Week 3

Authentication and Authorization

What is Authentication Bypass how does it work?

It happens when its possible to break authentication (log as other user, privilged or signed), either by hidden input, or removing parameters or brute force

Why is it important to prevent Authetn?

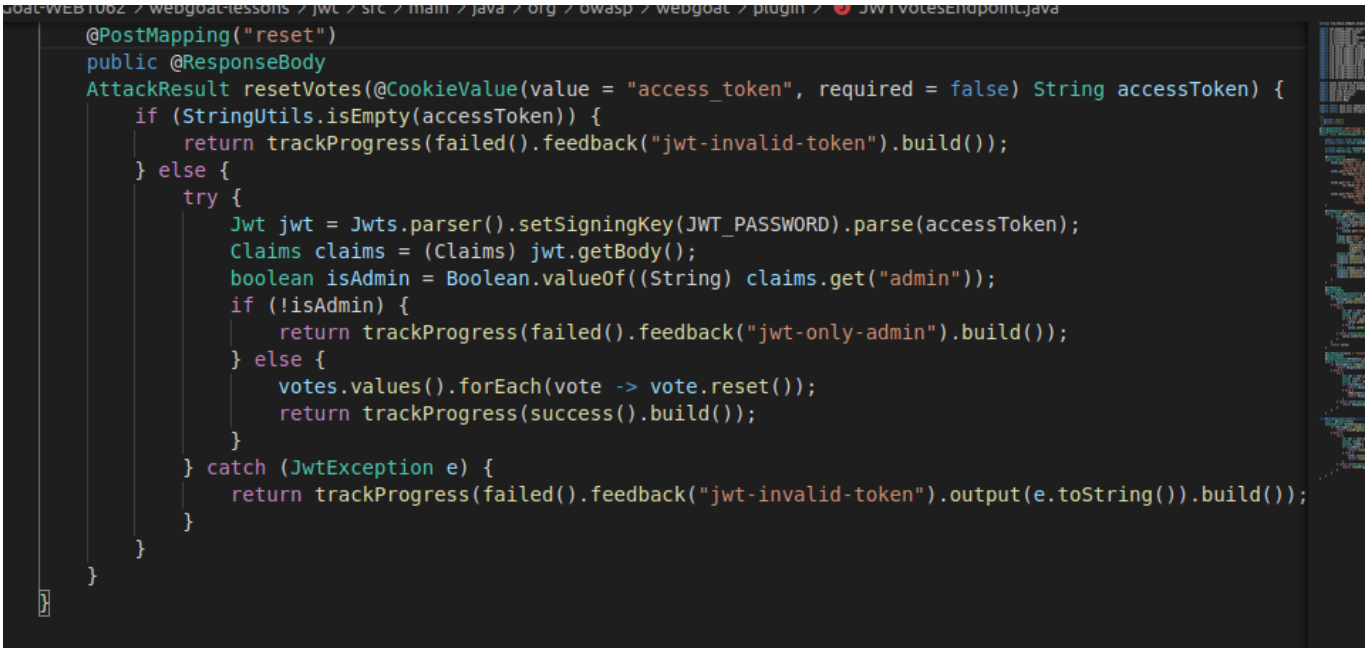
To maintain user confidentialy and not give admin rights, eventually

```
public boolean verifyAccount(Integer userId, HashMap<String,String> submittedQuestions ) {
    //short circuit if no questions are submitted
    if (submittedQuestions.entrySet().size() != secQuestionStore.get(verifyUserId).size()) {
        return false;
    }

    if (submittedQuestions.containsKey("secQuestion0") && !submittedQuestions.get("secQuestion0").equals(secQuestionStore.get(veri
        return false;
    }

    if (submittedQuestions.containsKey("secQuestion1") && !submittedQuestions.get("secQuestion1").equals(secQuestionStore.get(veri
        return false;
    }

    // else
    return true;

}
```

Patch

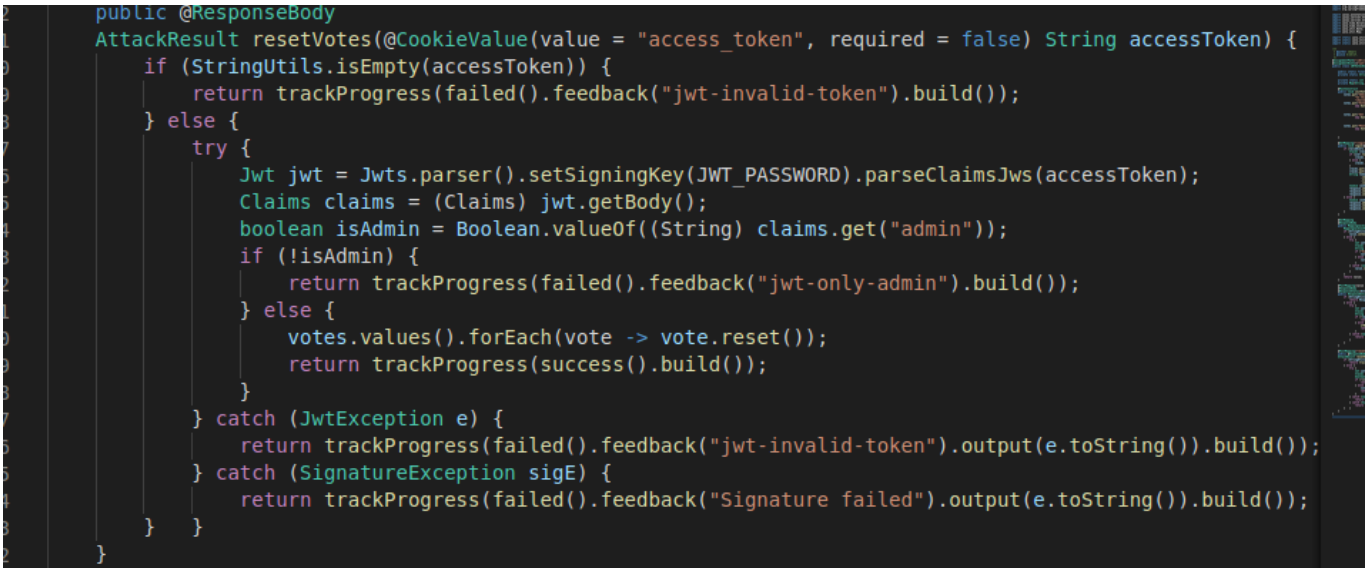
```
//        return true;

    if (submittedQuestions.entrySet().size() != secQuestionStore.get(verifyUserId).size()) {
        return false;
    }

    if (submittedQuestions.containsKey("secQuestion0") && !submittedQuestions.get("secQuestion0").equals(secQuestionStore.get(ve
        return true;
    }

    if (submittedQuestions.containsKey("secQuestion1") && !submittedQuestions.get("secQuestion1").equals(secQuestionStore.get(ve
        return true;
    }

    return false

}
```

JWT



```java
@PostMapping("reset")
public @ResponseBody
AttackResult resetVotes(@CookieValue(value = "access_token", required = false) String accessToken) {
    if (StringUtils.isEmpty(accessToken)) {
        return trackProgress(failed().feedback("jwt-invalid-token").build());
    } else {
        try {
            Jwt jwt = Jwts.parser().setSigningKey(JWT_PASSWORD).parse(accessToken);
            Claims claims = (Claims) jwt.getBody();
            boolean isAdmin = Boolean.valueOf((String) claims.get("admin"));
            if (!isAdmin) {
                return trackProgress(failed().feedback("jwt-only-admin").build());
            } else {
                votes.values().forEach(vote -> vote.reset());
                return trackProgress(success().build());
            }
        } catch (JwtException e) {
            return trackProgress(failed().feedback("jwt-invalid-token").output(e.toString()).build());
        }
    }
}
```

Patch



```java
public @ResponseBody
AttackResult resetVotes(@CookieValue(value = "access_token", required = false) String accessToken) {
    if (StringUtils.isEmpty(accessToken)) {
        return trackProgress(failed().feedback("jwt-invalid-token").build());
    } else {
        try {
            Jwt jwt = Jwts.parser().setSigningKey(JWT_PASSWORD).parseClaimsJws(accessToken);
            Claims claims = (Claims) jwt.getBody();
            boolean isAdmin = Boolean.valueOf((String) claims.get("admin"));
            if (!isAdmin) {
                return trackProgress(failed().feedback("jwt-only-admin").build());
            } else {
                votes.values().forEach(vote -> vote.reset());
                return trackProgress(success().build());
            }
        } catch (JwtException e) {
            return trackProgress(failed().feedback("jwt-invalid-token").output(e.toString()).build());
        } catch (SignatureException sigE) {
            return trackProgress(failed().feedback("Signature failed").output(e.toString()).build());
    }   }
}
```