# Vulnerable Software

So far we have seen request smuggling vulnerabilities that arise from improper parsing or a lack of support for the TE header. However, web servers or reverse proxies can also be vulnerable to request smuggling due to other bugs that cause the length of a request to be parsed incorrectly.

## Identification

In our lab, we will exploit a vulnerability in the Python [Gunicorn](https://gunicorn.org/) web server that was detailed in [this](https://grenfeldt.dev/2021/04/01/gunicorn-20.0.4-request-smuggling/) blog post. Gunicorn 20.0.4 contained a bug when encountering the HTTP header Sec-Websocket-Key1 that fixed the request body to a length of 8 bytes, no matter what value the CL and TE headers are set to. This is a special header used in the establishment of WebSocket connections. Since the reverse proxy does not suffer from this bug, this allows us to create desynchronization between the two systems.

To confirm the vulnerability, let's create a tab group in Burp Repeater with the following two requests:

Code: http

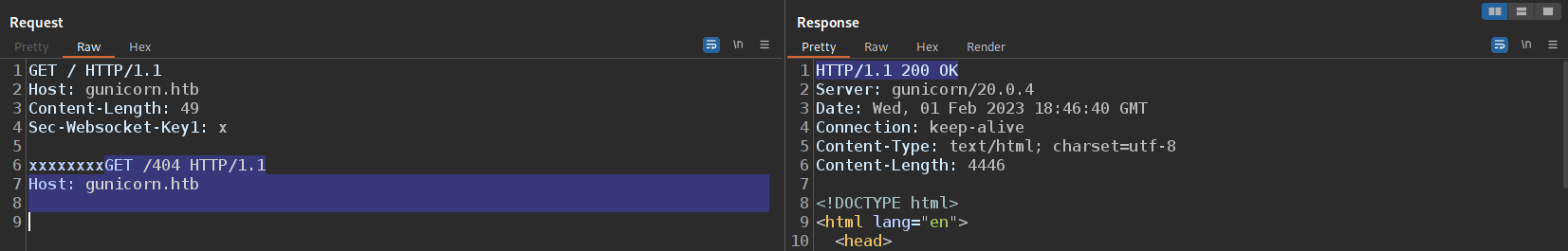
GET / HTTP/1.1  
Host: gunicorn.htb  
Content-Length: 49  
Sec-Websocket-Key1: x  
  
xxxxxxxxGET /404 HTTP/1.1  
Host: gunicorn.htb

and

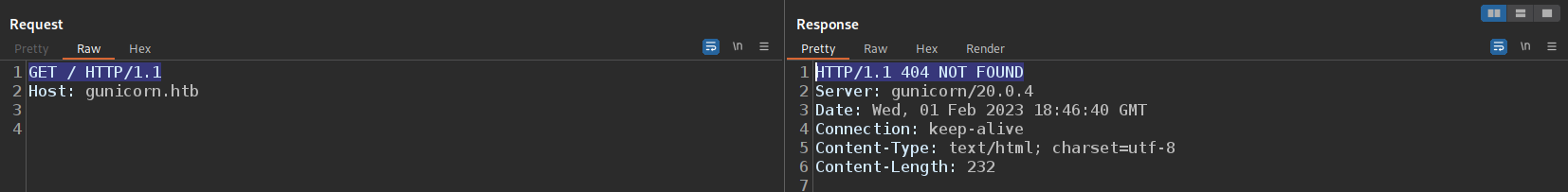
Code: http

GET / HTTP/1.1  
Host: gunicorn.htb

When we send the tab group via a single TCP connection, we can observe the following behavior. The first response contains the index of the website as we would expect:



However, while the second request also contains the path /, the response is a 404 status code:



To understand what happened here, we can again look at the TCP stream. Just like in the previous sections, we will look at it from the reverse proxy first:

GET / HTTP/1.1  
Host: gunicorn.htb  
Content-Length: 49  
Sec-Websocket-Key1: x  
  
xxxxxxxxGET /404 HTTP/1.1  
Host: gunicorn.htb  
  
GET / HTTP/1.1  
Host: gunicorn.htb

The reverse proxy parses the requests exactly how we would expect it. The first request is a GET request to / that has a body length of 49 bytes according to the CL header. After that, there is a second GET request to / with an empty request body. Now let's look at the TCP stream from the Gunicorn server's perspective:

GET / HTTP/1.1  
Host: gunicorn.htb  
Content-Length: 49  
Sec-Websocket-Key1: x  
  
xxxxxxxxGET /404 HTTP/1.1  
Host: gunicorn.htb  
  
GET / HTTP/1.1  
Host: gunicorn.htb

Due to the bug in Gunicorn, the web server assumes a body length of 8 bytes as soon as it parses the Sec-Websocket-Key1 HTTP header even though the CL header specifies a different body length. Therefore, the first request's body contains only xxxxxxxx. Afterward, the web server sees a second GET request to /404 to which it responds with the 404 response we can observe in Burp. Lastly, the web server parses our second GET request to / as a third request. We could also hide this request in the body of the request to /404 by specifying a CL header here. But that is unnecessary in this scenario.

## Exploitation

Just like in the previous section, our goal is to access the admin panel by bypassing the WAF that rejects all requests containing admin in the query string. We can do this by hiding the request to the admin panel from the WAF in our first request such that the WAF never parses it as an HTTP request's query string. The exploitation is thus similar to the exploit we showcased in the last section.