Authentication and Authorization

Notes

- Authentication is the process of verifying who you are.
- Authorization is what you are able to do; authorization attacks have to do with accessing information that the user does not have permission to access.

Strong Password Policy :

Length: at least 10 characters .
Composition : At least one uppercase - lowercase - digit - Special characters (% $;) .
Do not include personal information and dictionary words.
Change password regularly (monthly, annually).
Never use the same password twice.

Server side polices :

Store passwords hashed with salts.
Adds an increasing delay after each failed login attempt
After 3 failed attempts show a CAPTCHA puzzle
After 10 failed attempts, it locks the user for a certain amount of time

Some behaviours to look at:

user doesn't exist :	**-->**	**user exists**
cookies deleted	-->	new cookie, cookie not deleted
goes to known fixed page	-->	goes to user specific page
html is fixed	-->	html changes , not like an invalid user

Timing attacks:

Rely on the time taken in a specific process , you can infer some stuff like:

- User does not exist in the DB: show error + abort
- User exists in the DB: retrieve user, calculate password, check if the password matches

Use Burp Comparer

a tool in Burp Suite that finds visual differences between two *responses*.

Check :

- default credentials
- test user accounts : accounts made to test the application.
- Try:

Usernames:	Password
administrator	<blank>
admin	password
root	pass123
guest	guest
system	adminpassword

test   1234

- On forms:
  INPUT TYPE="password" AUTOCOMPLETE="on"
  enables the browser to cache the password.

- unlimited attempts to answer a secret question.
- blocking the IP after several consecutive tries.
- **Guessable** password reset link
- **Predictable** password reset **token**
- **Recyclable** password reset link (can be used more than once)

- **Session Resurrection** ---> read about it

- **CAPTCHA** : *Completely Automated Public Turing test to tell Computers and Humans Apart*.
Tools to bypass CAPTCHA's
- Cintruder: https://cintruder.03c8.net/
- Bypass CAPTCHA with OCR engine: http://www.debasish.in/2012/01/bypass-captcha-using-python-and.html
- Decoding CAPTCHA: https://boyter.org/decoding-captchas/
- OWASP: Testing for CAPTCHA: https://boyter.org/decoding-captchas/

IDOR : (Insecure Direct Object reference.)

mitigation : always include a check for authorization in the begging of the webpage
Example:

```
<?
    session_start();
    if (!isset($_SESSION['islogged'])) {
        header("Location: http://www.mysite.com/login");
        die();
    }
?>
```

Improper redirect : sensitive info is sent and depends on the browser redirection that the client won't see

```
<?
session_start();
if (!isset($_SESSION['logged'])) {
header("Location: http://www.elsfoo.com/login");
die(); }
?>
```

Challenges

- **Improper redirect :** https://www.root-me.org/en/Challenges/Web-Server/HTTP-Improper-redirect

References and Resources

- Common wordlists:
  https://www.openwall.com/wordlists/
  https://github.com/danielmiessler/SecLists
  https://wiki.skullsecurity.org/Passwords