全部课程 (/courses/) / Python打造漏洞扫描器 (/courses/761) / 自动生成网页报告

在线实验，请到PC端体验

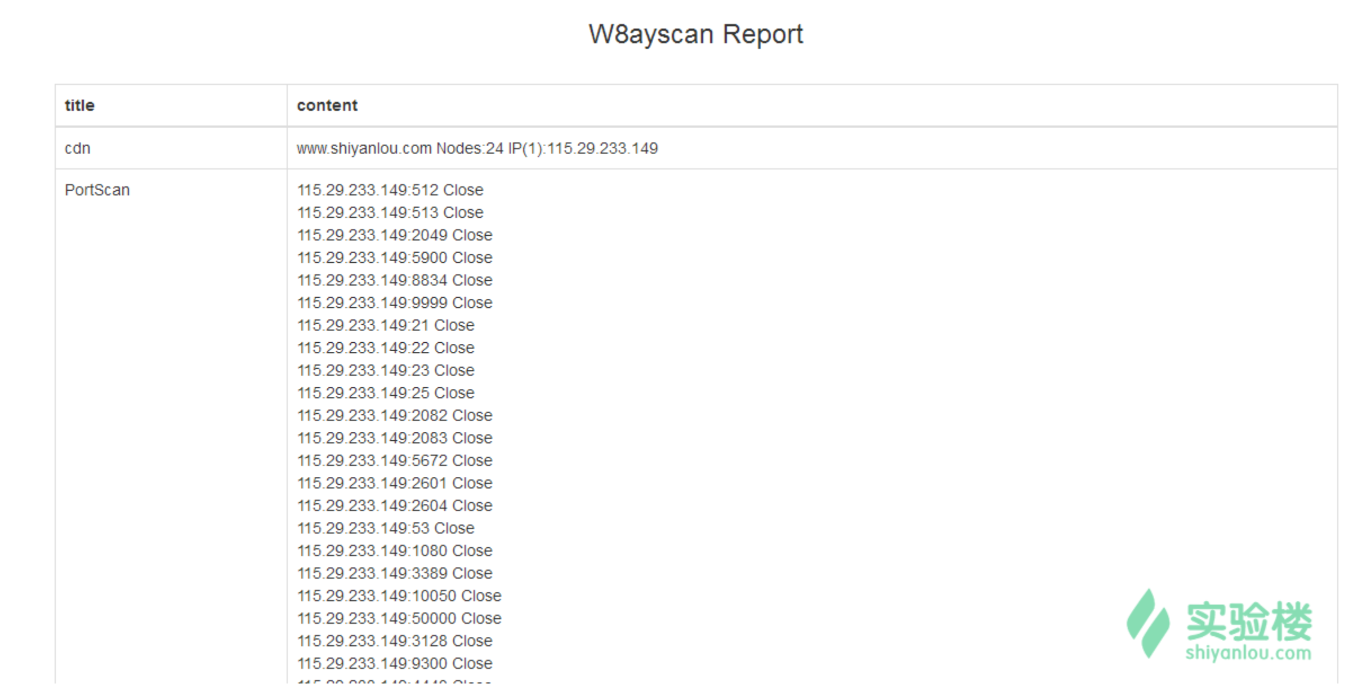
扫描器之自动生成网页报告

一、实验介绍

1.1 实验内容

前面我们写了很多功能模块，信息都是输出到控制台上的，有时候信息多了的时候根本看不过来，我们这节课做的就是将结果存储下来写到文件，然后写个网页生成器来自动生成网页报告。

最后结果是这样哒，是不是感觉酷酷的?



W8ayscan Report

title	content
cdn	www.shiyanlou.com Nodes:24 IP(1):115.29.233.149
PortScan	115.29.233.149:512 Close 115.29.233.149:513 Close 115.29.233.149:2049 Close 115.29.233.149:5900 Close 115.29.233.149:8834 Close 115.29.233.149:9999 Close 115.29.233.149:21 Close 115.29.233.149:22 Close 115.29.233.149:23 Close 115.29.233.149:25 Close 115.29.233.149:2082 Close 115.29.233.149:2083 Close 115.29.233.149:5672 Close 115.29.233.149:2601 Close 115.29.233.149:2604 Close 115.29.233.149:53 Close 115.29.233.149:1080 Close 115.29.233.149:3389 Close 115.29.233.149:10050 Close 115.29.233.149:50000 Close 115.29.233.149:3128 Close 115.29.233.149:9300 Close

1.2 实验知识点

- 数据收集与输出

1.3 实验环境

- Python2.7
- Xfce终端
- Sublime

1.4 适合人群

本课程难度为一般，属于初级级别课程，适合具有**Python**基础的用户，熟悉**python**基础知识加深巩固。

1.5 代码获取

```
$ wget http://labfile.oss.aliyuncs.com/courses/761/shiyanlouscan9.zip
$ unzip shiyanlouscan9.zip
```

二、实验内容

2.1 定义结果输出类

我们需要创建一个结果输出类，用这个类来收集数据，输出数据，并且可以写到hmtl中生成网页报告。

定义一个 **`ouputer`** 类

类中我们定义几个函数即可：

```
class outputer
    def add(self,key,data):通过字典方式添加数据
    def add_list(self,key,data): 通过列表方式添加数据
    def get(self,key):获取某个数据
    def show(self):显示加入的数据
    def build_html(self,name):生成网页 name为保存的文件名
```

2.2 完整代码

类中实列一个变量，因为类定义的时候实例化的变量是不会改变的。
完整代码：

```
import sys
reload(sys)
sys.setdefaultencoding('utf-8')

class outputer:
    data = {}

    def get(self,key):
        if key in self.data:
            return self.data[key]
        return None

    def add(self,key,data):
        self.data[key] = data

    def add_list(self,key,data):
        if key not in self.data:
            self.data[key] = []
        self.data[key].append(data)

    def show(self):
        for key in self.data:
            print "%s:%s"%(key,self.data[key])

    def _build_table(self):
        _str = ""
        for key in self.data:
            if isinstance(self.data[key],list):
                _td = ""
                for key2 in self.data[key]:
                    _td += key2 + '</br>'
                _str += "<tr><td>%s</td><td>%s</td></tr>"%(key,_td)
            else:
                _str += "<tr><td>%s</td><td>%s</td></tr>"%(key,self.data[key])
        return _str
    def build_html(self,filename):
        html_head = '''
        <!DOCTYPE html>
<html lang="zh-CN">
  <head>
    <meta charset="gbk">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>W8ayscan Report</title>
<link rel="stylesheet" href="https://cdn.bootcss.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384
-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://cdn.bootcss.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script src="https://cdn.bootcss.com/respond.js/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
<div class="container container-fluid">
    <div class="row-fluid">
        <div class="span12">
            <h3 class="text-center">
                W8ayscan Report
            </h3>
            </BR>
            <table class="table table-bordered">
                <thead>
                    <tr>
                        <th>
                            title
                        </th>
                        <th>
                            content
                        </th>
                    </tr>
                </thead>
                <tbody>
                    build_html_w8ayScan
                </tbody>
            </table>
        </div>
    </div>
</div>  </body>
</html>'''.replace("build_html_w8ayScan",self._build_table())
        file_object = open(filename+'.html', 'w')
        file_object.write(html_head)
        file_object.close()
```

保存在 /lib/core/outputer.py 。

2.3 使用方法

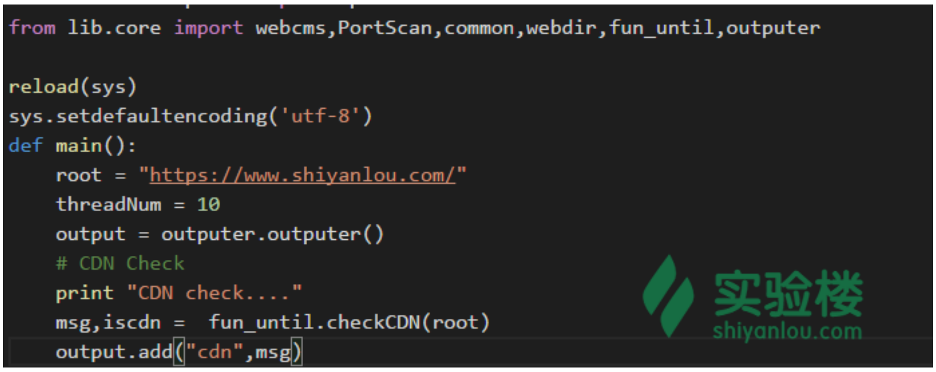
在需要的地方

```
from lib.core import outputer
output = outputer.outputer()
```

来初始化，通过 output.add() 或者 output.add_list() 加入数据，

在功能模块中显示数据的地方将数据添加进来即可。

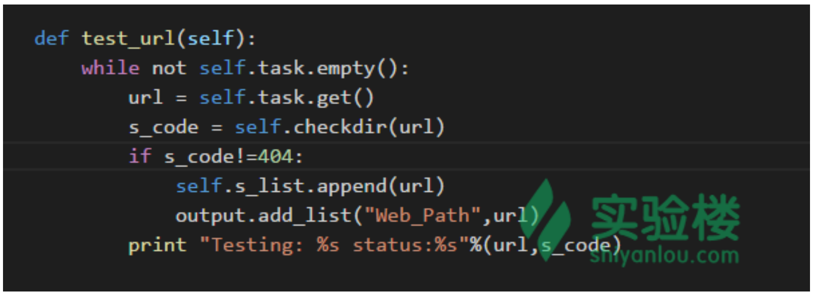
比如这是 w8ay.py 中的。



```
from lib.core import webcms,PortScan,common,webdir,fun_until,outputer

reload(sys)
sys.setdefaultencoding('utf-8')
def main():
    root = "https://www.shiyanlou.com/"
    threadNum = 10
    output = outputer.outputer()
    # CDN Check
    print "CDN check...."
    msg,iscdn =  fun_until.checkCDN(root)
    output.add("cdn",msg)
```

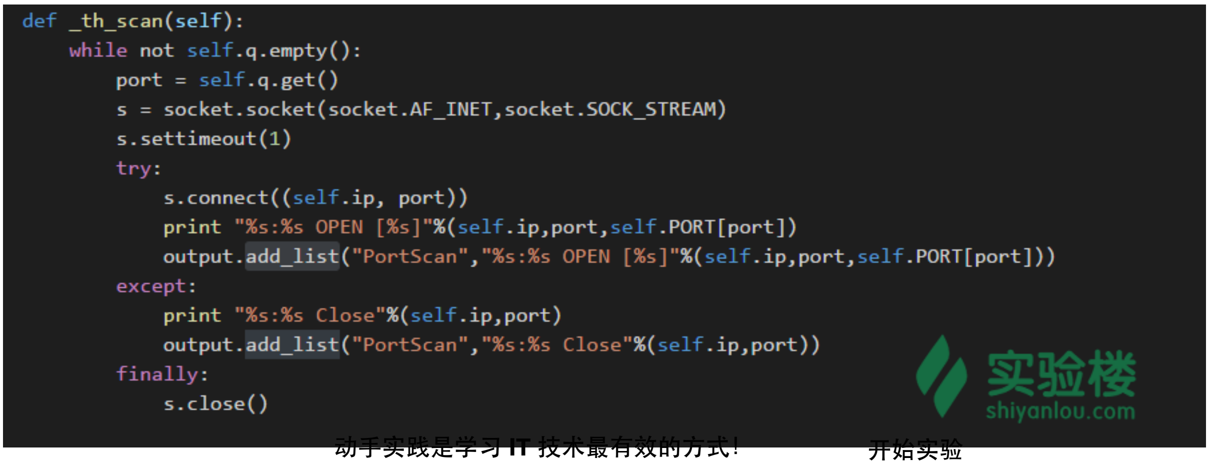
但是数据打印出来的地方都是在每个功能模块的内部实现的，所以我们要在每个功能模块中类似加入。是的，每个。。。

web敏感文件扫描模块加入



```
def test_url(self):
    while not self.task.empty():
        url = self.task.get()
        s_code = self.checkdir(url)
        if s_code!=404:
            self.s_list.append(url)
            output.add_list("Web_Path",url)
        print "Testing: %s status:%s"%(url,s_code)
```

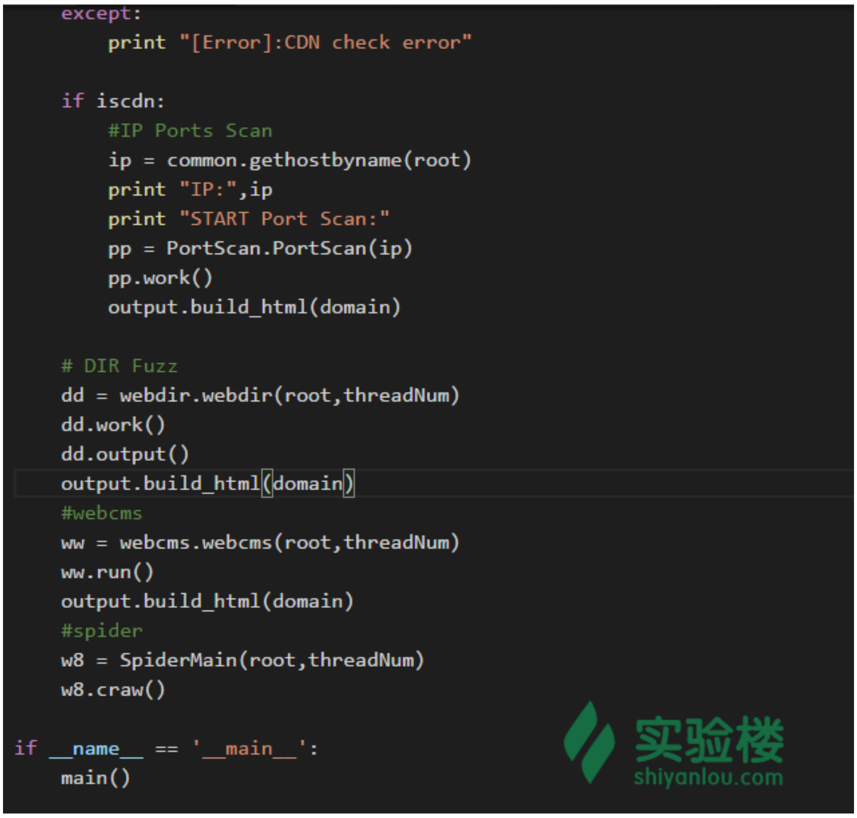
端口扫描中加入



```
def _th_scan(self):
    while not self.q.empty():
        port = self.q.get()
        s = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
        s.settimeout(1)
        try:
            s.connect((self.ip, port))
            print "%s:%s OPEN [%s]"%(self.ip,port,self.PORT[port])
            output.add_list("PortScan","%s:%s OPEN [%s]"%(self.ip,port,self.PORT[port]))
        except:
            print "%s:%s Close"%(self.ip,port)
            output.add_list("PortScan","%s:%s Close"%(self.ip,port))
        finally:
            s.close()
```

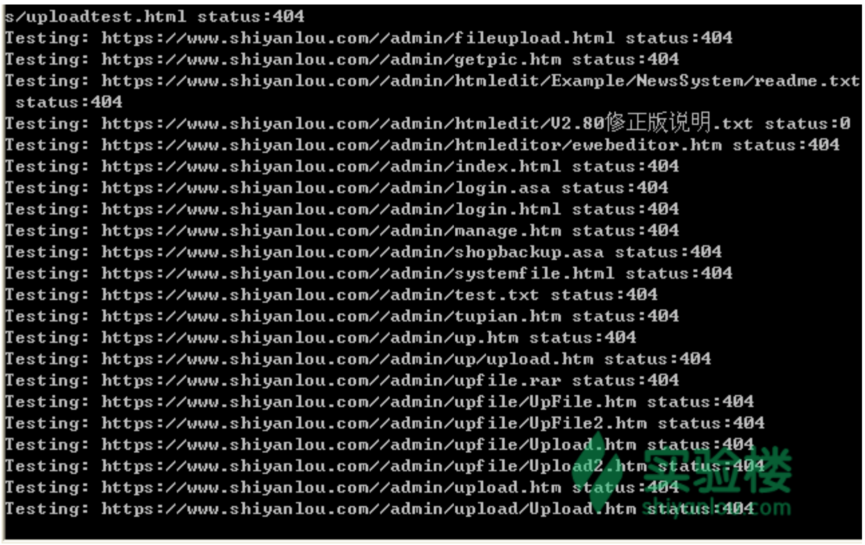
等等。。 只要是功能模块输出的地方全部加上这个类，爬虫插件里面也要加上。

最后，在主调度程序每调用完一个程序后生成一次html。



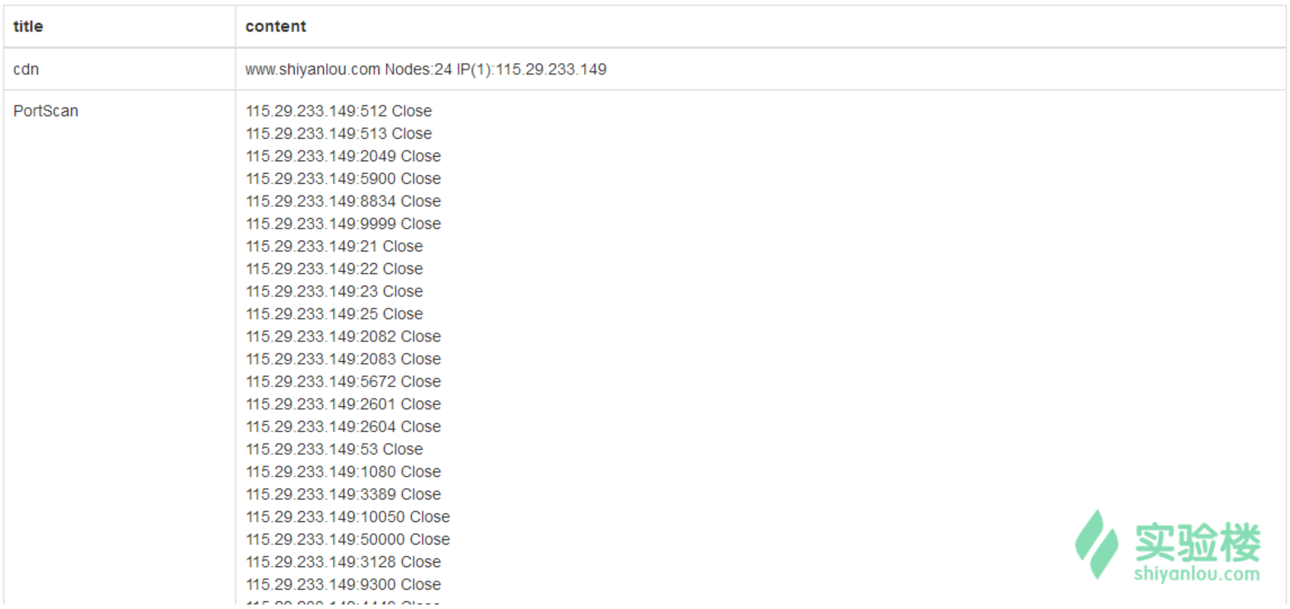
```
    except:
        print "[Error]:CDN check error"

    if iscdn:
        #IP Ports Scan
        ip = common.gethostbyname(root)
        print "IP:",ip
        print "START Port Scan:"
        pp = PortScan.PortScan(ip)
        pp.work()
        output.build_html(domain)

    # DIR Fuzz
    dd = webdir.webdir(root,threadNum)
    dd.work()
    dd.output()
    output.build_html(domain)
    #webcms
    ww = webcms.webcms(root,threadNum)
    ww.run()
    output.build_html(domain)
    #spider
    w8 = SpiderMain(root,threadNum)
    w8.craw()

if __name__ == '__main__':
    main()
```

扫描器运行截图



```
s/uploadtest.html status:404
Testing: https://www.shiyanlou.com//admin/fileupload.html status:404
Testing: https://www.shiyanlou.com//admin/getpic.htm status:404
Testing: https://www.shiyanlou.com//admin/htmledit/Example/NewsSystem/readme.txt
 status:404
Testing: https://www.shiyanlou.com//admin/htmledit/V2.80修正版说明.txt status:0
Testing: https://www.shiyanlou.com//admin/htmleditor/ewebeditor.htm status:404
Testing: https://www.shiyanlou.com//admin/index.html status:404
Testing: https://www.shiyanlou.com//admin/login.asa status:404
Testing: https://www.shiyanlou.com//admin/login.html status:404
Testing: https://www.shiyanlou.com//admin/manage.htm status:404
Testing: https://www.shiyanlou.com//admin/shopbackup.asa status:404
Testing: https://www.shiyanlou.com//admin/systemfile.html status:404
Testing: https://www.shiyanlou.com//admin/test.txt status:404
Testing: https://www.shiyanlou.com//admin/tupian.htm status:404
Testing: https://www.shiyanlou.com//admin/up.htm status:404
Testing: https://www.shiyanlou.com//admin/up/upload.htm status:404
Testing: https://www.shiyanlou.com//admin/upfile.rar status:404
Testing: https://www.shiyanlou.com//admin/upfile/UpFile.htm status:404
Testing: https://www.shiyanlou.com//admin/upfile/UpFile2.htm status:404
Testing: https://www.shiyanlou.com//admin/upfile/Upload.htm status:404
Testing: https://www.shiyanlou.com//admin/upfile/Upload2.htm status:404
Testing: https://www.shiyanlou.com//admin/upload.htm status:404
Testing: https://www.shiyanlou.com//admin/upload/Upload.htm status:404
```

生成的网页报告

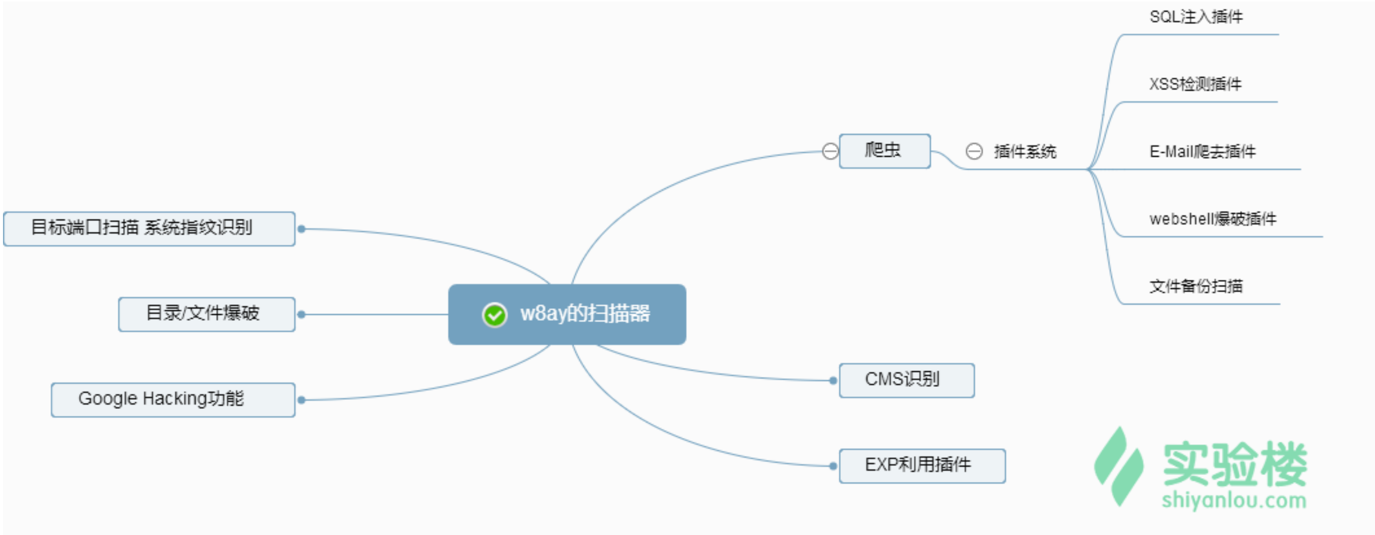


W8ayscan Report

title	content
cdn	www.shiyanlou.com Nodes:24 IP(1):115.29.233.149
PortScan	115.29.233.149:512 Close 115.29.233.149:513 Close 115.29.233.149:2049 Close 115.29.233.149:5900 Close 115.29.233.149:8834 Close 115.29.233.149:9999 Close 115.29.233.149:21 Close 115.29.233.149:22 Close 115.29.233.149:23 Close 115.29.233.149:25 Close 115.29.233.149:2082 Close 115.29.233.149:2083 Close 115.29.233.149:5672 Close 115.29.233.149:2601 Close 115.29.233.149:2604 Close 115.29.233.149:53 Close 115.29.233.149:1080 Close 115.29.233.149:3389 Close 115.29.233.149:10050 Close 115.29.233.149:50000 Close 115.29.233.149:3128 Close 115.29.233.149:9300 Close

三、总结

至此，扫描器课程已经完结了，还记得第一节课的扫描器脑图吗



已经完成了其中大部分的功能，有部分功能在后期编写过程中感觉不需要了，所以没有加上，其实，如果学完了这个系列课程的话，相信大家已经有一个基本的制作扫描器的概念了。

上一节 (/courses/761/labs/2672/document)

下一节 (/courses/761/labs/2674/document)

课程教师

new4

共发布过**1**门课程

查看老师的所有课程 > (/teacher/102428)

动手做实验，轻松学IT

公司

(http://weibo.com/shiyanlou2013

动手实践是学习 IT 技术最有效的方式！

合作

开始实验