

Web Servers

1 Defining Web Interfaces

- cgi scripting

2 Password Encryption and Authentication

- secure hash algorithm
- an open source database managed with Python
- login to your mailbox

3 JupyterHub

- a server for Jupyter notebooks
- phcpy in a SageMath kernel of a Jupyter notebook

MCS 507 Lecture 21
Mathematical, Statistical and Scientific Software
Jan Verschelde, 9 October 2023

Web Servers

1 Defining Web Interfaces

- cgi scripting

2 Password Encryption and Authentication

- secure hash algorithm
- an open source database managed with Python
- login to your mailbox

3 JupyterHub

- a server for Jupyter notebooks
- phcpy in a SageMath kernel of a Jupyter notebook

CGI scripting

Python comes with batteries included.

- 1 Posting and processing of HTML forms:
 - ▶ pure Python code prints the HTML code; and
 - ▶ the input of the forms is processed with Python functions.
- 2 MySQLdb does the management of user data:
 - ▶ names and encrypted passwords,
 - ▶ generic, random folder names to store data files.
- 3 With the module `smtplib` we define email exchanges:
 - ▶ automatic 2-step registration process,
 - ▶ automatic password recovery protocol.

Test offline on `localhost` till confident to deploy.

Web Servers

1 Defining Web Interfaces

- cgi scripting

2 Password Encryption and Authentication

- **secure hash algorithm**
- an open source database managed with Python
- login to your mailbox

3 JupyterHub

- a server for Jupyter notebooks
- phcpy in a SageMath kernel of a Jupyter notebook

Secure Hash Algorithm

for password encryption

We can use cookies for login names and passwords.

If passwords are unencrypted, then insecure.

Secure hash algorithm:

- 1 computationally infeasible to compute inverse
→ is *trapdoor* function
- 2 very low collision rate
→ very low chance that two different messages will generate the same key

Server encrypts password before sending to client.

Authentication by comparing encrypted passwords.

using the hashlib module

```
>>> from hashlib import sha1
>>> m = 'this is me'
>>> h = sha1(m.encode())
>>> h
<sha1 HASH object @ 0x1019a53f0>
>>> h.hexdigest()
'99cf08cddcc3ae19fae7ec8f53f5179786b11406'
```

Web Servers

1 Defining Web Interfaces

- cgi scripting

2 Password Encryption and Authentication

- secure hash algorithm
- an open source database managed with Python
- login to your mailbox

3 JupyterHub

- a server for Jupyter notebooks
- phcpy in a SageMath kernel of a Jupyter notebook

MySQL, MySQLdb, and pymysql

MySQL is an open source database,
developed by the company MySQL AB.

In February 2008, Sun Microsystems acquired MySQL AB and the
expertise of the GPL software for \$1 billion.

In January 2010, Oracle acquired Sun for \$7.38 billion.

MySQL can be downloaded for free from
<http://www.mysql.com/downloads>.

Following the instructions, install MySQL first.

MySQLdb is an interface to connect Python to MySQL.

An alternative to MySQLdb is pymysql.

MySQL is another example of client/server computing:

- the MySQL client interacts with the application and the server;
- the MySQL server manages data on disk and serves the client.

Python comes with the database `sqlite3`.

managing a database with python

Two advantages of using python to manage the database:

- 1 Wrapping of sometimes complicates queries.
- 2 Automated data entry and retrieval.

Password are stored *only* in encrypted form.

Web Servers

1 Defining Web Interfaces

- cgi scripting

2 Password Encryption and Authentication

- secure hash algorithm
- an open source database managed with Python
- login to your mailbox

3 JupyterHub

- a server for Jupyter notebooks
- phcpy in a SageMath kernel of a Jupyter notebook

connecting to gmail

The Simple Mail Transfer Protocol (SMTP) is an internet standard for email transmission

In Python we can send email using `smtplib`.

To connect to the `gmail` email server:

```
>>> from smtplib import SMTP
>>> s = SMTP('smtp.gmail.com', 587)
>>> s.starttls()
(220, b'2.0.0 Ready to start TLS')
```

The Transport Layer Security (TLS) protocol provides data encryption for socket based communication.

To access your gmail, an app password needs to be generated and that password has to be used on the computer that runs python.

the script `emailauth.py`

```
from smtplib import SMTP
from getpass import getpass

try:
    SERVER = SMTP(host='smtp.gmail.com', port=587)
    RESULT = SERVER.starttls()
    print(RESULT)
except:
    print('Failed to connect to the server.')
```

the script `emailauth.py` continued

```
SUCCESS = False
USER = input('Type your account : ')
USER = USER + '@gmail.com'
for _ in range(3): # allow for three attempts
    PASW = getpass('Type your password : ')
    try:
        RESULT = SERVER.login(USER, PASW)
        print(RESULT)
        SUCCESS = True
        break
    except:
        print('Login failed, please try again.')
```

composing the message

MIME (Multipurpose Internet Mail Extensions) is an internet standard to extend the format of email. In Python, `email.mime.text` represents `text/*` type MIME documents.

```
>>> from email.mime.text import MIMEText
>>> msg = MIMEText('This is a test.')
>>> msg['Subject'] = 'hello'
>>> msg['To'] = 'someuser@gmail.com'
>>> print(msg.as_string())
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Subject: hello
To: someuser@gmail.com

This is a test.
>>>
```

register and activate an account

The registration/activation is a two step process:

- 1 The user connects to our web site and submits a form with the email address of the user (which will be the name of the account) and a password, which is encrypted.

The `register.py` script

- 1 stores the account information,
- 2 generates a random code for one time use,
- 3 sends the email to the user with link to the activate script that takes the random code as input.

The email sent contains an URL of the following form:

```
/cgi-bin/activate.py?login=%s&code=%s
```

This login and the code are the email address and the code.

- 2 The activation process is executed by `activate.py`.

the activation of an account

- 2 The activation process is executed by `activate.py`.

This script checks whether the two parameters in the `activate` script match what is stored in the database.

If okay, if the access code gives a match for the login name, then the user can login via the login form.

In this process, the server only sends email, does not read it.

Web Servers

1 Defining Web Interfaces

- cgi scripting

2 Password Encryption and Authentication

- secure hash algorithm
- an open source database managed with Python
- login to your mailbox

3 JupyterHub

- a server for Jupyter notebooks
- phcpy in a SageMath kernel of a Jupyter notebook

a server for Jupyter notebooks

<https://jupyter.org/hub>



A multi-user version of the notebook designed for companies, classrooms and research labs

JupyterHub allows to make notebooks available to your users.

Web Servers

1 Defining Web Interfaces

- cgi scripting


2 Password Encryption and Authentication

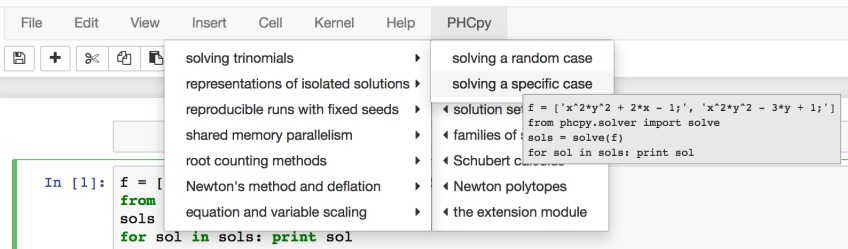
- secure hash algorithm
- an open source database managed with Python
- login to your mailbox

3 JupyterHub

- a server for Jupyter notebooks
- phcpy in a SageMath kernel of a Jupyter notebook

phcpy in a SageMath kernel of a Jupyter notebook

 jupyter code_snippet Last Checkpoint: 4 minutes ago (autosaved)



The screenshot shows a Jupyter notebook interface with the PHCpy menu open. The menu items are:

- solving trinomials
- representations of isolated solutions
- reproducible runs with fixed seeds
- shared memory parallelism
- root counting methods
- Newton's method and deflation
- equation and variable scaling
- solving a random case
- solving a specific case
- ◀ solution set
- ◀ families of
- ◀ Schubert classes
- ◀ Newton polytopes
- ◀ the extension module

The code cell contains the following code:

```
In [1]: f = [
from
sols
for sol in sols: print sol
```

```
PHCv2.4.64 released 2019-01-21 works!
total degree : 16
2-homogeneous Bezout number : 8
  with partition : { x }{ y }
general linear-product Bezout number : 8
  based on the set structure :
    { x }{ x }{ y }{ y }
    { x }{ x }{ y }{ y }
mixed volume : 4
stable mixed volume : 4
t : 1.000000000000000E+00  0.000000000000000E+00
m : 1
the solution for t :
  x : 4.86132470489966E-01  0.000000000000000E+00
  y : 3.42578353006690E-01 -2.28597478256455E-100
== err : 3.499E-17 = rco : 8.882E-01 = res : 4.857E-17 =
```

- Code snippets suggest typical applications, and guide the novice user.
- Solve polynomials by pointing and clicking.

Jupyter and JupyterHub

The Jupyter notebook supports language agnostic computations, supporting execution environments in several dozen languages.

With JupyterHub, we can run the code in a Python Terminal session, in a Jupyter notebook running Python, or in a SageMath session.

For the user administration, we recycled our first web interface.

With JupyterHub, we provide user accounts on our server.

- At login time, a new process is spawned.
- Users have generic, random login names.
- Actions of users must be isolated from each other.

The setup requires some system administration expertise.

Summary + Exercises

Exercises:

- 1 Demonstrate how you can access your mailbox from within a Python script.
- 2 Write code for the `register.py` described earlier.
- 3 Write code for the `activate.py` described earlier.