broken authentication

به طور کلی وقتی می خواهیم به اطلاعات غیر سری مثل ویژگی های محصول و ... نیازی به توکن ندارد و با صرفا با توکن "no token" می شود به ان اطلاعات درسترسی دارد و اما برای دسترسی های بیشتری استرینگ ای هنگام لاگین داده می شود:

در مدل کلاسی به نام token یا identification ID وجود دارد طرز کلی کار کردن token در پروژه این گونه است: هر بار که شخصی لاگین میکند یک token دریافت میکنم که در آن اطلاعات زمان ساخت توکن و یوزرنیم کاربر به صورت رمزی وجود دارد و طرز رمز آن فقط در سرور وجود دارد و خود کلاینت هم به آن دسترسی ندارد. یعنی اینکه ***توکن نمی شود برای کاربر دیگری*** استفاده شود و می تواند ***پس از مدتی منقضی*** شود و سپس به کمک تابع دیکدر میتوانیم اطلاعات را استخراج کنیم.

```
private static final SecureRandom secureRandom = new SecureRandom();
private static final Base64.Encoder base64Encoder = Base64.getUrlEncoder();

public static String generateNewToken(String username) {
    byte[] randomBytes = new byte[100];
    secureRandom.nextBytes(randomBytes);
    String s = System.currentTimeMillis() + "--caption neuer--" + username + "--caption neuer--" + base64Encoder.encodeToString(randomBytes);
    String token = base64Encoder.encodeToString(s.getBytes());
    token = new StringBuilder(token).reverse().toString();
    token = base64Encoder.encodeToString(token.getBytes());
    Token.token.add(token);
    return token;
}

public static String decode(String token) {
    token = new String(Base64.getDecoder().decode(token));
    token = new StringBuilder(token).reverse().toString();
    token = new String(Base64.getDecoder().decode(token));
    return token;
}
```

برای مثال یک خروجی نمونه ان به صورت زیر است:

PT1RUDlFa2RSSlVlSFJuYndObVlLRlVXMFFrTjRSamE1ZGtVR1IwWkNKRldISjFVdEVYU3dvV2RPcEZ
PVzlrZXdOalRxcEhkVFJrV3Z0MlZ6VjFOcjF5WklsWFNXNW1hYVZEVXljRFp2TjJOVlpqV3hRVWFXVlhU
UmxFY1pKRmU1bEdOUFJGZWpSWFNJWjNRSHRVTzJFVVY1VkVPeUFuVEdGVVFqbG1iaWhFTmps
RU4yRkVVeGhtY3QwaWNsVlhadUJpYnZsR2R3RjJZdDBpYnBWMmN6OUdhdDBpY2xWWFp1Qmlidmx
HZHdGMll0MENOeVFqTTVNRE4xUVRONVVUTQ==

و واضح است که به حدس زدن ان با توجه به اینکه طول مشخصی ندارد (رشته اولیه توسط SecureRandom تولید میشود و سپس endoce میشود) و کاملا رندم است غیر ممکن است بدین ترتیب هر یوزر توکن مشخصی هنگام لاگین دریافت میکند و نوع پیام ها به سرور به این شکل باید باشد که رمز شده است:

```
String result = "this is a client" + "--1989--" + MenuHandler.getToken() + "--1989--" +
        command + "--1989--" + System.currentTimeMillis() + "--1989--" +
        ((MenuHandler.getUsername() == null) ? "no username" : MenuHandler.getUsername());
```

نخست باز هم واضح است که بدون داشتن الگوریتم رمز کردن پیام غیر ممکن است پس فرض کنیم که کد های کلاینت در دسترس است که یعنی می تواند زمان ارسال پیام را تغییر دهد.

دوما identifier string به این دلیل است که اگر ریکوئست با ان شروع نشده بود اصلا بررسی نخواهد شد و اگر از خارج استرینگ رندمی داده شود و با این شروع نشده بود کاملا ایگنور می شود.

```
if (!command.startsWith("this is a client")) {
    blackListOfIPs.add(getIP(socket));
    return;
}
```

سپس توکن رو بررسی میکنیم اولا اگر token اشتباه بود پیام را نادید گرفته و IP مورد نظر هم به لیست سیاه IP ها اضافه می شوند و سپس اگر اشتباه نبود زمان ان چک می شود که اگر تازه ارسال نشده بود اگنور یا اگر متعلق به 1000 ثانیه قبل تر (حدود 15 دقیقه) بوده باشد و کاربر به طور اتوماتیک logout می شود و توکن ش می سوزد.

```
String token, message, username;
long time_sent;

try {
    token = command.split( regex: "--1989--")[1];
    message = command.split( regex: "--1989--")[2];
    time_sent = Long.parseLong(command.split( regex: "--1989--")[3]);
    username = command.split( regex: "--1989--")[4];
} catch (Exception e) {
    System.out.println("we're under attackkkkkkk");
    blackListOfIPs.add(getIP(socket));
    return;
}

//the time is old

if (System.currentTimeMillis() - time_sent > 100) {
    blackListOfIPs.add(getIP(socket));
    return;
}
```

در غیر این صورت پیام بررسی می شود و با یوزر نیم ارسالی مطابقت داده می شود و اگر کسی هم که به کلاینت دسترسی کامل دارد اگر سعی کند پیام زیاد بفرستد به لیست سیاه ip ها وارد می شود که در قسمت دیگر بررسی شده و فقط نکته مهم در اینجا این است که توکن خودش را فرد دارد پس به جز اطلاعات خودش نمیتواند کار دیگری را بکند.

```
//making sure it doesn't send more than 10 requests in 100 milliseconds

if (IP.addIp(getIP(socket))) {
    blackListOfIPs.add(getIP(socket));
    return;
}
```

```
public static boolean addIp(String Ip) {
    IP ip = getIp(Ip);
    if (ip == null) {
        new IP(Ip);
        return false;
    }
    if (ip.ip.equals(Ip)) {
        if (ip.tenRecentTimes.size() >= BOUND) {
            ip.tenRecentTimes.remove( index: 0);
            ip.tenRecentTimes.add(System.currentTimeMillis());
            return ip.tenRecentTimes.get(BOUND-1) - ip.tenRecentTimes.get(0) < 100;
        } else {
            ip.tenRecentTimes.add(System.currentTimeMillis());
        }
    }
    return false;
}
```

و این متود برای بررسی expire نشدن است:

```
public static boolean hasTokenExpired(String token) {
    try {
        token = decode(token);
        long past = Long.parseLong(token.substring(0, 13));
        boolean flag = System.currentTimeMillis() - past > 1000000;
        if (flag) {
            Token.token.remove(token);
        }
        return flag;
    } catch (Exception e) {
        return false;
    }
}
```

حذف کردن توکن:

```
public void deleteToken(String username) {
    token.remove(username);
}
```

توجه کنید که هنگام لاگین کردن توکن داده می شود و نه هنگام ثبت نام و توکن ها در فایلی ذخیره نمی شوند و برای یک فرد هم ثابت نیستند پس بنا بر این امکان لو رفتن انها تنها منوط به دست رسی کامل به کلاس Token از مدل است و راه دیگری ندارد بنا بر این امنیت کامل دارد و چون سریع هم عوض میشود (هنگام logout یا پس از 1000 ثانیه) به لحاظ امنیتی قابل اعتماد میباشد ...

```
if (Token.isTokenValid(token)) {

    try {
        String decodedToken = Token.decode(token);
        long time = Long.parseLong(decodedToken.split( regex: "--caption neuer--")[0]);
        Account account = Storage.getAccountWithUsername(decodedToken.split( regex: "--caption neuer--")[1]);
        if (!Token.getUsernameFromToken(token).equals(account.getUsername())) {
            throw new Exception("piss off");
        }
    } catch (Exception e) {
        System.out.println("we're under attack by trying wring tokens");
        blackListOfIPs.add(getIP(socket));
        return;
    }

    //checking that it's still authentic

    if (!Token.hasTokenExpired(token)) {
        Server.server.takeAction(message);
        Token.addOnlineUsers(Token.getUsernameFromToken(token), System.currentTimeMillis());
    } else {
        Server.server.takeAction( command: "token has expired");
        Token.addOnlineUsers(Token.getUsernameFromToken(token), (long) -1);
    }

}
```

یک کار دیگری هم که موقع ثبت نام انجام می شود چک می شود که پسورد حداقل دارای طول 8 و شامل عدد و رقم باشد که امکان لو رفتن پسورد و جا زدن به جای کس دیگر را ضعیف تر میکند که البته در brute force هم بررسی می شود. و در آنجا هم گفته میشود که هر کاربر ip ش را هم نگه می داریم و اجازه داده نمی شود کسی با ای پی متفاوت به اکانت فرد دیگری وصل شود.

```java
public void logout() {
    Alert alert = new Alert(Alert.AlertType.ERROR,  contentText: "you token has expired, login again!", ButtonType.OK),
    alert.showAndWait();
    MenuHandler.setToken("no token");
    MenuHandler.setUsername(null);
    MenuHandler.setUserType(null);
    MenuHandler.setIsUserLogin(false);
    Parent root = null;
    try {
        root = FXMLLoader.load(getClass().getResource( name: "/GUI/ProductScene/ProductScene.fxml"));
    } catch (Exception e) {
        e.printStackTrace();
    }
    MenuHandler.getStage().setScene(new Scene(root));
}
```

*** متود logout وقتی که توکن expire میشود توسط کلاینت صدا زده می شود.