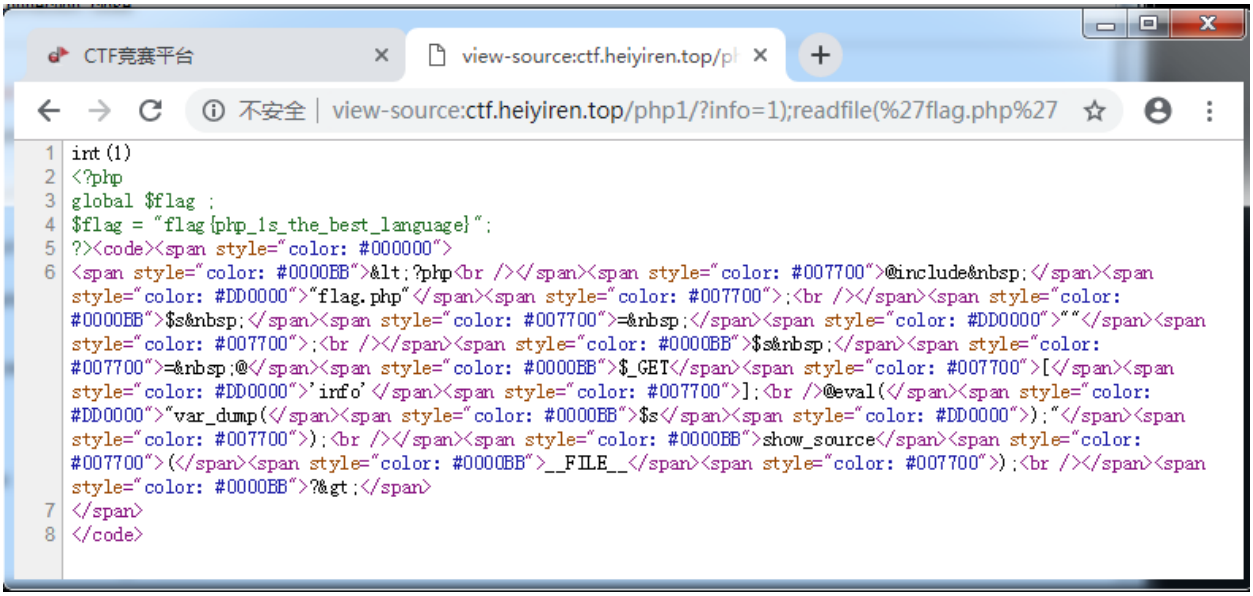

图 0-12 通过命令劫持获得 flag

5.4 文件上传漏洞资源测试

文件上传漏洞在渗透的过程中也十分常见，这里选择最为简单的文件上传漏洞绕过方式进行测试，针对不同的服务器有不同的上传方式，这里的所有上传方式都是通用的，文件上传基础 2 只针对文件上传类型进行检测，作为讲解最为合适，首先打开文件上传基础 2 的练习界面：

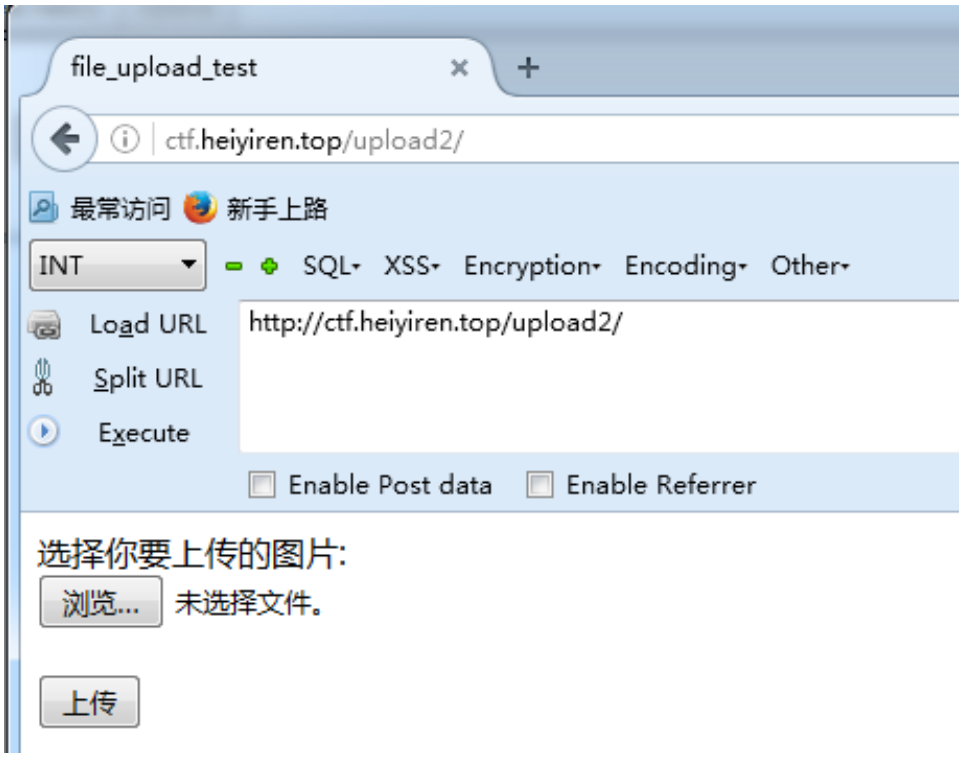

图 0-13 文件上传基础 2

选择一个文件进行上传，这里选择精心制作的 php 文件，并将文件改为.png 后缀：

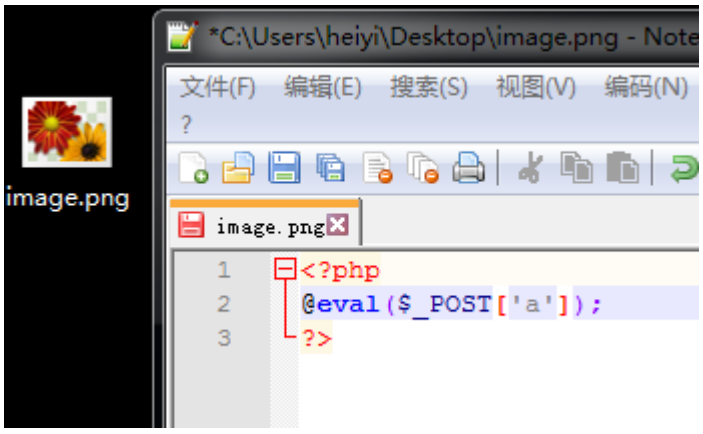

图 0-14 精心制作的 png 文件

通过 Burp 抓包拦截，修改上传文件名的后缀名从而达到欺骗服务器的效果，传递参数时所传递的文件类型仍然是 image/png，服务器获取到的参数也同样是 image/png，但所上传文件的文件后缀却是我们所制定的.php，只要上传了该一句话木马，我们就可以对服务器进行控制，权限相当于 Web 管理程序所拥有的权限，这里将其称之为 webshell。

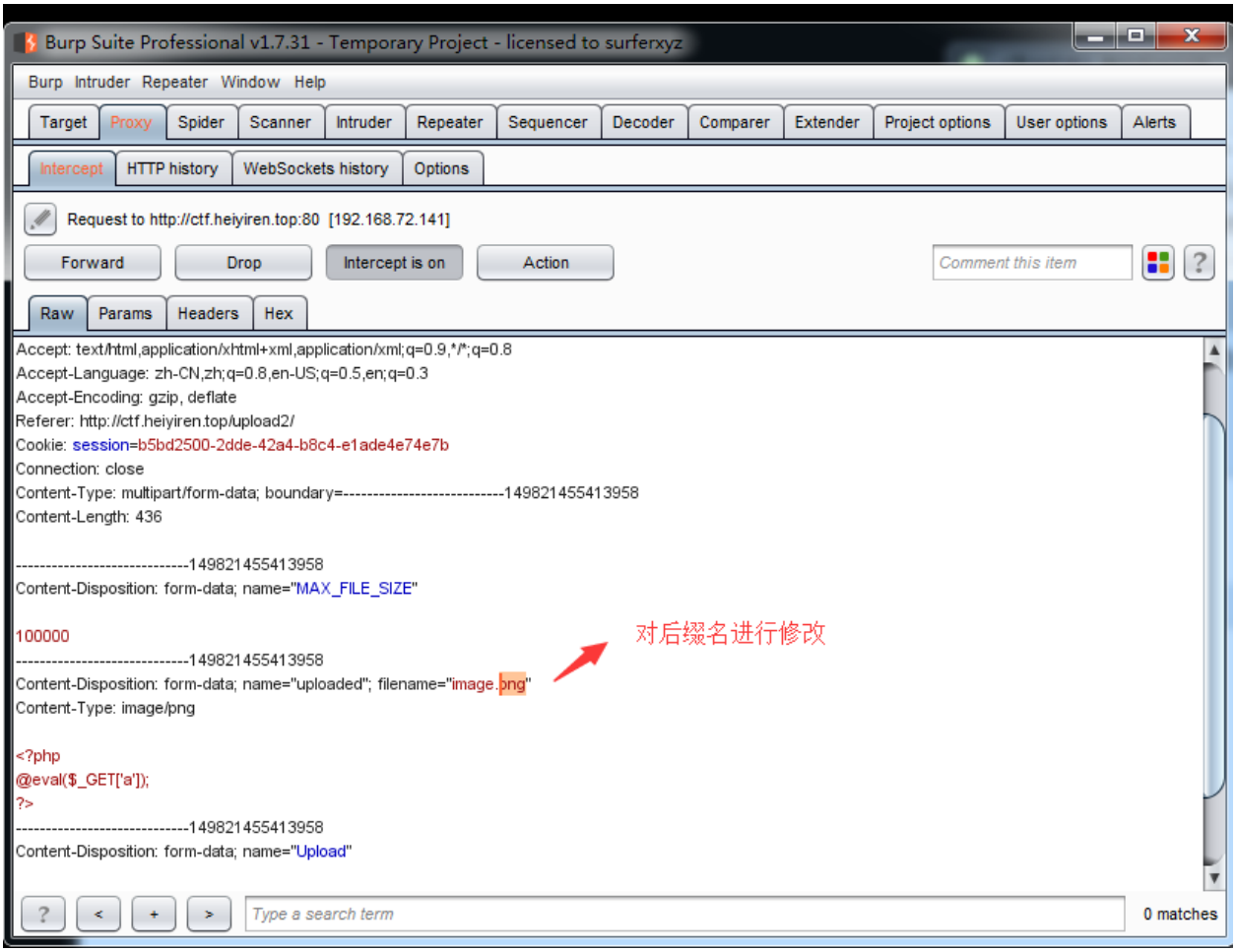

图 0-15 上传一句话木马

此时将会得到提示文件上传成功：

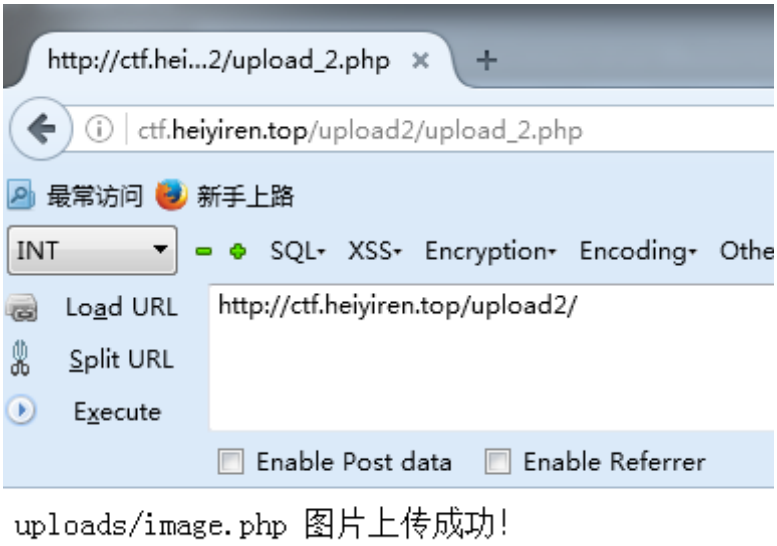

图 0-16 文件上传成功

此时我们已经获得服务器 webshell 权限，通过菜刀工具连接服务终端：

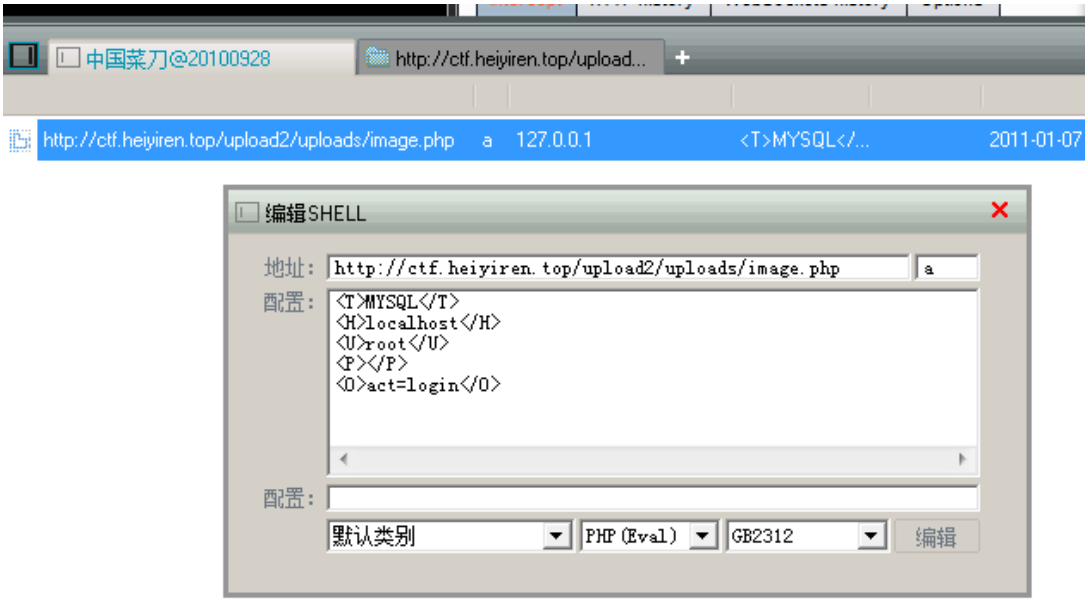

图 0-17 菜刀配置

最后会其网站根目录下的 flag 文件：

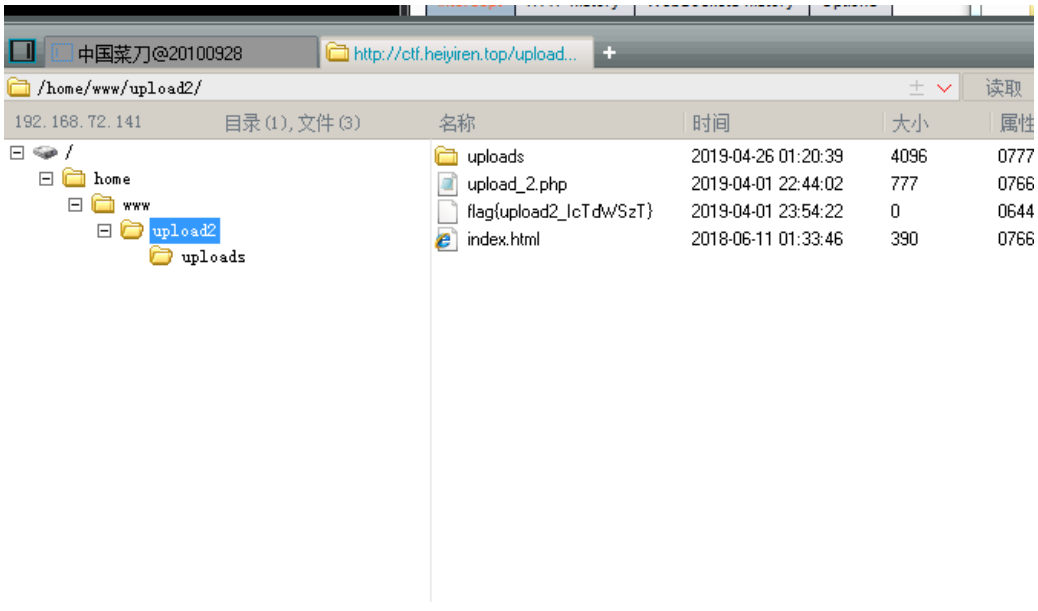

图 0-18 获得网站根目录下的 Flag

结　论

通过对网络安全教学方式的研究，设计并实现了基于 Python+Flask 框架的网络安全教学平台，该平台使用了三层架构模式决定了其可维护性强且安全性高，平台题库则采用 Docker 虚拟化容器进行封装，进一步的保障了平台的安全，平台采取竞赛得分机制，榜单显示等多种激励手段，能够让学生自主的去学习网络安全技术，激发学生学习潜力。

本课题的主要内容是对网络安全教学平台题库内容的设计和开发，结合众多网络安全技术，针对 WEB 通信、HTTP 协议、渗透基础、PHP 基础、SQL 数据库原理、常用渗透方式等基础知识设计和开发出 WEB 基础知识资源题库 15 个，SQL 注入漏洞资源 8 个，PHP 基础知识资源 4 个，文件上传漏洞资源 9 个，合计资源总数 36 个，能够帮助学生循序渐进的理解和掌握网络安全技术基础。

平台也有很多不足之处可以改正，如 Docker 容器的管理目前采用手工管理方式，其实可以开发出一套 Docker 容器管理系统，将 Docker 管理系统融合到网络安全教学平台，平台的实用性会大大提升，同时对于资源的开发还不够全面，网络安全还涵盖了软件安全等内容，对于已完成的资源开发也未做到一定深度，展示出来的网络安全技术也仅仅是冰山一角。

通过本次课题的实现，我学到了很多以前没接触过的网络安全知识，也深刻的理解到虚拟化技术对网络安全技术的传播有着非常重大的意义，同时对三层架构的 WEB 开发模式有了新的认识。

参考文献

[1] 王沛.基于 Flask 框架的创新创业平台系统的设计与实现[D].山东大学,2018.

[2] 葛育佳,李亮,陆冬磊.基于 Docker 服务器的云平台实现[J].电脑知识与技,2019,15(04):230-232.

[3] 李源.Web 漏洞及安全性探析[J].信息系统工程,2019(01):76.

[4] 尹毅.代码审计:企业级 Web 代码安全架构[M].机械工业出版社,2016.

[5] 刘彦鑫.SQL 注入原理及防范方法[J].电子世界,2019(07):178-179.

[6] 郝子希,王志军,刘振宇.文件上传漏洞的攻击方法与防御措施研究[J].计算机技术与发展,2019,29(02):129-134.

[7] 袁慧,张轩.Docker 框架下的虚拟化应用平台建设研究[J].自动化与仪器仪表,2019(03):39-42.

[8] 谭彬,杨明,梁业裕,宁建创.Web 安全漏洞研究和防范[J].通信技术,2017,50(04):795-802.

[9] 陶国武.网络渗透测试技术分析[J].通讯世界,2018(09):48-49.

[10] BuiT.Analysis of docker security[J].arXiv preprint arXiv:1501.02967,2015.