

此文档为《Java代码审计零基础入门到项目实战》配套教材，由【闪石星曜CyberSecurity】出品。

请勿对外泄露，一经发现严肃处理！

课程学习中有任何疑问，可添加好友 Power_7089 寻求帮助，为你答疑解惑。

在前面的任意文件上传漏洞和任意文件读取与下载漏洞两章节中简单提到了目录穿越漏洞，本节我们配合JTopCms详细讲解下目录穿越漏洞。

一、目录穿越漏洞

1、什么是目录穿越漏洞

目录穿越漏洞，也叫做目录遍历/路径遍历漏洞。常发生于文件上传，文件下载，文件下载等处。由于后端直接接受使用前端传来的文件名，并没有对文件名进行过滤，从而导致攻击者可通过使用 `../` 的方式进行目录穿越，以达到下载任意文件，删除任意文件，或是将文件上传到任意目录下。

其中 `../` 在Windows和Linux系统下意思均为向上一层目录。

2、目录穿越漏洞代码

任意文件上传和任意读取与下载章节案例代码均存在目录穿越漏洞，大家可以打开前两期代码回顾学习。

下面两处代码来自Java开源漏洞代码。从中可以看到，目录遍历漏洞明显特征就是直接拼接了前端传来的文件名，并且对敏感字符没有进行过滤。

大家自行分析调试研究。

```
来自: https://github.com/j3ers3/Hello-Java-Sec/blob/master/src/main/java/com/best/hello/controller/Traversal.java

/**
 * @poc http://127.0.0.1:8888/Traversal/download?
filename=../../../../../../../../etc/passwd
 */
@ApiOperation(value = "vul: 任意文件下载")
@GetMapping("/download")
public String download(String filename, HttpServletRequest request,
HttpServletRequestResponse response) {
    String filePath = System.getProperty("user.dir") + "/logs/" + filename;
    log.info("[vul] 目录遍历: " + filePath);

    try {
        File file = new File(filePath);
        InputStream fis = new BufferedInputStream(new FileInputStream(file));
        byte[] buffer = new byte[fis.available()];
        fis.read(buffer);
        fis.close();

        response.reset();
        response.setHeader("Content-Disposition", "attachment;filename=" +
filename);
        response.setHeader("Content-Length", "" + file.length());
    }
}
```

```

        OutputStream toClient = new
BufferedOutputStream(response.getOutputStream());
        response.setContentType("application/octet-stream");
        toClient.write(buffer);
        toClient.flush();
        toClient.close();
        return "下载文件成功: " + filePath;
    } catch (Exception e) {
        e.printStackTrace();
        return "未找到文件: " + filePath;
    }
}

/**
 * @poc http://127.0.0.1:8888/Traversal/download/safe?filename=../
 */
@ApiOperation(value = "safe: 过滤../")
@GetMapping("/download/safe")
public String download_safe(String filename) {

    if (!Security.checkTraversal(filename)) {
        String filePath = System.getProperty("user.dir") + "/logs/" +
filename;
        return "安全路径: " + filePath;
    } else {
        return "检测到非法遍历";
    }
}
}

```

来自: [https://github.com/JoyChou93/java-sec-](https://github.com/JoyChou93/java-sec-code/blob/master/src/main/java/org/joychou/controller/PathTraversal.java)

[code/blob/master/src/main/java/org/joychou/controller/PathTraversal.java](https://github.com/JoyChou93/java-sec-code/blob/master/src/main/java/org/joychou/controller/PathTraversal.java)

```

public class PathTraversal {

    protected final Logger logger = LoggerFactory.getLogger(this.getClass());

    /**
     * http://localhost:8080/path_traversal/vul?
filepath=../../../../../../etc/passwd
     */
    @GetMapping("/path_traversal/vul")
    public String getImage(String filepath) throws IOException {
        return getImgBase64(filepath);
    }

    @GetMapping("/path_traversal/sec")
    public String getImageSec(String filepath) throws IOException {
        if (SecurityUtil.pathFilter(filepath) == null) {
            logger.info("Illegal file path: " + filepath);
            return "Bad boy. Illegal file path.";
        }
        return getImgBase64(filepath);
    }

    private String getImgBase64(String imgFile) throws IOException {

```

```

logger.info("working directory: " + System.getProperty("user.dir"));
logger.info("File path: " + imgFile);

File f = new File(imgFile);
if (f.exists() && !f.isDirectory()) {
    byte[] data = Files.readAllBytes(Paths.get(imgFile));
    return new String(Base64.encodeBase64(data));
} else {
    return "File doesn't exist or is not a file.";
}
}

public static void main(String[] argv) throws IOException {
    String aa = new String(Files.readAllBytes(Paths.get("pom.xml")),
StandardCharsets.UTF_8);
    System.out.println(aa);
}

}

public static String pathFilter(String filepath) {
    String temp = filepath;

    // use while to solve multi urlencode
    while (temp.indexOf('%') != -1) {
        try {
            temp = URLDecoder.decode(temp, "utf-8");
        } catch (UnsupportedEncodingException e) {
            logger.info("Unsupported encoding exception: " + filepath);
            return null;
        } catch (Exception e) {
            logger.info(e.toString());
            return null;
        }
    }
}
}

```

3、目录穿越漏洞绕过

目录穿越漏洞绕过并不是百分百好使。在这里仅是提供一些思路。在做黑盒测试时可以尝试使用这些绕过方法。在做代码审计时还得具体代码具体分析。

比如上面漏洞代码中所提供的一处代码是安全代码，使用了while循环解码URL中字符，防范了各种URL编码来绕过。

3.1、URL编码

单次的URL编码，`../` 结果为：`..%2F`，`%2E%2E%2F`。

3.2、双重URL编码

进行两次URL编码

```
. = %252e  
/ = %252f  
\ = %255c
```

3.3、Unicode的URL编码

对字符进行Unicode编码后再进行URL编码

```
. = %u002e  
/ = %u002f  
\ = %u005c
```

3.4、UTF-8的Unicode编码

```
. = %c0%2e, %e0%40%ae, %c0%ae  
/ = %c0%af, %e0%80%af, %c0%2f  
\ = %c0%5c, %c0%80%5c
```

3.5、超长UTF-8编码

```
.: %c0%2e, %e0%40%ae, %c0%ae  
/: %c0%af, %e0%80%af, %c0%2f  
\: %c0%5c, %c0%80%5c
```

3.6、空字节截断

也就是大家熟知的00阶段，用于判断后缀名，使用空字节URL编码绕过。

```
../../../../../../../../passwd%00.jpg
```

3.7、双重 ../

仅做一次判断删除或替换 ../ 情况，可使用 ../../ 方式绕过。

%u002e%u002e%u002f

4、目录穿越漏洞敏感文件

相对于前面章节给出的敏感文件，下面更全面一些。

4.1、Windows系统

```
### C:/Users/Administrator/NTUser.dat  
用户个人设置和配置文件。
```

```
### C:/Documents and Settings/Administrator/NTUser.dat  
旧版本的用户个人设置和配置文件（Windows XP及更早版本）。
```

C:/apache/logs/access.log

Apache web服务器的访问日志，记录了所有HTTP请求。

C:/apache/logs/error.log

Apache web服务器的错误日志，记录了服务器运行过程中的错误信息。

C:/apache/php/php.ini

Apache服务器中PHP模块的配置文件。

C:/boot.ini

Windows操作系统启动时使用的配置文件。

C:/inetpub/wwwroot/global.asa

IIS (Internet Information Services) 服务器的ASP应用程序的全局配置文件。

C:/MySQL/data/hostname.err

MySQL数据库的错误日志文件，记录了与特定主机名相关的错误信息。

C:/MySQL/data/mysql.err

MySQL数据库的错误日志文件，记录了通用的MySQL错误信息。

C:/MySQL/data/mysql.log

MySQL数据库的通用日志文件，记录了一般的数据库活动信息。

C:/MySQL/my.cnf

MySQL数据库服务器的配置文件。

C:/MySQL/my.ini

MySQL数据库服务器的配置文件。

C:/php4/php.ini

PHP 4版本的配置文件。

C:/php5/php.ini

PHP 5版本的配置文件。

C:/php/php.ini

PHP配置文件。

C:/Program Files/Apache Group/Apache2/conf/httpd.conf

Apache web服务器的主配置文件。

C:/Program Files/Apache Group/Apache/conf/httpd.conf

Apache web服务器的主配置文件 (旧版本路径)。

C:/Program Files/Apache Group/Apache/logs/access.log

Apache web服务器的访问日志。

C:/Program Files/Apache Group/Apache/logs/error.log

Apache web服务器的错误日志。

C:/Program Files/FileZilla Server/FileZilla Server.xml

FileZilla FTP服务器的配置文件。

C:/Program Files/MySQL/data/hostname.err

MySQL数据库的错误日志文件，记录了与特定主机名相关的错误信息。

C:/Program Files/MySQL/data/mysql-bin.log

MySQL数据库二进制日志文件，记录了数据库中的更改。

C:/Program Files/MySQL/data/mysql.err

MySQL数据库的错误日志文件，记录了通用的MySQL错误信息。

C:/Program Files/MySQL/data/mysql.log

MySQL数据库的通用日志文件，记录了一般的数据库活动信息。

C:/Program Files/MySQL/my.ini

MySQL数据库服务器的配置文件。

C:/Program Files/MySQL/my.cnf

MySQL数据库服务器的配置文件。

C:/Program Files/MySQL/MySQL Server 5.0/data/hostname.err

MySQL数据库的错误日志文件，记录了与特定主机名相关的错误信息（MySQL 5.0版本）。

C:/Program Files/MySQL/MySQL Server 5.0/data/mysql-bin.log

MySQL数据库二进制日志文件，记录了数据库中的更改（MySQL 5.0版本）。

C:/Program Files/MySQL/MySQL Server 5.0/data/mysql.err

MySQL数据库的错误日志文件，记录了通用的MySQL错误信息（MySQL 5.0版本）。

C:/Program Files/MySQL/MySQL Server 5.0/data/mysql.log

MySQL数据库的通用日志文件，记录了一般的数据库活动信息（MySQL 5.0版本）。

C:/Program Files/MySQL/MySQL Server 5.0/my.cnf

MySQL数据库服务器的配置文件（MySQL 5.0版本）。

C:/Program Files/MySQL/MySQL Server 5.0/my.ini

MySQL数据库服务器的配置文件（MySQL 5.0版本）。

C:/Program Files (x86)/Apache Group/Apache2/conf/httpd.conf

Apache web服务器的主配置文件（32位程序文件夹路径）。

C:/Program Files (x86)/Apache Group/Apache/conf/httpd.conf

Apache web服务器的主配置文件（32位程序文件夹路径，旧版本）。

C:/Program Files (x86)/Apache Group/Apache/conf/access.log

Apache web服务器的访问日志（32位程序文件夹路径）。

C:/Program Files (x86)/Apache Group/Apache/conf/error.log

Apache web服务器的错误日志（32位程序文件夹路径）。

C:/Program Files (x86)/FileZilla Server/FileZilla Server.xml

FileZilla FTP服务器的配置文件（32位程序文件夹路径）。

C:/Program Files (x86)/xampp/apache/conf/httpd.conf

XAMPP集成开发环境中Apache服务器的主配置文件。

C:/WINDOWS/php.ini

windows操作系统中PHP的配置文件。

C:/WINDOWS/Repair/SAM

Windows操作系统的系统文件，用于系统修复。

C:/windows/repair/system

Windows操作系统的系统文件，用于系统修复。

C:/windows/repair/software

Windows操作系统的系统文件，用于系统修复。

C:/windows/repair/security

Windows操作系统的系统文件，用于系统修复。

C:/WINDOWS/System32/drivers/etc/hosts

Windows操作系统的主机文件，用于IP地址与主机名的映射。

C:/windows/win.ini

Windows操作系统的INI配置文件，包含Windows设置信息。

C:/WINNT/php.ini

Windows NT操作系统中PHP的配置文件。

C:/WINNT/win.ini

Windows NT操作系统的INI配置文件，包含Windows设置信息。

C:/xampp/apache/bin/php.ini

XAMPP集成开发环境中Apache服务器的PHP模块配置文件。

C:/xampp/apache/logs/access.log

XAMPP集成开发环境中Apache服务器的访问日志。

C:/xampp/apache/logs/error.log

XAMPP集成开发环境中Apache服务器的错误日志。

C:/windows/Panther/Unattend/Unattended.xml

Windows操作系统的无人值守安装配置文件。

C:/windows/Panther/Unattended.xml

Windows操作系统的无人值守安装配置文件（旧版本路径）。

C:/windows/debug/NetSetup.log

Windows操作系统的网络设置日志文件。

C:/windows/system32/config/AppEvent.Evt

Windows事件查看器的应用程序事件日志。

C:/windows/system32/config/SecEvent.Evt

Windows事件查看器的安全事件日志。

C:/windows/system32/config/default.sav

Windows操作系统的系统配置备份文件。

C:/windows/system32/config/security.sav

Windows操作系统的安全配置备份文件。

C:/windows/system32/config/software.sav

Windows操作系统的软件配置备份文件。

C:/windows/system32/config/system.sav

windows操作系统的系统配置备份文件。

C:/windows/system32/config/regback/default

windows注册表的系统配置备份文件。

C:/windows/system32/config/regback/sam

windows注册表的安全配置备份文件。

C:/windows/system32/config/regback/security

windows注册表的安全配置备份文件。

C:/windows/system32/config/regback/system

windows注册表的系统配置备份文件。

C:/windows/system32/config/regback/software

windows注册表的软件配置备份文件。

C:/Program Files/MySQL/MySQL Server 5.1/my.ini

MySQL数据库服务器的配置文件（MySQL 5.1版本）。

C:/windows/System32/inetsrv/config/schema/ASPNET_schema.xml

IIS服务器的ASP.NET配置文件。

C:/windows/System32/inetsrv/config/applicationHost.config

IIS服务器的主机配置文件。

C:/inetpub/logs/LogFiles/W3SVC1/u_ex[YYMMDD].log

IIS服务器的访问日志，以日期为后缀的文件名。

4.2、Linux系统

/etc/passwd

系统用户的账户信息文件，包含了每个用户的基本信息。

/etc/shadow

系统用户的密码信息文件，存储了加密后的用户密码。

/etc/aliases

系统邮件别名文件，用于将邮件发送到指定的用户或邮件列表。

/etc/anacrontab

Anacron任务调度器的配置文件，用于管理系统定期执行的任务。

/etc/apache2/apache2.conf

Apache2 web服务器的主配置文件。

/etc/apache2/httpd.conf

Apache2 web服务器的主配置文件（旧版本路径）。

/etc/at.allow

允许使用at命令的用户列表。

/etc/at.deny

禁止使用**at**命令的用户列表。

/etc/bashrc

Bash shell的全局配置文件，用于配置Bash的环境变量和别名等。

/etc/bootptab

BOOTP（Bootstrap Protocol）服务器的配置文件，用于网络启动配置。

/etc/chrootUsers

指定了可以被chroot（限制用户访问的根目录）的用户列表。

/etc/chtcp.conf

Caudium Web服务器的配置文件。

/etc/cron.allow

允许使用cron任务调度器的用户列表。

/etc/cron.deny

禁止使用cron任务调度器的用户列表。

/etc/crontab

系统范围的cron任务调度器配置文件，用于定时执行系统任务。

/etc/cups/cupsd.conf

CUPS（Common UNIX Printing System）打印服务器的主配置文件。

/etc/exports

NFS（Network File System）服务器的共享目录配置文件。

/etc/fstab

文件系统表，记录了系统启动时需要挂载的文件系统信息。

/etc/ftpaccess

FTP服务器的访问控制文件。

/etc/ftpchroot

指定了FTP用户的根目录。

/etc/ftphosts

指定了FTP服务器的访问控制规则。

/etc/groups

系统用户组的配置文件，包含了每个用户组的信息。

/etc/grub.conf

GRUB引导加载程序的配置文件。

/etc/hosts

主机名与IP地址的映射关系文件。

/etc/hosts.allow

允许访问网络服务的主机列表。

/etc/hosts.deny

拒绝访问网络服务的主机列表。

/etc/httpd/access.conf

Apache HTTP服务器的访问控制文件。

/etc/httpd/conf/httpd.conf

Apache HTTP服务器的主配置文件。

/etc/httpd/httpd.conf

Apache HTTP服务器的主配置文件。

/etc/httpd/logs/access_log

Apache HTTP服务器的访问日志。

/etc/httpd/logs/access.log

Apache HTTP服务器的访问日志。

/etc/httpd/logs/error_log

Apache HTTP服务器的错误日志。

/etc/httpd/logs/error.log

Apache HTTP服务器的错误日志。

/etc/httpd/php.ini

Apache服务器中PHP模块的配置文件。

/etc/httpd/srm.conf

Apache HTTP服务器的资源管理器配置文件。

/etc/inetd.conf

inetd守护进程的配置文件，用于启动和管理网络服务。

/etc/inittab

系统初始化进程的配置文件，定义了系统启动时要执行的任务。

/etc/issue

系统登录时显示的信息文件。

/etc/lighttpd.conf

Lighttpd web服务器的主配置文件。

/etc/lilo.conf

LILO引导加载程序的配置文件。

/etc/logrotate.d/ftp

FTP日志文件的日志轮转配置文件。

/etc/logrotate.d/proftpd

ProFTPD日志文件的日志轮转配置文件。

/etc/logrotate.d/vsftpd.log

vsftpd日志文件的日志轮转配置文件。

/etc/lsb-release

Linux标准基础版本（LSB）的版本信息文件。

/etc/motd

登录时显示的欢迎信息文件。

/etc/modules.conf

Linux内核模块的配置文件。

/etc/mtab

当前已挂载文件系统的信息文件。

/etc/my.cnf

MySQL数据库服务器的配置文件。

/etc/my.conf

MySQL数据库服务器的配置文件。

/etc/mysql/my.cnf

MySQL数据库服务器的配置文件。

/etc/network/interfaces

Linux网络接口的配置文件。

/etc/networks

网络名称和IP地址的映射关系文件。

/etc/npasswd

NIS (Network Information Service) 密码文件。

/etc/php4.4/fcgi/php.ini

PHP FastCGI模块的配置文件。

/etc/php4/apache2/php.ini

Apache2中PHP模块的配置文件。

/etc/php4/apache/php.ini

Apache中PHP模块的配置文件。

/etc/php4/cgi/php.ini

CGI模式下PHP的配置文件。

/etc/php4/apache2/php.ini

Apache2中PHP模块的配置文件。

/etc/php5/apache2/php.ini

Apache2中PHP5模块的配置文件。

/etc/php5/apache/php.ini

Apache中PHP5模块的配置文件。

/etc/php/apache2/php.ini

Apache2中PHP模块的配置文件。

/etc/php/apache/php.ini

Apache中PHP模块的配置文件。

/etc/php/cgi/php.ini

CGI模式下PHP的配置文件。

/etc/php.ini

PHP的通用配置文件。

/etc/php/php4/php.ini

PHP4模块的配置文件。

/etc/php/php.ini

PHP模块的配置文件。

/etc/printcap

打印机设备的配置文件。

/etc/profile

系统全局的用户环境配置文件。

/etc/proftpd.conf

ProFTPD FTP服务器的主配置文件。

/etc/proftpd/proftpd.conf

ProFTPD FTP服务器的主配置文件。

/etc/pure-ftpd.conf

Pure-FTPd FTP服务器的主配置文件。

/etc/pureftpd.passwd

Pure-FTPd FTP服务器的用户密码文件。

/etc/pureftpd.pdb

Pure-FTPd FTP服务器的用户数据库文件。

/etc/pure-ftpd/pure-ftpd.conf

Pure-FTPd FTP服务器的主配置文件。

/etc/pure-ftpd/pure-ftpd.pdb

Pure-FTPd FTP服务器的用户数据库文件。

/etc/pure-ftpd/putreftpd.pdb

Pure-FTPd FTP服务器的用户数据库文件。

/etc/redhat-release

Red Hat操作系统的版本信息文件。

/etc/resolv.conf

DNS解析器的配置文件。

/etc/samba/smb.conf

Samba服务器的配置文件。

/etc/snmpd.conf

SNMP守护进程的配置文件。

/etc/ssh/ssh_config

SSH客户端的配置文件。

/etc/ssh/sshd_config

SSH服务器的配置文件。

/etc/ssh/ssh_host_dsa_key

SSH服务器的DSA密钥文件。

/etc/ssh/ssh_host_dsa_key.pub

SSH服务器的DSA公钥文件。

/etc/ssh/ssh_host_key

SSH服务器的主机密钥文件。

/etc/ssh/ssh_host_key.pub

SSH服务器的主机公钥文件。

/etc/sysconfig/network

Linux系统网络配置文件。

/etc/syslog.conf

系统日志守护进程的配置文件。

/etc/termcap

终端功能描述数据库文件。

/etc/vhcs2/proftpd/proftpd.conf

ProFTPD FTP服务器的配置文件（VHCS2控制面板）。

/etc/vsftpd.chroot_list

指定了被限制到chroot目录的用户列表。

/etc/vsftpd.conf

vsftpd FTP服务器的主配置文件。

/etc/vsftpd/vsftpd.conf

vsftpd FTP服务器的主配置文件。

/etc/wu-ftp/ftppass

wu-ftp FTP服务器的访问控制文件。

/etc/wu-ftp/ftpshosts

wu-ftp FTP服务器的主机允许文件。

/etc/wu-ftp/ftpusers

wu-ftp FTP服务器的用户访问文件。

/logs/pure-ftp.log

Pure-FTPd FTP服务器的日志文件。

/logs/security_debug.log

安全调试日志文件。

/logs/security.log

安全日志文件。

/opt/lampp/etc/httpd.conf

XAMPP集成开发环境中Apache服务器的主配置文件。

/opt/xampp/etc/php.ini

XAMPP集成开发环境中PHP的配置文件。

/proc/cpuinfo

当前系统CPU信息的虚拟文件。

/proc/filesystems

当前系统支持的文件系统列表。

/proc/interrupts

当前系统中断请求信息的虚拟文件。

/proc/ioports

当前系统IO端口信息的虚拟文件。

/proc/meminfo

当前系统内存信息的虚拟文件。

/proc/modules

当前加载的内核模块信息的虚拟文件。

/proc/mounts

当前已挂载的文件系统信息的虚拟文件。

/proc/stat

当前系统CPU和其他统计信息的虚拟文件。

/proc/swaps

当前系统交换分区信息的虚拟文件。

/proc/version

当前系统内核版本信息的虚拟文件。

/proc/self/net/arp

当前系统ARP（Address Resolution Protocol）缓存信息的虚拟文件。

/root/anaconda-ks.cfg

Anaconda安装程序的Kickstart配置文件。

/usr/etc/pure-ftpd.conf

Pure-FTPd FTP服务器的配置文件（usr路径）。

/usr/lib/php.ini

PHP的配置文件（usr路径）。

/usr/lib/php/php.ini

PHP的配置文件（usr路径）。

/usr/local/apache/conf/modsec.conf

Apache服务器的ModSecurity配置文件。

/usr/local/apache/conf/php.ini

Apache服务器的PHP模块配置文件。

/usr/local/apache/log

Apache服务器的日志目录。

/usr/local/apache/logs

Apache服务器的日志目录。

/usr/local/apache/logs/access_log

Apache服务器的访问日志。

/usr/local/apache/logs/access.log

Apache服务器的访问日志。

/usr/local/apache/audit_log

Apache服务器的审计日志文件。

/usr/local/apache/error_log

Apache服务器的错误日志。

/usr/local/apache/error.log

Apache服务器的错误日志。

/usr/local/cpanel/logs

cPanel控制面板的日志目录。

/usr/local/cpanel/logs/access_log

cPanel控制面板的访问日志。

/usr/local/cpanel/logs/error_log

cPanel控制面板的错误日志。

/usr/local/cpanel/logs/license_log

cPanel控制面板的许可证日志。

/usr/local/cpanel/logs/login_log

cPanel控制面板的登录日志。

/usr/local/cpanel/logs/stats_log

cPanel控制面板的统计日志。

/usr/local/etc/httpd/logs/access_log

Apache服务器的访问日志（usr/local路径）。

/usr/local/etc/httpd/logs/error_log

Apache服务器的错误日志（usr/local路径）。

/usr/local/etc/php.ini

PHP的配置文件（usr/local路径）。

/usr/local/etc/pure-ftpd.conf

Pure-FTPd FTP服务器的配置文件（usr/local路径）。

/usr/local/etc/pureftpd.pdb

Pure-FTPd FTP服务器的用户数据库文件（usr/local路径）。

/usr/local/lib/php.ini

PHP的配置文件（usr/local路径）。

/usr/local/php4/httpd.conf

Apache2中PHP4模块的配置文件。

/usr/local/php4/httpd.conf.php

Apache2中PHP4模块的配置文件。

/usr/local/php4/lib/php.ini

PHP4的配置文件（usr/local路径）。

/usr/local/php5/httpd.conf

Apache2中PHP5模块的配置文件。

/usr/local/php5/httpd.conf.php

Apache2中PHP5模块的配置文件。

/usr/local/php5/lib/php.ini

PHP5的配置文件（usr/local路径）。

/usr/local/php/httpd.conf

Apache2中PHP模块的配置文件。

/usr/local/php/httpd.conf.ini

Apache2中PHP模块的配置文件。

/usr/local/php/lib/php.ini

PHP的配置文件（usr/local路径）。

/usr/local/pureftpd/etc/pure-ftpd.conf

Pure-FTPd FTP服务器的配置文件（usr/local/pureftpd路径）。

/usr/local/pureftpd/etc/pureftpd.pdn

Pure-FTPd FTP服务器的用户数据库文件（usr/local/pureftpd路径）。

/usr/local/pureftpd/sbin/pure-config.pl

Pure-FTPd FTP服务器的配置工具（usr/local/pureftpd路径）。

/usr/local/www/logs/httpd_log

Apache服务器的访问日志（usr/local/www路径）。

/usr/local/Zend/etc/php.ini

Zend Server的PHP模块的配置文件。

/usr/sbin/pure-config.pl

Pure-FTPd FTP服务器的配置工具。

/var/adm/log/xferlog

FTP传输日志文件。

/var/apache2/config.inc

Apache2服务器的配置文件。

/var/apache/logs/access_log

Apache服务器的访问日志。

/var/apache/logs/error_log

Apache服务器的错误日志。

/var/cpanel/cpanel.config

cPanel控制面板的配置文件。

/var/lib/mysql/my.cnf
MySQL数据库服务器的配置文件（var路径）。

/var/lib/mysql/mysql/user.MYD
MySQL数据库服务器的用户数据文件。

/var/local/www/conf/php.ini
PHP的配置文件（var路径）。

/var/log/apache2/access_log
Apache服务器的访问日志。

/var/log/apache2/error_log
Apache服务器的错误日志。

/var/log/apache2/error.log
Apache服务器的错误日志。

/var/log/apache/access_log
Apache服务器的访问日志。

/var/log/apache/access.log
Apache服务器的访问日志。

/var/log/apache/error_log
Apache服务器的错误日志。

/var/log/apache/error.log
Apache服务器的错误日志。

/var/log/apache-ssl/access.log
Apache SSL服务器的访问日志。

/var/log/apache-ssl/error.log
Apache SSL服务器的错误日志。

/var/log/auth.log
系统认证日志文件。

/var/log/boot
系统启动日志文件。

/var/htmp
临时HTTP文件目录。

/var/log/chttp.log
Caudium web服务器的日志文件。

/var/log/cups/error.log
CUPS打印服务器的错误日志。

/var/log/daemon.log
守护进程日志文件。

/var/log/debug

调试日志文件。

/var/log/dmesg

系统启动信息文件。

/var/log/dpkg.log

Debian软件包管理器日志文件。

/var/log/exim_mainlog

Exim邮件传输代理的主日志文件。

/var/log/exim/mainlog

Exim邮件传输代理的主日志文件。

/var/log/exim_paniclog

Exim邮件传输代理的严重错误日志文件。

/var/log/exim.paniclog

Exim邮件传输代理的严重错误日志文件。

/var/log/exim_rejectlog

Exim邮件传输代理的拒绝日志文件。

/var/log/exim/rejectlog

Exim邮件传输代理的拒绝日志文件。

/var/log/faillog

登录失败日志文件。

/var/log/ftplog

FTP服务器的日志文件。

/var/log/ftp-proxy

FTP代理服务器的日志目录。

/var/log/ftp-proxy/ftp-proxy.log

FTP代理服务器的日志文件。

/var/log/httpd/access_log

Apache服务器的访问日志。

/var/log/httpd/access.log

Apache服务器的访问日志。

/var/log/httpd/error_log

Apache服务器的错误日志。

/var/log/httpd/error.log

Apache服务器的错误日志。

/var/log/httpsd/ssl.access_log

Apache SSL服务器的访问日志。

/var/log/httpsd/ssl_log

Apache SSL服务器的日志文件。

/var/log/kern.log

内核日志文件。

/var/log/lastlog

记录用户上次登录时间的日志文件。

/var/log/lighttpd/access.log

Lighttpd web服务器的访问日志。

/var/log/lighttpd/error.log

Lighttpd web服务器的错误日志。

/var/log/lighttpd/lighttpd.access.log

Lighttpd web服务器的访问日志。

/var/log/lighttpd/lighttpd.error.log

Lighttpd web服务器的错误日志。

/var/log/mail.info

邮件系统的信息日志文件。

/var/log/mail.log

邮件系统的日志文件。

/var/log/maillog

邮件系统的日志文件。

/var/log/mail.warn

邮件系统的警告日志文件。

/var/log/message

系统消息日志文件。

/var/log/messages

系统消息日志文件。

/var/log/mysqlerror.log

MySQL数据库服务器的错误日志。

/var/log/mysql.log

MySQL数据库服务器的日志文件。

/var/log/mysql/mysql-bin.log

MySQL数据库服务器的二进制日志文件。

/var/log/mysql/mysql.log

MySQL数据库服务器的日志文件。

/var/log/mysql/mysql-slow.log

MySQL数据库服务器的慢查询日志文件。

/var/log/proftpd

ProFTPD FTP服务器的日志目录。

/var/log/pureftpd.log

Pure-FTPd FTP服务器的日志文件。

/var/log/pure-ftpd/pure-ftpd.log

Pure-FTPd FTP服务器的日志文件。

/var/log/secure

系统安全日志文件。

/var/log/vsftpd.log

vsftpd FTP服务器的日志文件。

/var/log/wtmp

登录和注销事件的记录文件。

/var/log/xferlog

FTP传输日志文件。

/var/log/yum.log

YUM软件包管理器的日志文件。

/var/mysql.log

MySQL数据库服务器的日志文件（var路径）。

/var/run/utmp

当前登录用户信息的文件。

/var/spool/cron/crontabs/root

root用户的cron任务调度器配置文件。

/var/webmin/miniserv.log

webmin控制面板的日志文件。

/var/www/log/access_log

web服务器的访问日志。

/var/www/log/error_log

web服务器的错误日志。

/var/www/logs/access_log

web服务器的访问日志。

/var/www/logs/error_log

web服务器的错误日志。

/var/www/logs/access.log

web服务器的访问日志。

/var/www/logs/error.log

web服务器的错误日志。

~/.atfp_history

用户at命令历史记录文件。

~/.bash_history

用户Bash shell命令历史记录文件。

~/.bash_logout

用户退出Bash shell时执行的脚本文件。

~/.bash_profile

用户Bash shell的配置文件，用于配置用户的环境变量和别名等。

~/.bashrc

用户Bash shell的配置文件，用于配置用户的环境变量和别名等。

~/.gtkrc

用户GTK+图形工具包的配置文件。

~/.login

用户登录时执行的脚本文件。

~/.logout

用户退出登录时执行的脚本文件。

~/.mysql_history

用户MySQL命令历史记录文件。

~/.nano_history

用户Nano文本编辑器命令历史记录文件。

~/.php_history

用户PHP命令历史记录文件。

~/.profile

用户登录时执行的脚本文件。

~/.ssh/authorized_keys

SSH服务器允许登录的用户公钥列表文件。

~/.ssh/id_dsa

用户DSA私钥文件。

~/.ssh/id_dsa.pub

用户DSA公钥文件。

~/.ssh/id_rsa

用户RSA私钥文件。

~/.ssh/id_rsa.pub

用户RSA公钥文件。

~/.ssh/identity

用户SSH1身份验证私钥文件。

~/.ssh/identity.pub

用户SSH1身份验证公钥文件。

~/.viminfo

用户Vim编辑器的历史记录文件。

~/.wm_style

用户Window Maker窗口管理器的配置文件。

```
### ~/.Xdefaults
用户X窗口系统的默认设置文件。

### ~/.xinitrc
用户X窗口系统启动时执行的脚本文件。

### ~/.Xresources
用户X窗口系统的资源配置文件。

### ~/.xsession
用户X窗口系统启动时执行的脚本文件。
```

二、JTopCMS的目录穿越漏洞

该系统内还存在其他漏洞，但本节我们的视角仅放在目录穿越漏洞处。目录穿越漏洞配合文件下载功能可以实现任意文件下载。

1、环境部署

1.1、官方网站

<https://www.jtopcms.com/>

1.2、下载地址

<https://gitee.com/mjtop/JTopCMSV3/releases/tag/JTopCMSV3.0.2-OP>

1.3、所需环境

名称	版本
JTopCMS	V3版本为开源版本，V4为商业版本。我们使用的是V3版本。
Java版本	JDK1.7+即可，我使用的是JDK1.8_261
IDEA	版本随意，我用的最新版(破解方式可看之前发过的主题： https://t.zsxq.com/089yMeGC6)
mysql	5.5.29（我用的PHPstudy集成的Mysql，该版本可从软件管理处下载）
Tomcat	Tomcat8.5.1（版本只要是Tomcat7+都可以）

1.4、部署环境

（我装的IDEA新版中下载了中文插件，对于功能指引都是以中文讲述，英文版的朋友自行对照）

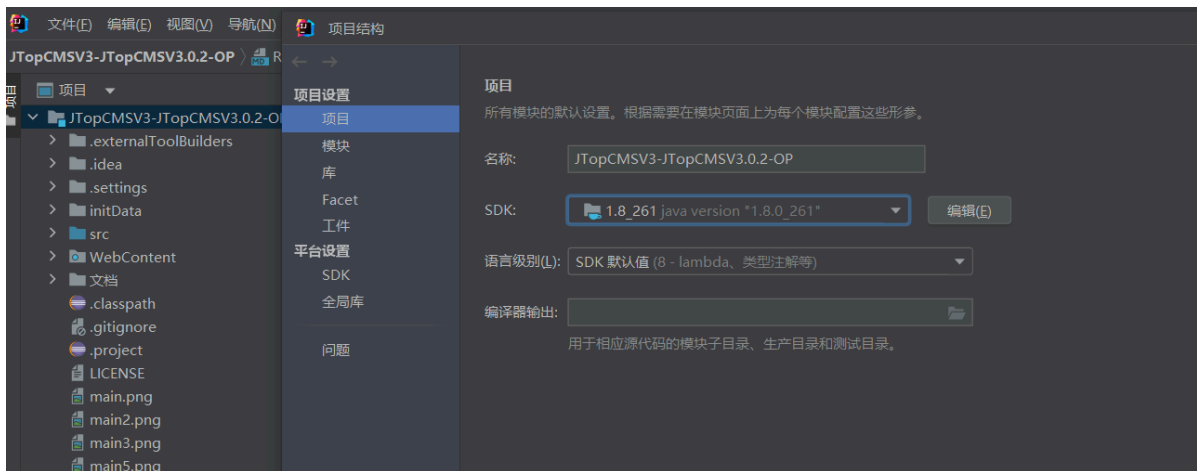
1.4.1、IDEA配置

①、使用IDEA打开JTopCMS，如果有如下图提示，勾选Maven项目，如下图所示：

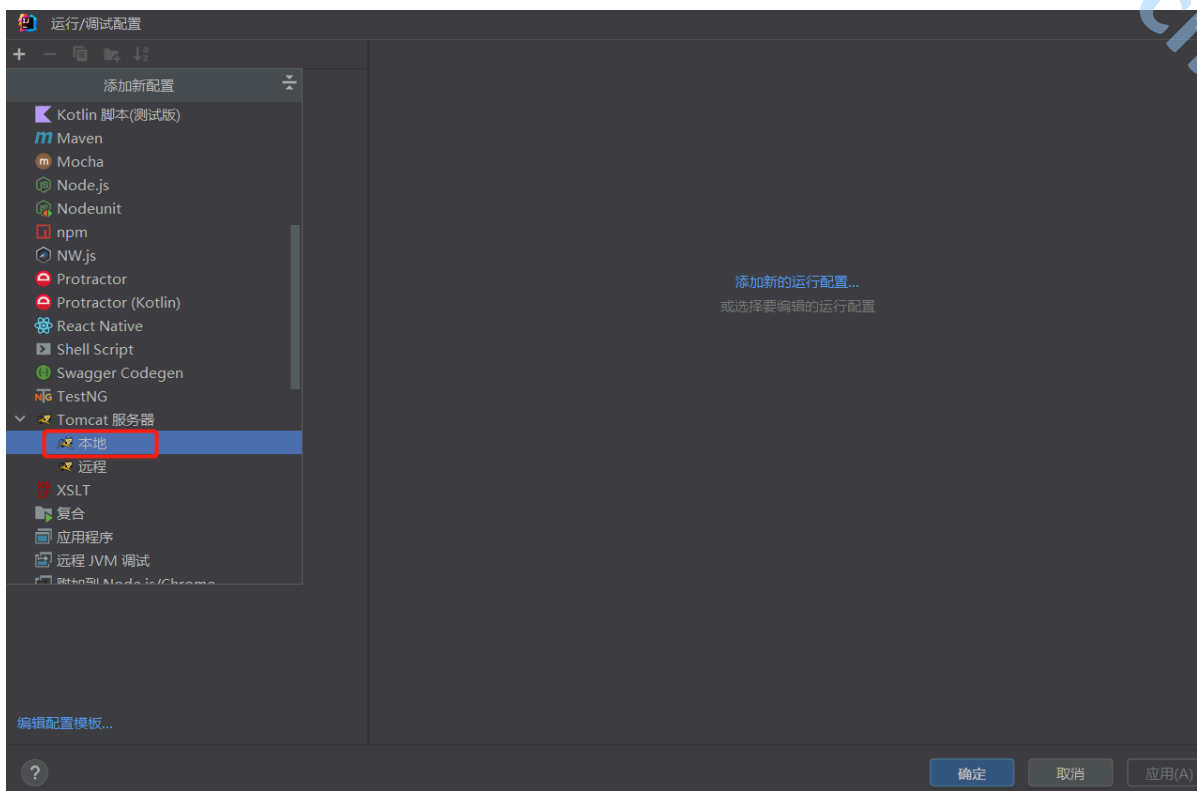


进入后，耐心等待一会，IDEA自动下载相关依赖。

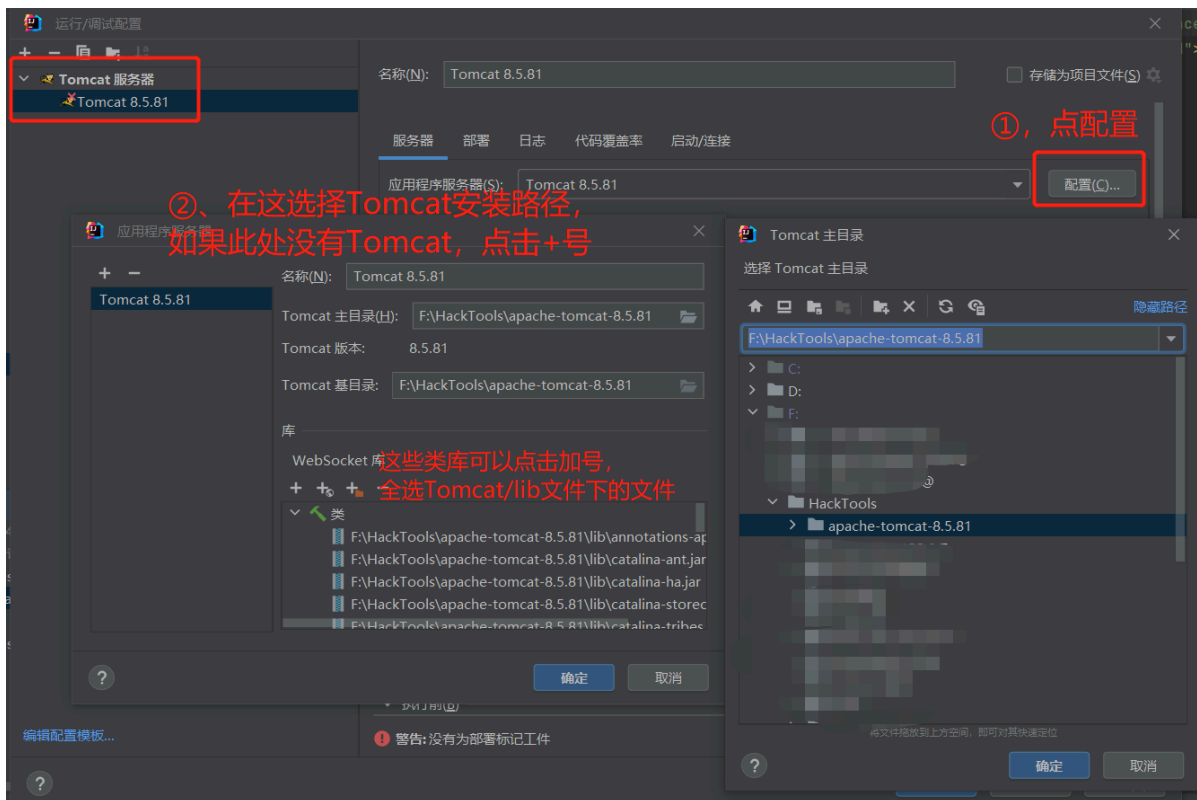
②、如果你的系统中Java有多个版本，请检查下 **文件-项目结构-项目** 中SDK是否为 1.8，如下图所示：



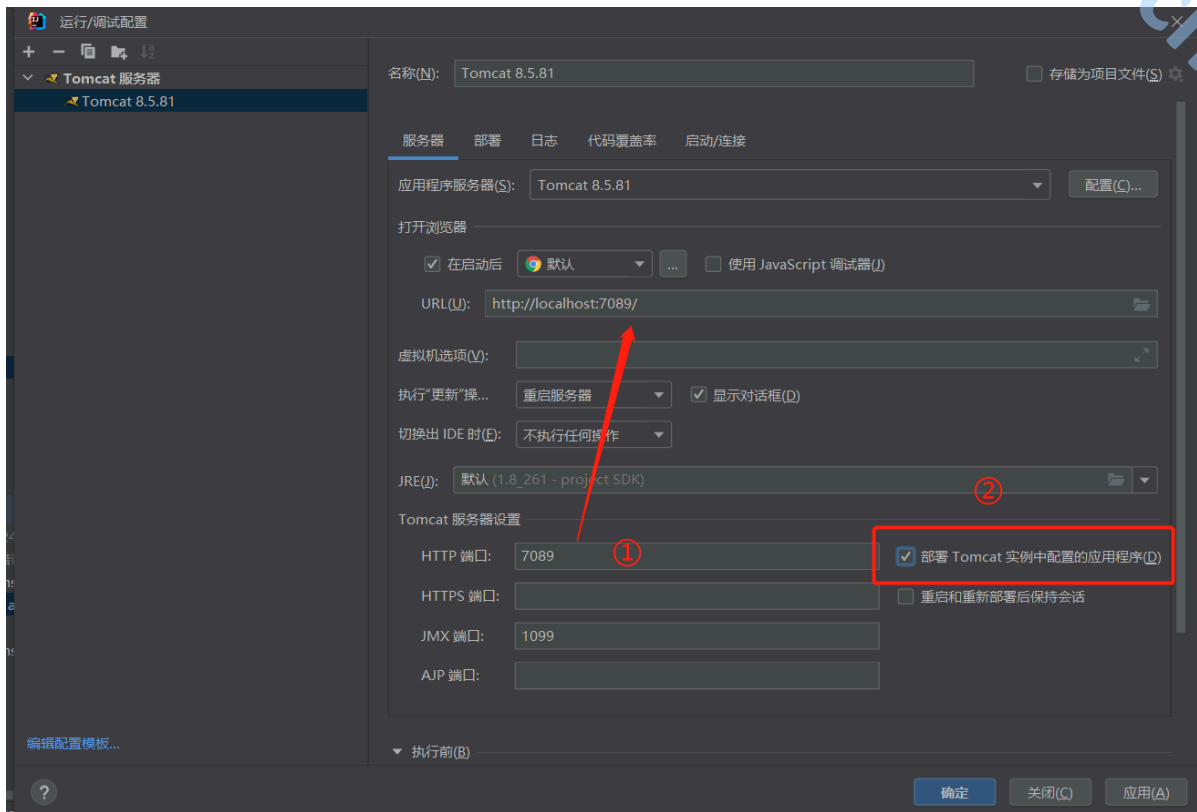
③、下面给项目添加 Tomcat 运行环境，点击左上侧 **编辑配置...**，进入运行/调试配置页面，点击左上侧加号，选择 Tomcat 服务器下本地，如下图所示：



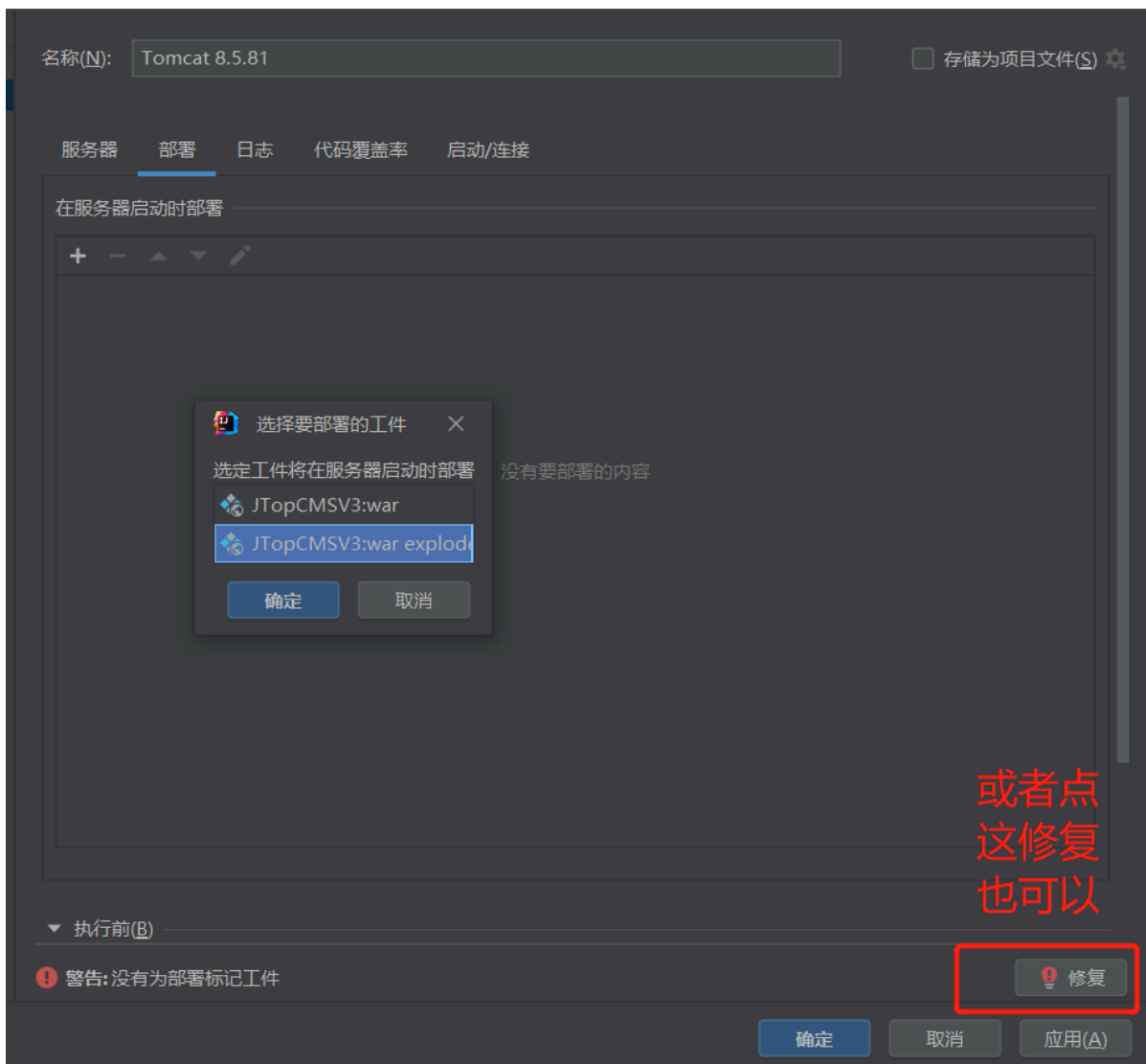
④、下面配置 Tomcat 安装路径，点击配置后，在 Tomcat 主目录中选择你所安装的 Tomcat 路径，如下图所示：



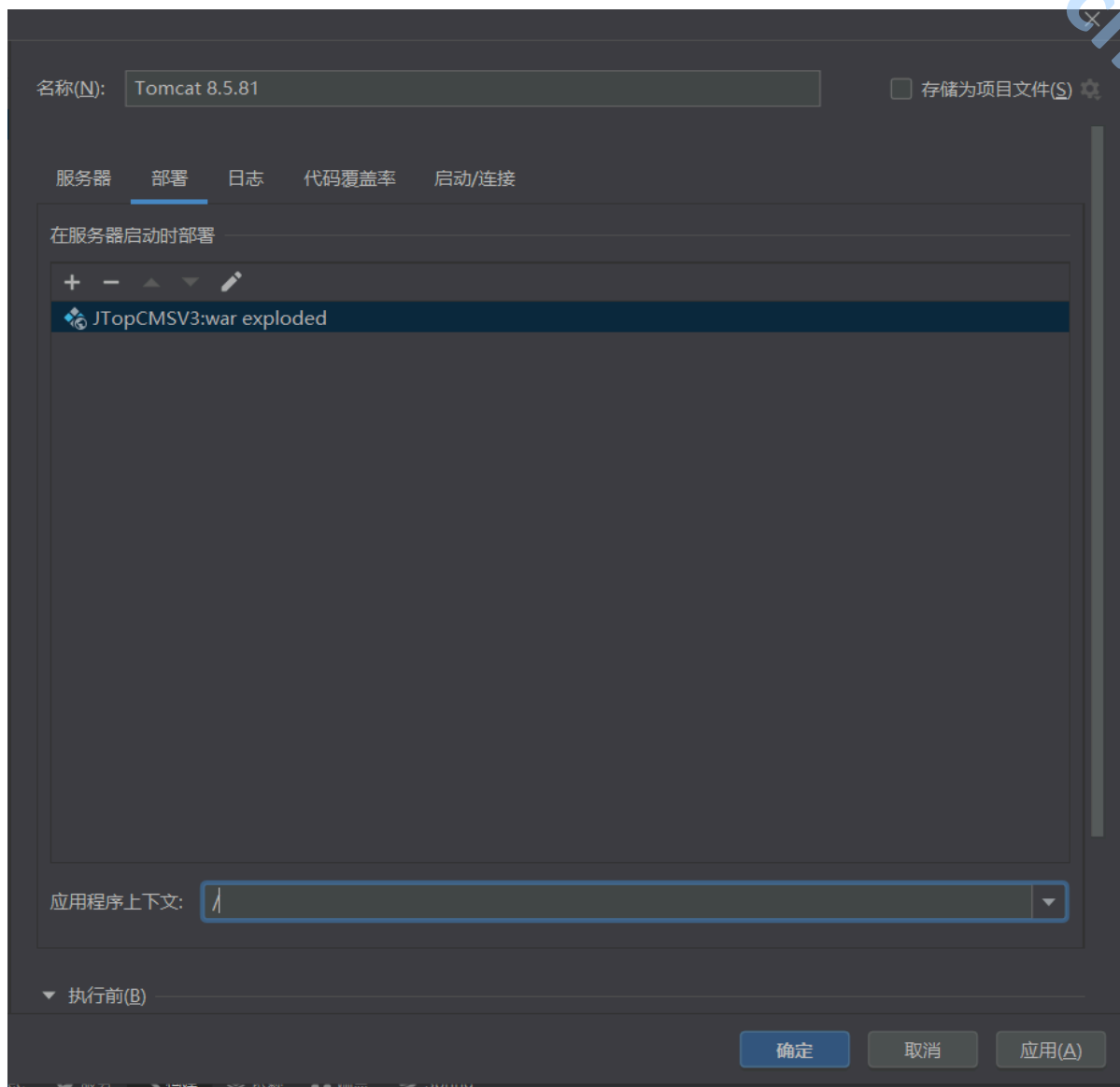
⑤、防止与其他软件冲突，Tomcat 服务器设置下端口号，自己随意。注意！并且勾选端口号旁边的一个选项，中文是 部署 Tomcat 实例中配置的应用程序 如下图所示：



⑥、部署下工件，点击 部署，点击加号，选择工件，选择部署为 war_exploded，如下图所示：



⑦、修改下应用程序上下文也就是URL访问路径，删除默认路径，如下图所示：

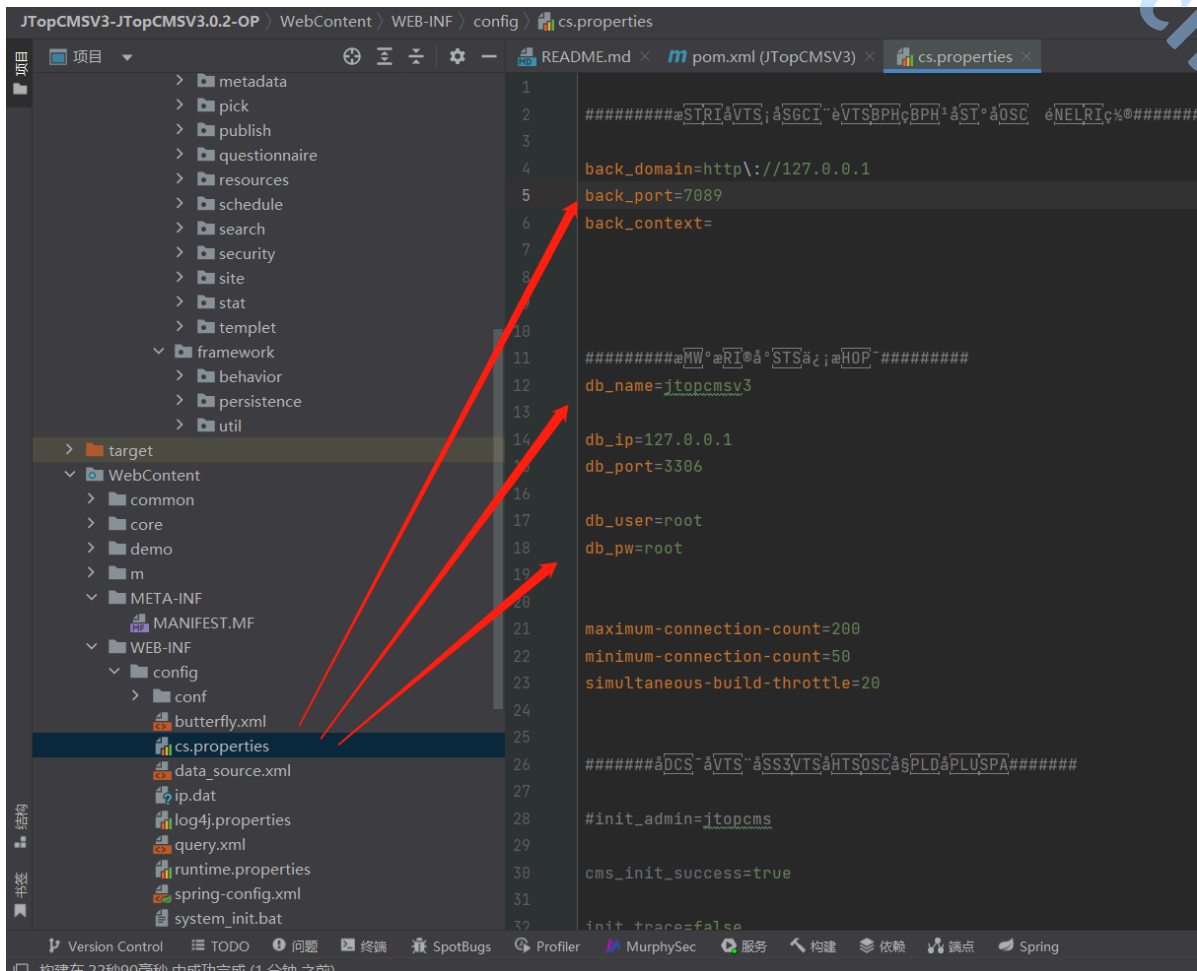


⑧、修改下项目中配置文件信息，`cs.properties` 为项目配置文件，位于 `CJTopCMSV3-JTopCMSV3.0.2-OP\WebContent\WEB-INF\config\cs.properties`。

修改 `back_port` 为访问端口号，要与 Tomcat 中设置的端口号一致

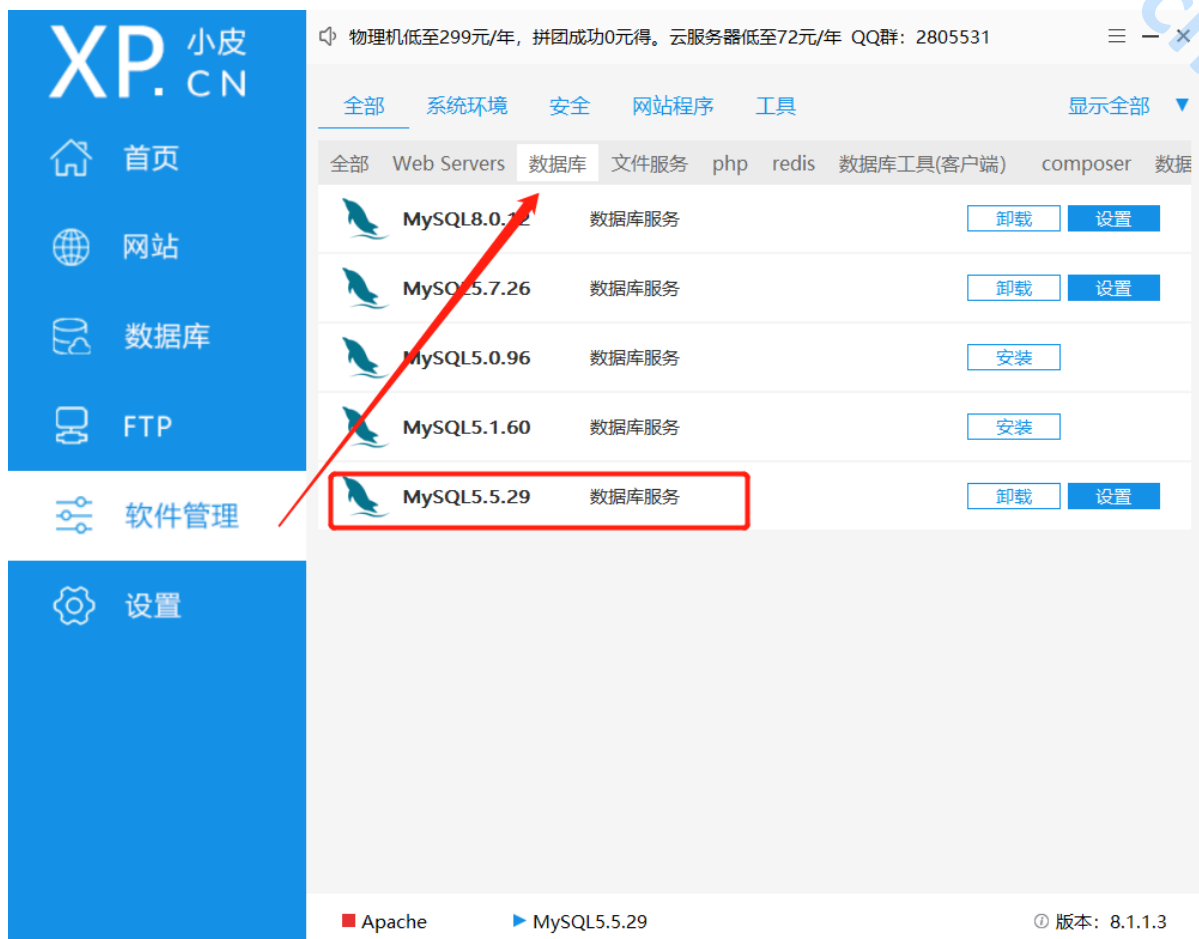
修改 `db_name` 为 `jtopcmsv3`。

修改 `db_pw` 为数据库密码。如下图所示：



1.4.2、Mysql导入数据

①、打开PHPstudy，在 软件管理-数据库 下自行安装 Mysql15.5.29 版本，如下图所示：



②、在首页处启动Mysql5.5.29。然后命令行进入Mysql。

创建一个名为 jtopcmsv3 的数据库。

进入 jtopcmsv3 数据库。

如下图所示：

```
GoodLuckToday$$$ mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 105
Server version: 5.5.29 MySQL Community Server (GPL)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

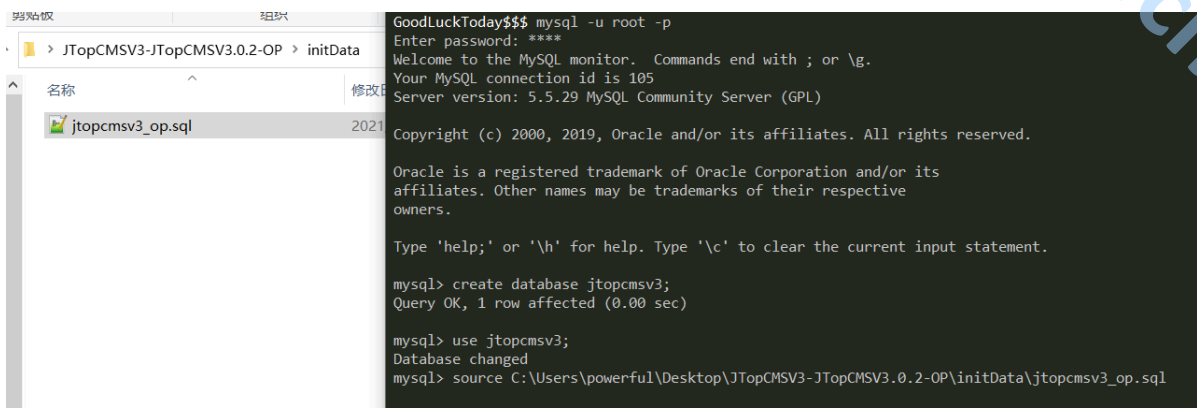
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database jtopcmsv3;
Query OK, 1 row affected (0.00 sec)

mysql> use jtopcmsv3;
Database changed
mysql>
```

③、导入数据，数据文件在 \JTopCMSV3-JTopCMSV3.0.2-OP\initData 目录下，使用source命令导入。回车即可。如下图所示：



注意：如果路径是反斜杠尾部可以不用加分号；，如果路径是正斜杠需要加分号；，否则会报错。

至此，配置完毕。

IDEA启动项目。

后端管理地址：http://127.0.0.1:7089/login_page，登录账号密码为 admin/jtopcms。

前台地址需要在后端管理的 站点维护-站点与栏目维护 下进行配置后才可以访问。

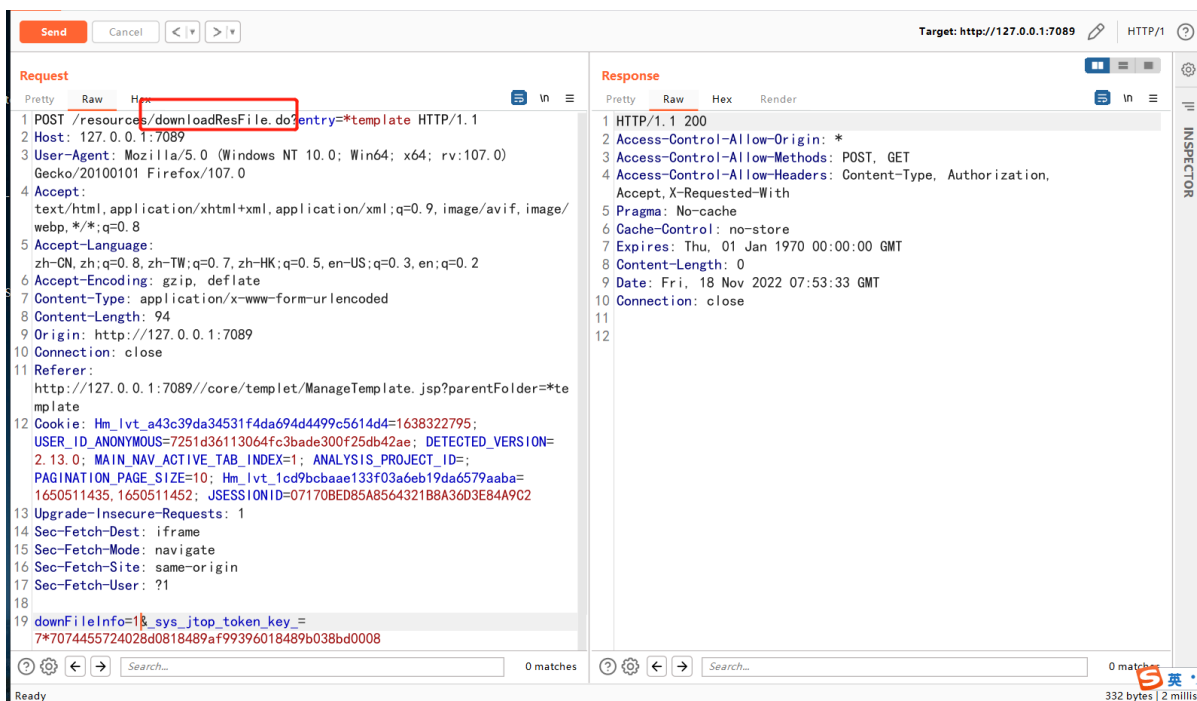
2、目录穿越漏洞

这个系统存在下载文件功能，但开发人员对下载目录和下载文件名做了限制，以及限制读取/下载 WEB-INF 目录文件。

如何确定的下载功能？

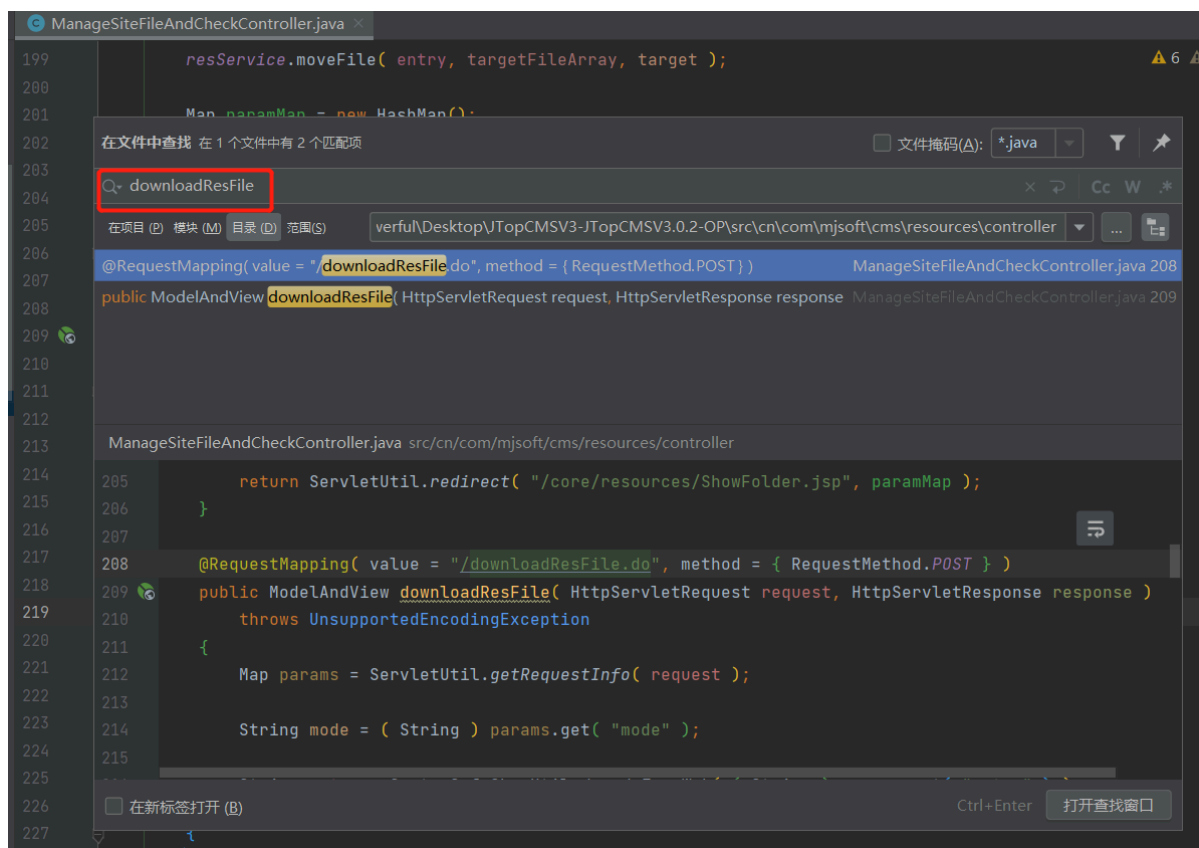
本项目中文档文件下用户手册。部署环境后通过抓取下载数据包确定路径后全局搜索 `downloadResFile`。使用经典关键字 `download` 等都可以确定存在下载功能。

代码太多了，节约时间。我是通过部署环境后抓取下载功能数据包，在代码中全局搜索路径确定的下载文件代码，如下图所示：

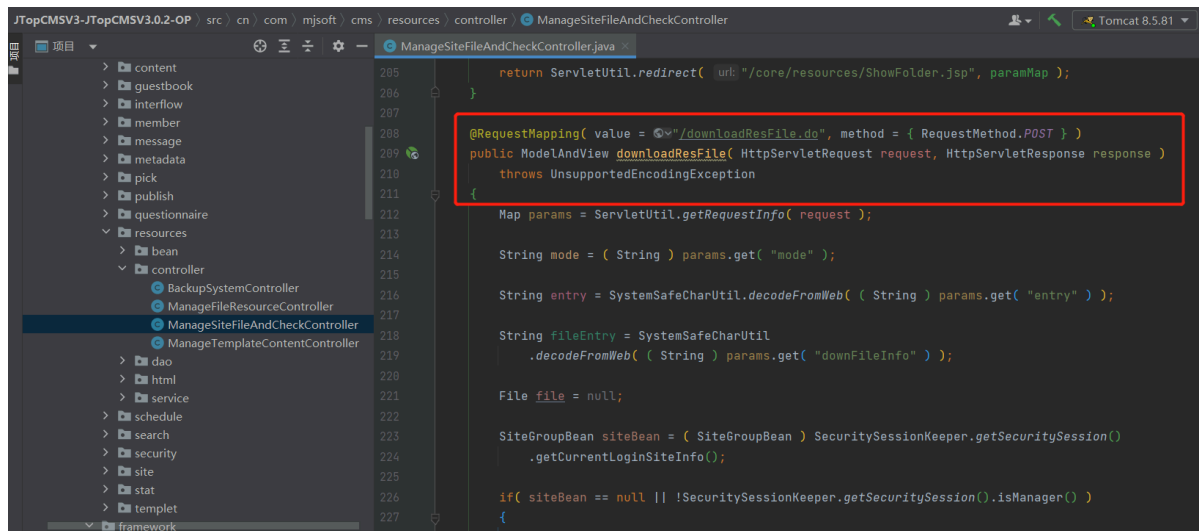


根据下载路径 `downloadResFile` 全局搜索，最终定位下载文件功能代码位于 `JTopCMSV3-JTopCMSV3.0.2-`

`OP\src\cn\com\mjsoft\cms\resources\controller\ManageSiteFileAndCheckController.java` 中。如下图所示：



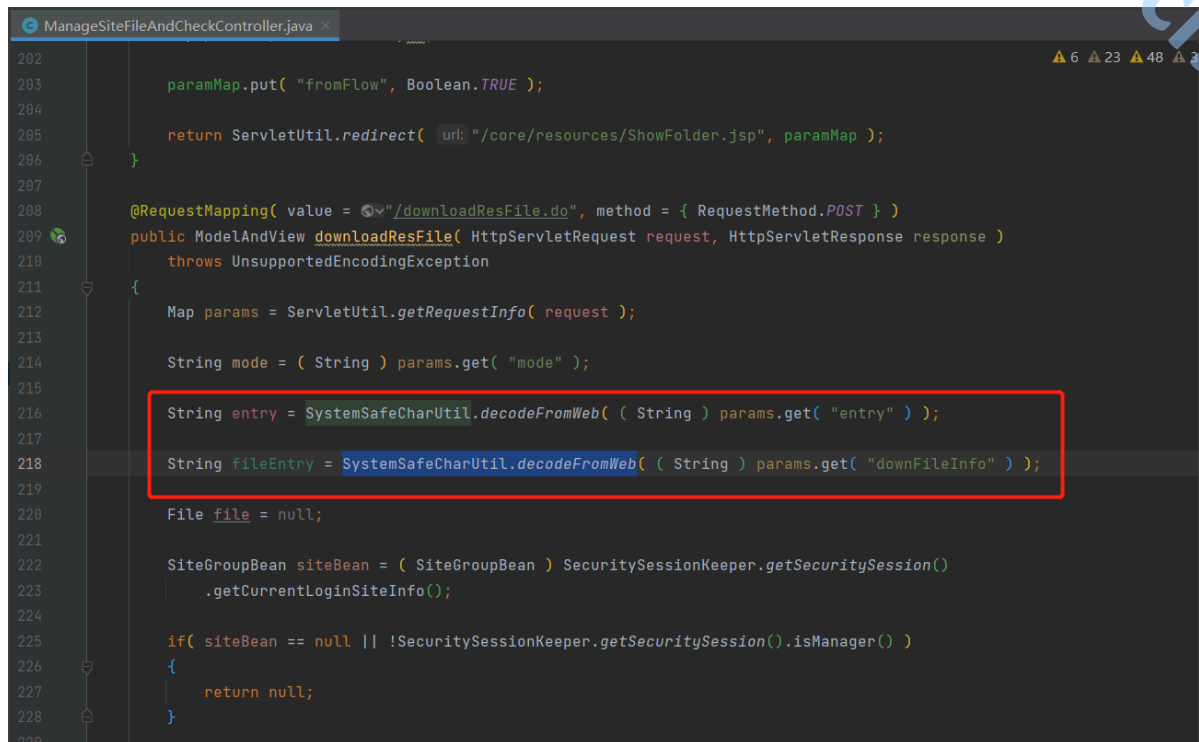
点击进入 `ManageSiteFileAndCheckController.java` 代码文件。第208行到第311行为下载功能代码，如下图所示：



2.1、代码审计分析

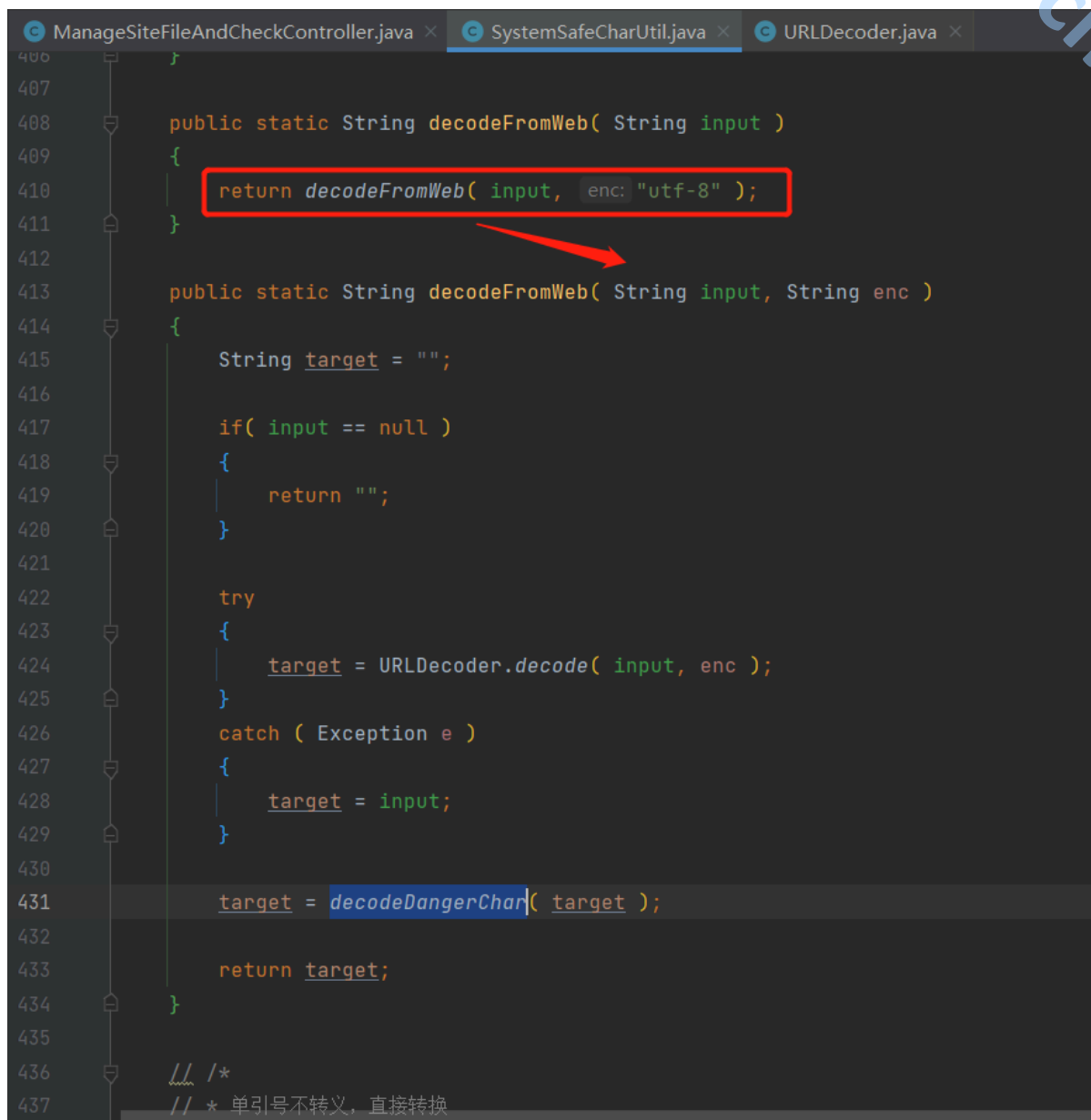
开始分析代码，按照我当时的思路写下来的，部分步骤省略，代码不难，自行分析即可。

①、通过BurpSuite抓包也可以看到传入了三个参数，主要关注 `entry` 和 `downFileInfo` 这两个，刚看到时感觉就跟路径和文件名有关。在代码中分别第216行和218行处接受了这两个参数，并且使用了 `SystemSafeCharUtil.decodeFromWeb()` 自己写的类对参数进行了处理，如下图所示：



```
202
203     paramMap.put( "fromFlow", Boolean.TRUE );
204
205     return ServletUtil.redirect( url: "/core/resources/ShowFolder.jsp", paramMap );
206 }
207
208 @RequestMapping( value = "/downloadResFile.do", method = { RequestMethod.POST } )
209 public ModelAndView downloadResFile( HttpServletRequest request, HttpServletResponse response )
210     throws UnsupportedEncodingException
211 {
212     Map params = ServletUtil.getRequestInfo( request );
213
214     String mode = ( String ) params.get( "mode" );
215
216     String entry = SystemSafeCharUtil.decodeFromWeb( ( String ) params.get( "entry" ) );
217     String fileEntry = SystemSafeCharUtil.decodeFromWeb( ( String ) params.get( "downFileInfo" ) );
218
219     File file = null;
220
221     SiteGroupBean siteBean = ( SiteGroupBean ) SecuritySessionKeeper.getSecuritySession()
222         .getCurrentLoginSiteInfo();
223
224     if( siteBean == null || !SecuritySessionKeeper.getSecuritySession().isManager() )
225     {
226         return null;
227     }
228 }
```

②、跟进 `systemSafeCharUtil.decodeFromWeb`，进入 `SystemSafeCharUtil.java` 工具类，在第408到434行对参数进行了处理。判断了是否为空，并将传入的参数在做了解码操作后赋值给了 `target`，最后 `target` 又做了一步 `decodeDangerChar` 操作，如下图所示：



```
406 }
407
408 public static String decodeFromWeb( String input )
409 {
410     return decodeFromWeb( input, enc: "utf-8" );
411 }
412
413 public static String decodeFromWeb( String input, String enc )
414 {
415     String target = "";
416
417     if( input == null )
418     {
419         return "";
420     }
421
422     try
423     {
424         target = URLDecoder.decode( input, enc );
425     }
426     catch ( Exception e )
427     {
428         target = input;
429     }
430
431     target = decodeDangerChar( target );
432
433     return target;
434 }
435
436 /**
437  * 单引号不转义，直接转换
```

③、继续跟进 `decodeDangerChar` 方法，对target做了一些替换操作，有点奇怪当时也没多想，反正没有过滤 `../` 敏感字符，也就放过去了，继续了后面的跟踪，但其实这是绕过目录穿越限制的关键地方，下面再说。代码如下图所示：


```

498 public static String decodeDangerChar( String input )
499 {
500     if( StringUtil.isStringNull( input ) || !haveDangerChar( input ) )
501     {
502         return input;
503     }
504     String target = input;
505     target = StringUtil.replaceString( target, replacelt: "***1**", replacement: "*", toLowerCase: false, prefixMode: false );
506     target = StringUtil.replaceString( target, replacelt: "***2**", replacement: "(", toLowerCase: false, prefixMode: false );
507     target = StringUtil.replaceString( target, replacelt: "***3**", replacement: ")", toLowerCase: false, prefixMode: false );
508     target = StringUtil.replaceString( target, replacelt: "***4**", replacement: "..", toLowerCase: false, prefixMode: false );
509     // target = StringUtil.replaceString( target, "***5**", "\\", false,
510     // false );
511     target = StringUtil.replaceString( target, replacelt: "***6**", replacement: "\"", toLowerCase: false, prefixMode: false );
512     // target = StringUtil.replaceString( target, "***7**", "\\\"", false,
513     // false );
514     target = StringUtil.replaceString( target, replacelt: "***8**", replacement: "<", toLowerCase: false, prefixMode: false );
515     target = StringUtil.replaceString( target, replacelt: "***9**", replacement: ">", toLowerCase: false, prefixMode: false );
516     target = StringUtil.replaceString( target, replacelt: "***10**", replacement: "|", toLowerCase: false, prefixMode: false );
517     target = StringUtil.replaceString( target, replacelt: "***11**", replacement: "\\\"", toLowerCase: false, prefixMode: false );
518     target = StringUtil.replaceString( target, replacelt: "***12**", replacement: "+", toLowerCase: false, prefixMode: false );
519     // target = StringUtil.replaceString( target, "***13**", ";", false,
520     // false );
521     target = StringUtil.replaceString( target, replacelt: "***14**", replacement: "@", toLowerCase: false, prefixMode: false );
522     target = StringUtil.replaceString( target, replacelt: "***15**", replacement: "$", toLowerCase: false, prefixMode: false );
523     target = StringUtil.replaceString( target, replacelt: "***16**", replacement: ":", toLowerCase: false, prefixMode: false );
524     // target = StringUtil.replaceString( target, "***17**", "/", false,
525     // false );
526     target = StringUtil.replaceString( target, replacelt: "***18**", replacement: " a", toLowerCase: false, prefixMode: false );
527     target = StringUtil.replaceString( target, replacelt: "***19**", replacement: " A", toLowerCase: false, prefixMode: false );
528     target = StringUtil.replaceString( target, replacelt: "***20**", replacement: "/*", toLowerCase: false, prefixMode: false );

```

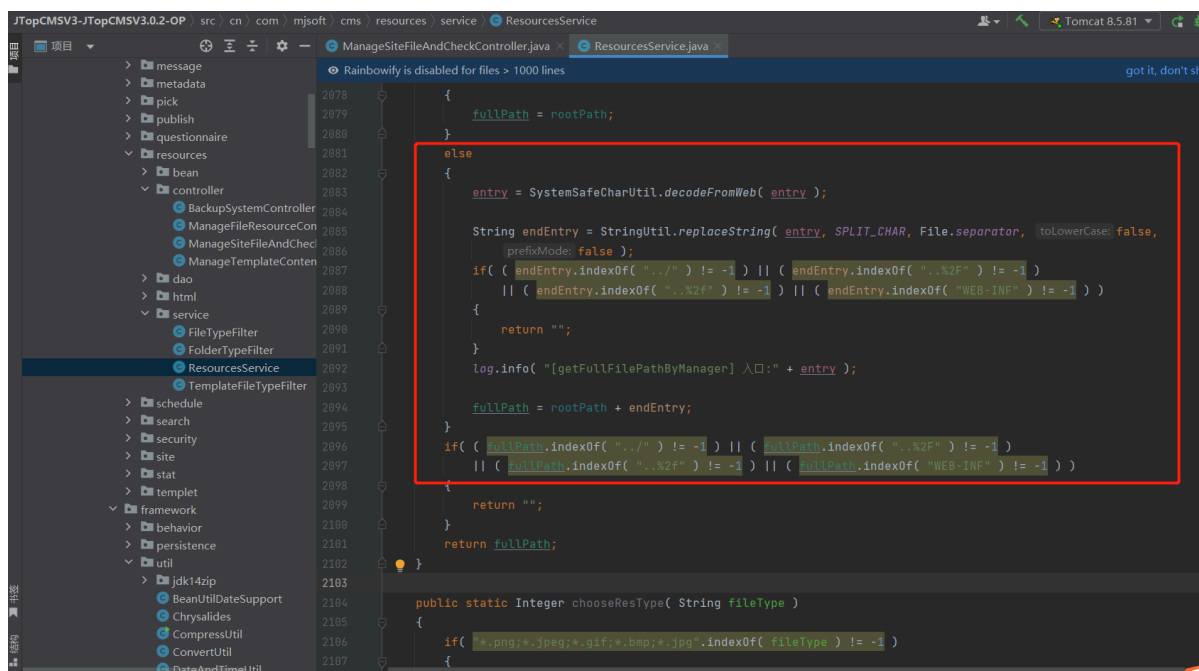
④、回到ManageSiteFileAndCheckController.java 继续分析。第233行使用 tempRoot + ".zip"; 拼接了下载文件的名称赋值给 zipName，双击选中 zipName 跟踪发现在第265行处使用到了 zipName，如下图所示：

```

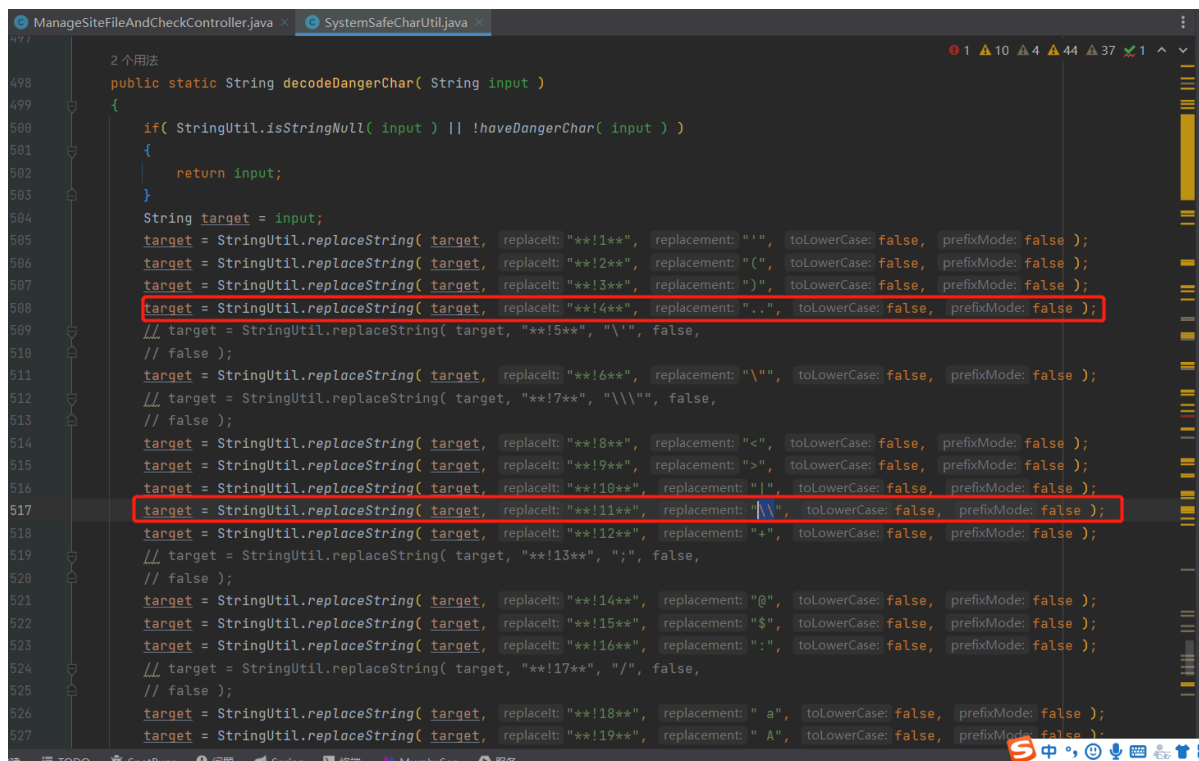
219
220     File file = null;
221
222     SiteGroupBean siteBean = ( SiteGroupBean ) SecuritySessionKeeper.getSecuritySession().getCu
223
224     if( siteBean == null || !SecuritySessionKeeper.getSecuritySession().isManager() )
225     {...}
226
227
228
229     String tempRoot = siteBean.getSiteRoot() + "_"
230         + DateAndTimeUtil.getCurrentDayAndTime( DateAndTimeUtil.DEAULT_FORMAT_YMD ) + "_"
231         + DateAndTimeUtil.clusterTimeMillis();
232
233     String zipName = tempRoot + ".zip";
234
235     if( "allTemplate".equals( mode ) )
236     {...}
237     else
238     {
239
240         String fullZipPath = resService.getFullFilePathByManager( entry: "*"
241             + Constant.CONTENT.TEMPLATE_TEMP_BASE + "*" + zipName );
242
243         if( fullZipPath == null )
244         {
245             return null;
246         }
247
248         String[] targetFileArray = StringUtil.split( fileEntry, splitChar: "\\*" );
249
250         String fileFullPath = null;
251
252         String fileFullPath = null;

```

⑤、跟进 `resService.getFullFilePathByManager` 进行分析，进入到 `ResourcesService.java` 代码中，位于 `JTopCMSV3-JTopCMSV3.0.2-OP\src\cn\com\mjsoft\cms\resources\service\ResourcesService.java`。在第2085行到2102行过滤了敏感字符，如下图所示：



⑥、过滤的挺全面的，考虑的挺周到的，在这绕了一会没绕过去。然后想到了当时奇奇怪怪的替换字符，研究了下发现，如果路径中/文件名中存在那些字符，则会替换成对应的字符。代码如下图所示：

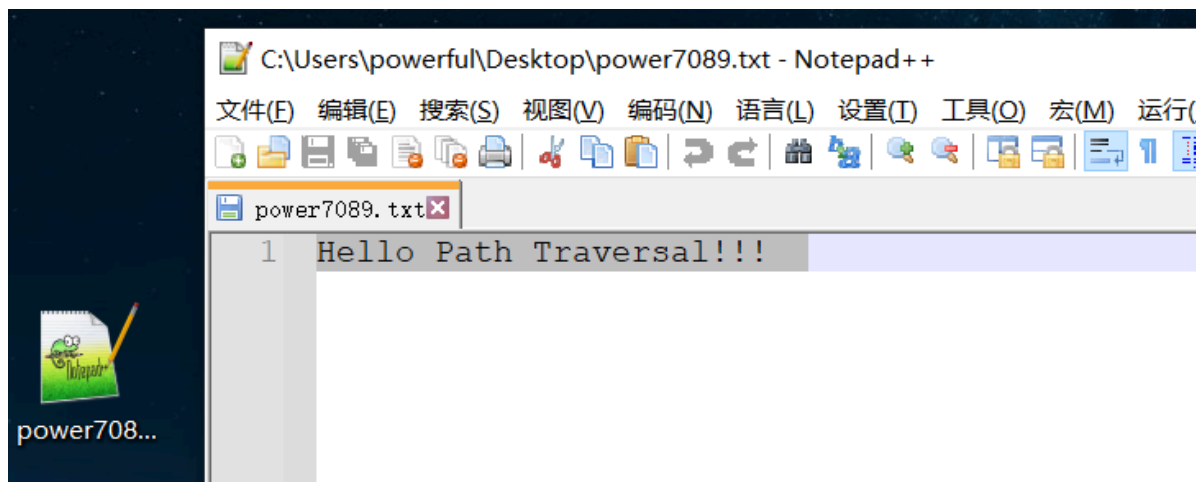


通过代码可以得到如果路径中存在 `**!4**` 则会被替换成 `..`，`**!11**` 会被替换成 `\\`。这样说的话路径中存在 `**!4****!11**` 的话就可以向上遍历目录了。下面渗透测试验证一下。

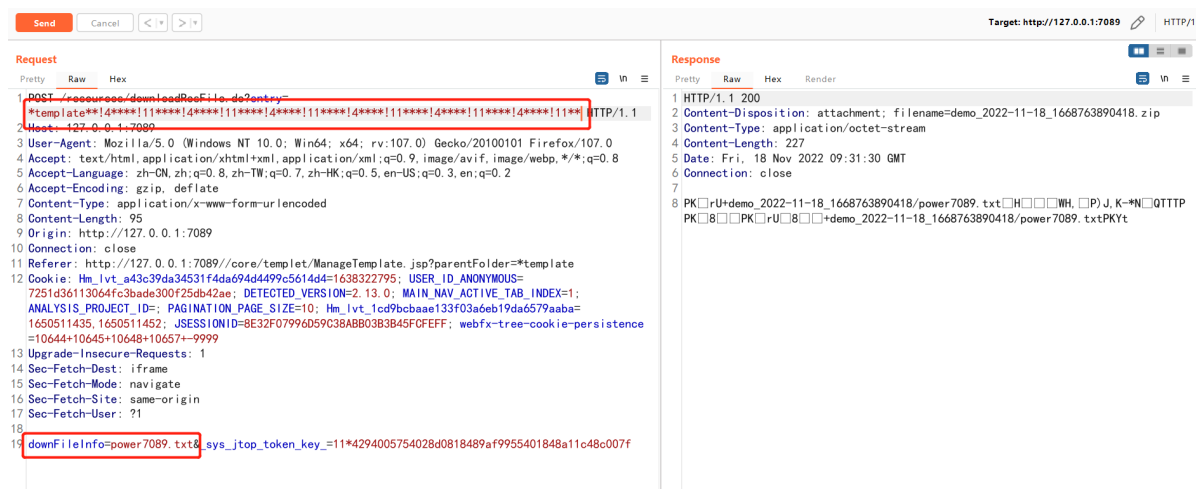
2.2、渗透测试验证

渗透测试验证在Windows环境下。

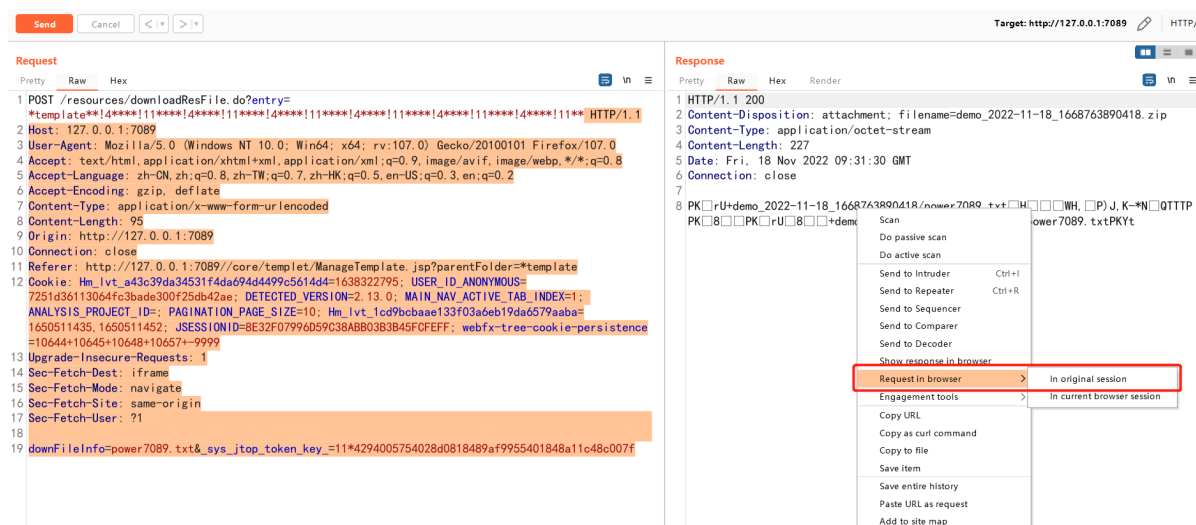
①、先在桌面下新建一个名为 power7089.txt 的文件，内容随意，如下图所示：

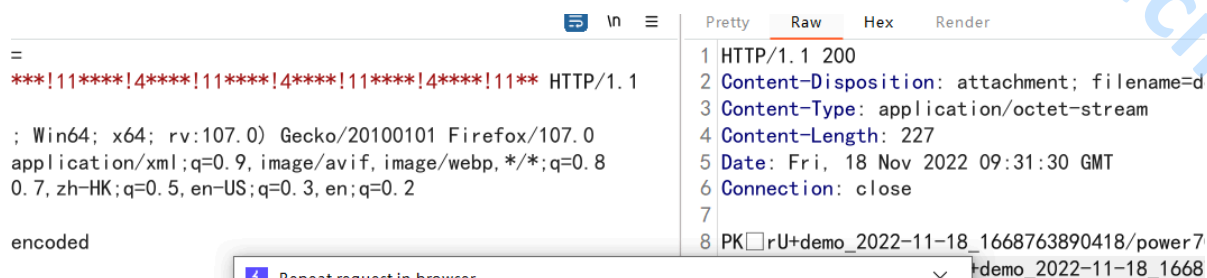


②、进入JTopCMS内容管理系统后台，访问 站点维护-模板管理 功能，选择任意一个文件夹后点击下载，此时使用BurpSuite进行抓包，丢入Repeater模块下，将 downFileInfo 改为 power7089.txt，在 entry 后面加上 **!4****!11**，经过多次尝试最终穿越当前目录，读取到了桌面上的 power7089.txt 文件，如下图所示：



③、在响应处右键选择 Request in browser - in original session，copy链接，如下图所示：

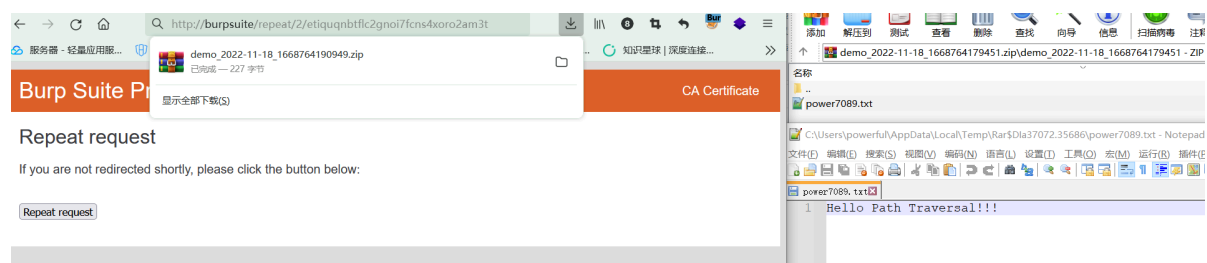




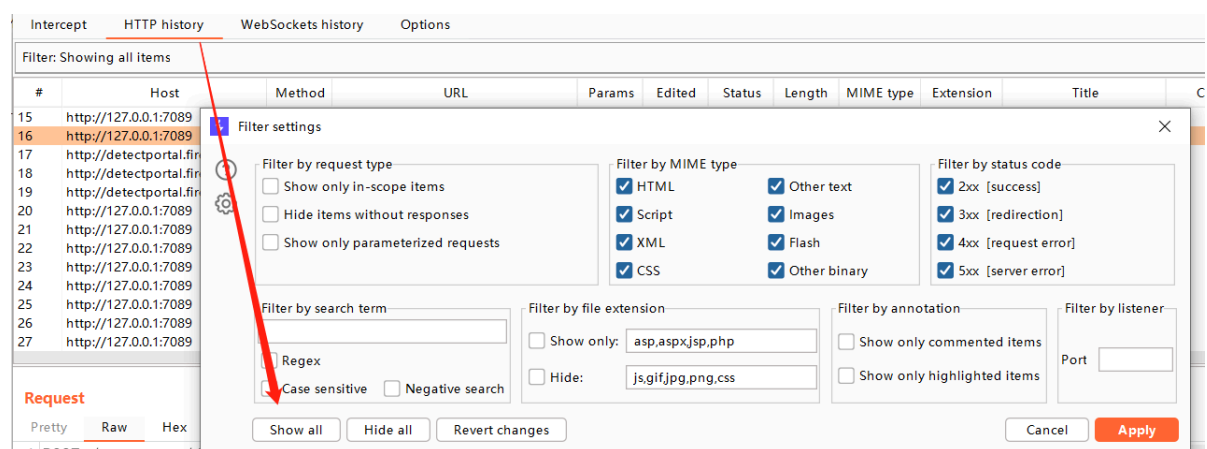
plet/ManageTemplate.
c5614d4=1638322795;
ED_VERSION=2.13.0; M
ZE=10; Hm_lvt_1cd9bc
7996D59C38ABB03B3B45

en key =11*4294005754028d0818489af9955401848a11c48c007f

④、需要在浏览器开着BurpSuite代理的情况下访问上述链接，最终成功下载了桌面上的power7089.txt 文件，如下图所示：



(备注：如果抓不到数据包，可以试下在HTTP history中点击show all显示全部，因为下载的MIME type是压缩。文件如下图所示：)



至此，目录穿越漏洞讲解完毕。其中JTopCMS审计过程仅提炼了几个重要的点来讲解，因为代码不难，大家自行调试分析学习一下吧。代码审计自己动手分析的每一个过程都很重要！不懂的地方可以将问题整理完成后在星球里面向我提问！