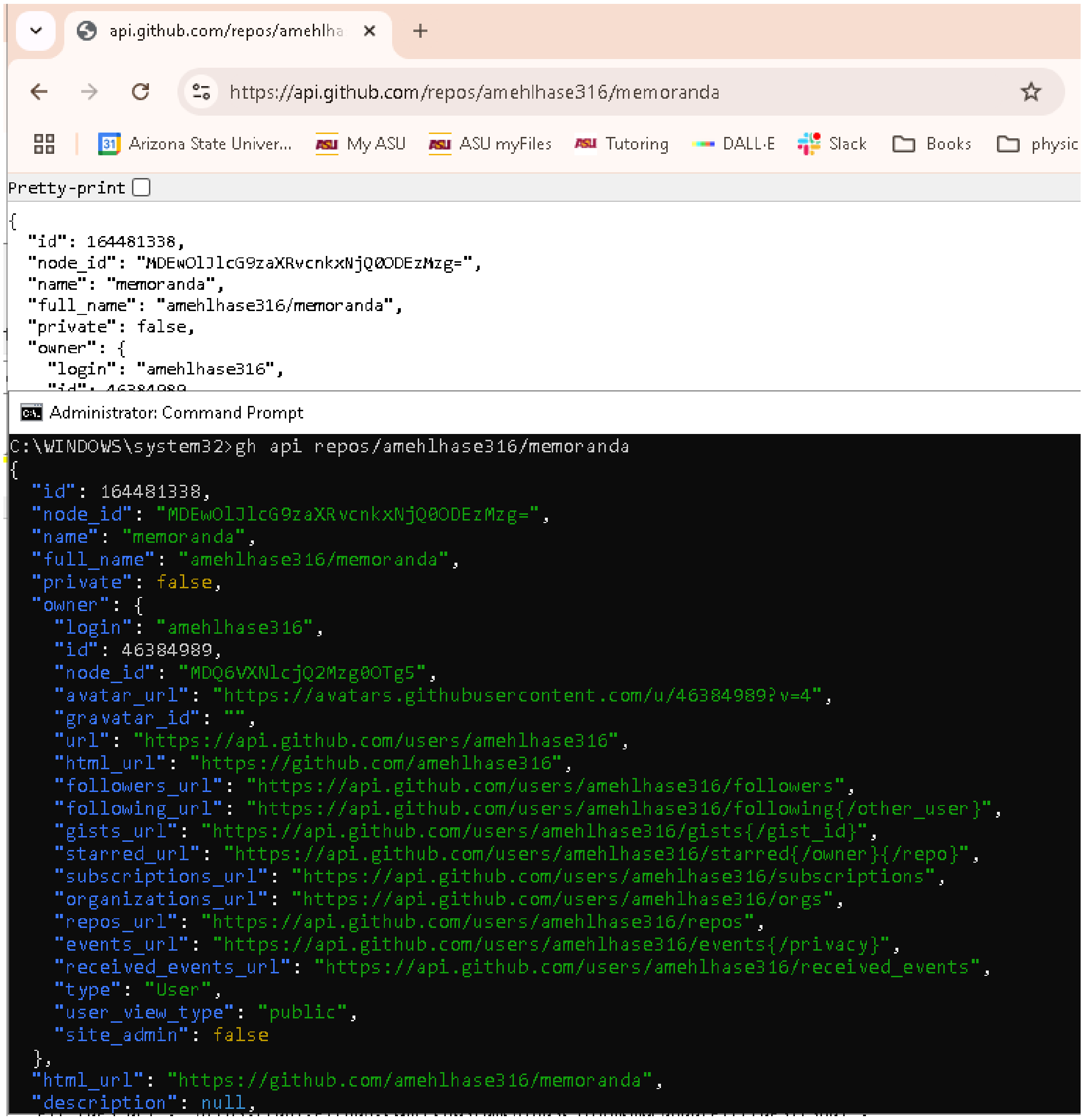
Nathaniel Davis-Perez
3/31/25
SER 321

Assignment 2

https://github.com/notnatedavis/ser321-spring25B-ndavispe

1. URL used : (https://api.github.com/repos/amehlhase316/memoranda)


api.github.com/repos/amehlha
https://api.github.com/repos/amehlhase316/memoranda
Arizona State Univer...
My ASU
ASU myFiles
Tutoring
DALL·E
Slack
Books
physic
Pretty-print □
Administrator: Command Prompt

```
{
  "id": 164481338,
  "node_id": "MDEwOlJlcG9zaXRvcnkxNjQ0ODEzMzg=",
  "name": "memoranda",
  "full_name": "amehlhase316/memoranda",
  "private": false,
  "owner": {
    "login": "amehlhase316",
```

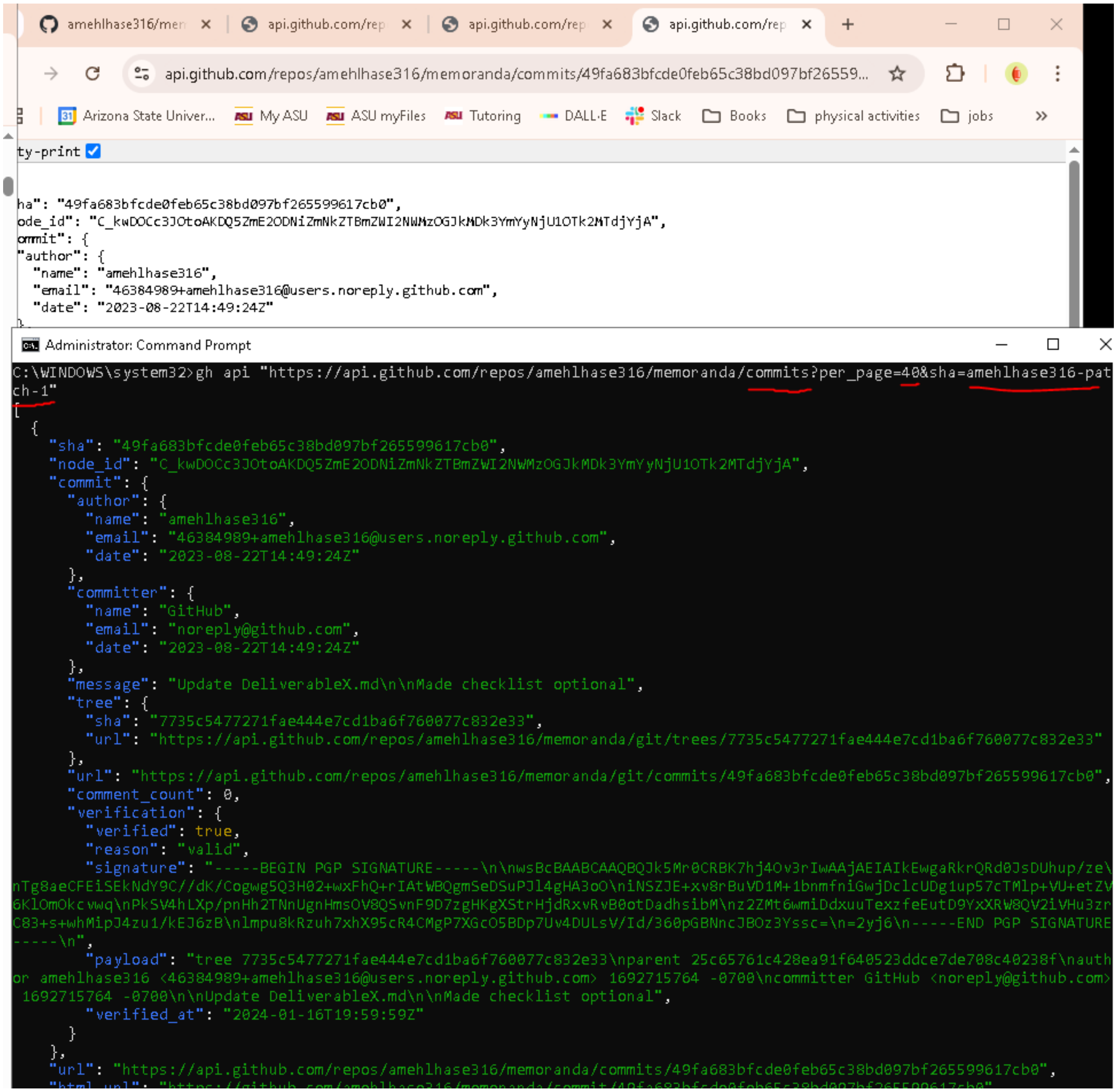
```
C:\WINDOWS\system32>gh api repos/amehlhase316/memoranda
{
  "id": 164481338,
  "node_id": "MDEwOlJlcG9zaXRvcnkxNjQ0ODEzMzg=",
  "name": "memoranda",
  "full_name": "amehlhase316/memoranda",
  "private": false,
  "owner": {
    "login": "amehlhase316",
    "id": 46384989,
    "node_id": "MDQ6VXNlcjQ2Mzg0OTg5",
    "avatar_url": "https://avatars.githubusercontent.com/u/46384989?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/amehlhase316",
    "html_url": "https://github.com/amehlhase316",
    "followers_url": "https://api.github.com/users/amehlhase316/followers",
    "following_url": "https://api.github.com/users/amehlhase316/following{/other_user}",
    "gists_url": "https://api.github.com/users/amehlhase316/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/amehlhase316/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/amehlhase316/subscriptions",
    "organizations_url": "https://api.github.com/users/amehlhase316/orgs",
    "repos_url": "https://api.github.com/users/amehlhase316/repos",
    "events_url": "https://api.github.com/users/amehlhase316/events{/privacy}",
    "received_events_url": "https://api.github.com/users/amehlhase316/received_events",
    "type": "User",
    "user_view_type": "public",
    "site_admin": false
  },
  "html_url": "https://github.com/amehlhase316/memoranda",
  "description": null,
```

URLs used : (https://api.github.com/repos/amehlhase316/memoranda/branches)

URLs used : (https://api.github.com/repos/amehlhase316/memoranda/commits/49fa683bfcde0feb65c38bd097bf265599617cb0)

Nathaniel Davis-Perez
3/31/25
SER 321

Final URL used :
(https://api.github.com/repos/amehlhase316/memoranda/commits?per_page=40&sha=amehlhase316-patch-1)



```
ha": "49fa683bfcde0feb65c38bd097bf265599617cb0",
ode_id": "C_kwDOCc3JOtoAKDQ5ZmE2ODNiZmNkZTBmZWI2NWMzOGJkMDk3YmYyNjU1OTk2MTdjYjA",
ommit": {
"author": {
  "name": "amehlhase316",
  "email": "46384989+amehlhase316@users.noreply.github.com",
  "date": "2023-08-22T14:49:24Z"
```

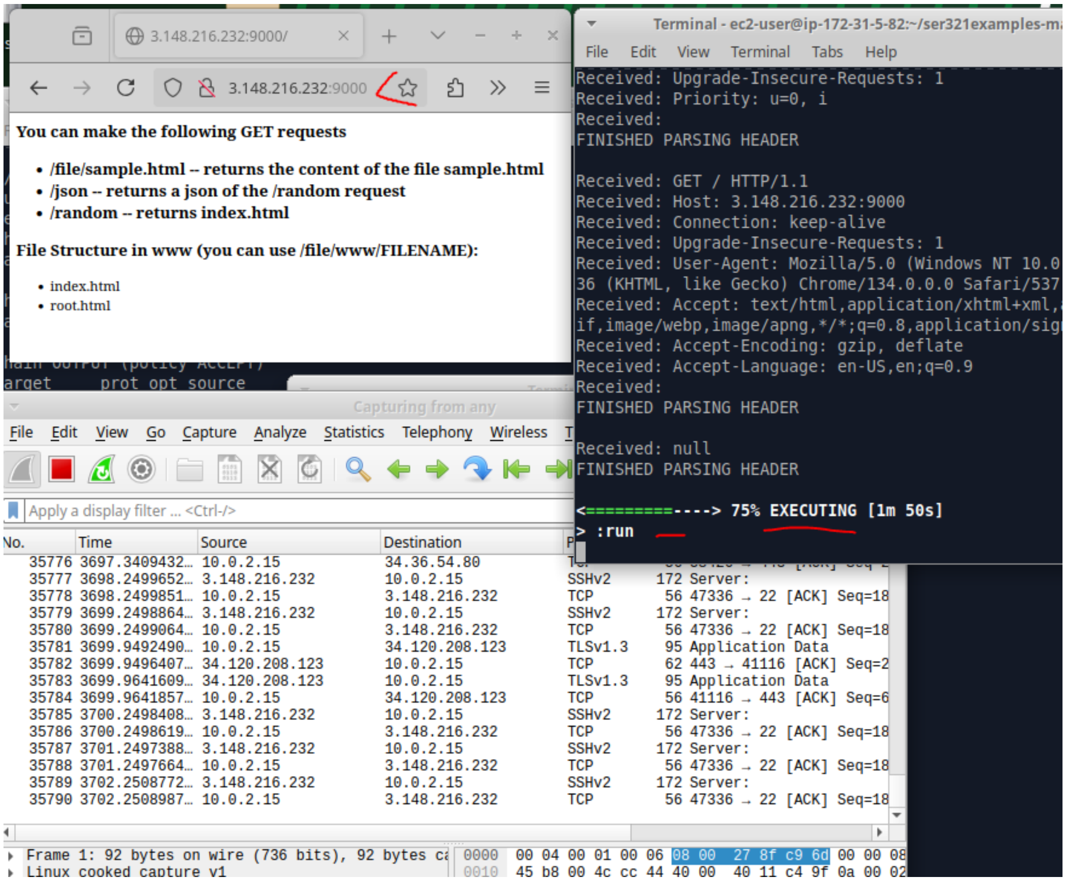
```
C:\WINDOWS\system32>gh api "https://api.github.com/repos/amehlhase316/memoranda/commits?per_page=40&sha=amehlhase316-pat
ch-1"
[
  {
    "sha": "49fa683bfcde0feb65c38bd097bf265599617cb0",
    "node_id": "C_kwDOCc3JOtoAKDQ5ZmE2ODNiZmNkZTBmZWI2NWMzOGJkMDk3YmYyNjU1OTk2MTdjYjA",
    "commit": {
      "author": {
        "name": "amehlhase316",
        "email": "46384989+amehlhase316@users.noreply.github.com",
        "date": "2023-08-22T14:49:24Z"
      },
      "committer": {
        "name": "GitHub",
        "email": "noreply@github.com",
        "date": "2023-08-22T14:49:24Z"
      },
      "message": "Update DeliverableX.md\n\nMade checklist optional",
      "tree": {
        "sha": "7735c5477271fae444e7cd1ba6f760077c832e33",
        "url": "https://api.github.com/repos/amehlhase316/memoranda/git/trees/7735c5477271fae444e7cd1ba6f760077c832e33"
      },
      "url": "https://api.github.com/repos/amehlhase316/memoranda/git/commits/49fa683bfcde0feb65c38bd097bf265599617cb0",
      "comment_count": 0,
      "verification": {
        "verified": true,
        "reason": "valid",
        "signature": "-----BEGIN PGP SIGNATURE-----\n\nwsBcBAABCAAQBQJk5Mr0CRBK7hj4Ov3rIwAAjAEIAIkEwgaRkrQRd0JsDUhup/ze\
nTg8aeCFEiSEkNdY9C//dK/Cogwg5Q3H02+wxFhQ+rIAtWBQgmSeDSuPJl4gHA3oO\niNSZJE+xv8rBuVD1M+1bnmfniGwjDclcUDg1up57cTMlp+VU+etZV
6KlOmOkcvwq\nPkSV4hLXp/pnHh2TNnUgnHmsOV8QSvnF9D7zgHKgXStrHjdRxvRvB0otDadhsibM\nz2ZMt6wmiDdxuuTexzfeEutD9YxXRW8QV2iVHu3zr
C83+s+whMipJ4zu1/kEJ6zB\nlmpu8kRzuh7xhX95cR4CMgP7XGcO5BDp7Uv4DULsV/Id/360pGBNncJBOz3Yssc=\n=2yj6\n-----END PGP SIGNATURE
-----\n",
        "payload": "tree 7735c5477271fae444e7cd1ba6f760077c832e33\nparent 25c65761c428ea91f640523ddce7de708c40238f\nauth
or amehlhase316 <46384989+amehlhase316@users.noreply.github.com> 1692715764 -0700\ncommitter GitHub <noreply@github.com>
 1692715764 -0700\n\nUpdate DeliverableX.md\n\nMade checklist optional",
        "verified_at": "2024-01-16T19:59:59Z"
      }
    },
    "url": "https://api.github.com/repos/amehlhase316/memoranda/commits/49fa683bfcde0feb65c38bd097bf265599617cb0",
    "html_url": "https://github.com/amehlhase316/memoranda/commit/49fa683bfcde0feb65c38bd097bf265599617cb0"
```

1.1 : The specific API calls I used were GitHub API endpoint (/repos/amehlhase316/memoranda/commits), and query params; per_page (per_page=40) along with sha for branch specification (sha=amehlhase316-patch-1), referencing from [https://gh.r-lib.org/reference/gh.html]

Nathaniel Davis-Perez
3/31/25
SER 321

1.2 : Stateless Communication is where all/each request to the server is independent and contains all the information needed to process like in this current case for accessing api.github and does not keep information between requests. Stateful Communication on is when the server keeps/maintains information between requests like an example being user authentication to github such that next instance that information is retained and used again.

2.2 :


3.148.216.232:9000/
3.148.216.232:9000
Terminal - ec2-user@ip-172-31-5-82:~/ser321examples-ma
File Edit View Terminal Tabs Help
Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless T
Apply a display filter ... <Ctrl-/>

You can make the following GET requests

- **/file/sample.html -- returns the content of the file sample.html**
- **/json -- returns a json of the /random request**
- **/random -- returns index.html**

File Structure in www (you can use /file/www/FILENAME):

- index.html
- root.html

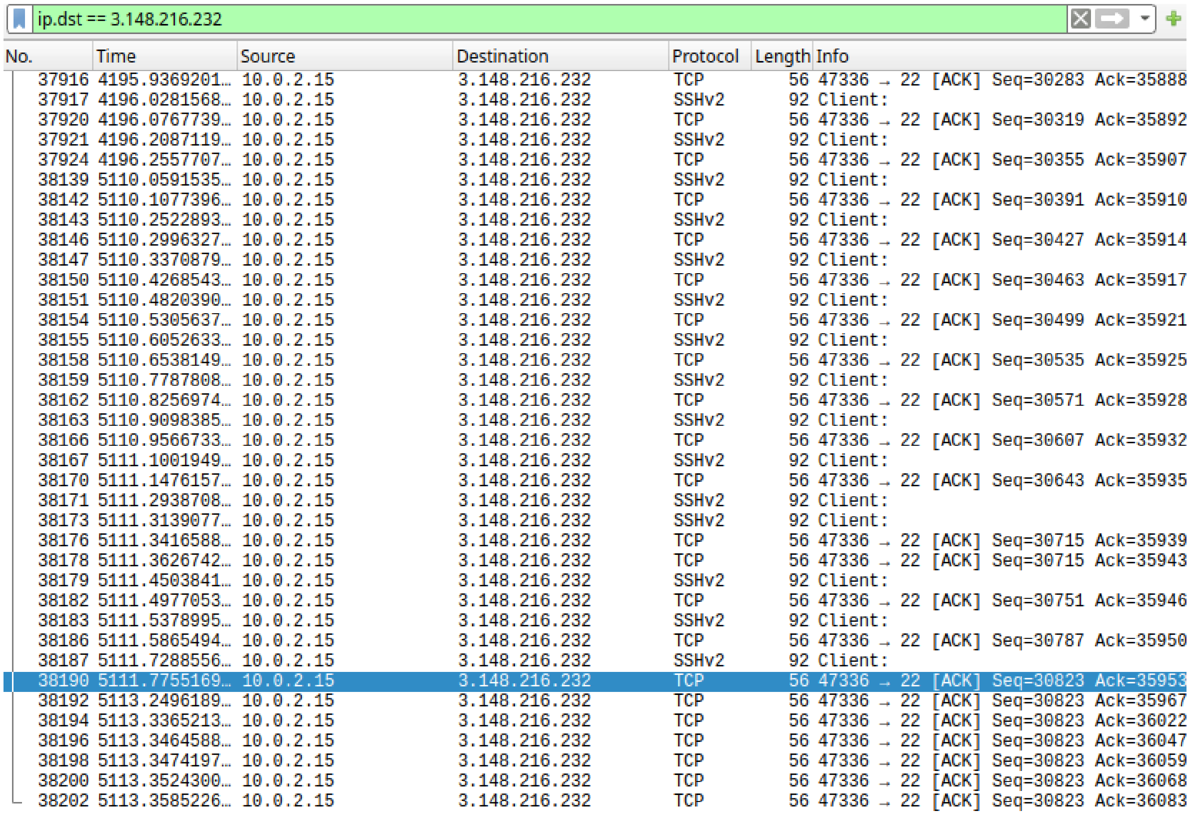
```
hain OUTPUT (policy ACCEPT)
arget     prot opt source
```

```
Received: Upgrade-Insecure-Requests: 1
Received: Priority: u=0, i
Received:
FINISHED PARSING HEADER

Received: GET / HTTP/1.1
Received: Host: 3.148.216.232:9000
Received: Connection: keep-alive
Received: Upgrade-Insecure-Requests: 1
Received: User-Agent: Mozilla/5.0 (Windows NT 10.0
36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537
Received: Accept: text/html,application/xhtml+xml,
if,image/webp,image/apng,*/*;q=0.8,application/sig
Received: Accept-Encoding: gzip, deflate
Received: Accept-Language: en-US,en;q=0.9
Received:
FINISHED PARSING HEADER

Received: null
FINISHED PARSING HEADER

<=========----> 75% EXECUTING [1m 50s]
> :run
```

```
No.   Time          Source          Destination     P
35776 3697.3409432… 10.0.2.15       34.36.54.80
35777 3698.2499652… 3.148.216.232   10.0.2.15       SSHv2    172 Server:
35778 3698.2499851… 10.0.2.15       3.148.216.232   TCP       56 47336 → 22 [ACK] Seq=18
35779 3699.2498864… 3.148.216.232   10.0.2.15       SSHv2    172 Server:
35780 3699.2499064… 10.0.2.15       3.148.216.232   TCP       56 47336 → 22 [ACK] Seq=18
35781 3699.9492490… 10.0.2.15       34.120.208.123  TLSv1.3   95 Application Data
35782 3699.9496407… 34.120.208.123  10.0.2.15       TCP       62 443 → 41116 [ACK] Seq=2
35783 3699.9641609… 34.120.208.123  10.0.2.15       TLSv1.3   95 Application Data
35784 3699.9641857… 10.0.2.15       34.120.208.123  TCP       56 41116 → 443 [ACK] Seq=6
35785 3700.2498408… 3.148.216.232   10.0.2.15       SSHv2    172 Server:
35786 3700.2498619… 10.0.2.15       3.148.216.232   TCP       56 47336 → 22 [ACK] Seq=18
35787 3701.2497388… 3.148.216.232   10.0.2.15       SSHv2    172 Server:
35788 3701.2497664… 10.0.2.15       3.148.216.232   TCP       56 47336 → 22 [ACK] Seq=18
35789 3702.2508772… 3.148.216.232   10.0.2.15       SSHv2    172 Server:
35790 3702.2508987… 10.0.2.15       3.148.216.232   TCP       56 47336 → 22 [ACK] Seq=18

▸ Frame 1: 92 bytes on wire (736 bits), 92 bytes ca   0000  00 04 00 01 00 06 08 00  27 8f c9 6d 00 00 08
▸ Linux cooked capture v1                             0010  45 b8 00 4c cc 44 40 00  40 11 c4 9f 0a 00 02
```

Nathaniel Davis-Perez
3/31/25
SER 321

2.3



```
ip.dst == 3.148.216.232

No.   Time            Source     Destination    Protocol Length Info
37916 4195.9369201… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30283 Ack=35888
37917 4196.0281568… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
37920 4196.0767739… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30319 Ack=35892
37921 4196.2087119… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
37924 4196.2557707… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30355 Ack=35907
38139 5110.0591535… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
38142 5110.1077396… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30391 Ack=35910
38143 5110.2522893… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
38146 5110.2996327… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30427 Ack=35914
38147 5110.3370879… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
38150 5110.4268543… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30463 Ack=35917
38151 5110.4820390… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
38154 5110.5305637… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30499 Ack=35921
38155 5110.6052633… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
38158 5110.6538149… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30535 Ack=35925
38159 5110.7787808… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
38162 5110.8256974… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30571 Ack=35928
38163 5110.9098385… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
38166 5110.9566733… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30607 Ack=35932
38167 5111.1001949… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
38170 5111.1476157… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30643 Ack=35935
38171 5111.2938708… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
38173 5111.3139077… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
38176 5111.3416588… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30715 Ack=35939
38178 5111.3626742… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30715 Ack=35943
38179 5111.4503841… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
38182 5111.4977053… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30751 Ack=35946
38183 5111.5378995… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
38186 5111.5865494… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30787 Ack=35950
38187 5111.7288556… 10.0.2.15  3.148.216.232  SSHv2    92 Client:
38190 5111.7755169… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30823 Ack=35953
38192 5113.2496189… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30823 Ack=35967
38194 5113.3365213… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30823 Ack=36022
38196 5113.3464588… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30823 Ack=36047
38198 5113.3474197… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30823 Ack=36059
38200 5113.3524300… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30823 Ack=36068
38202 5113.3585226… 10.0.2.15  3.148.216.232  TCP      56 47336 → 22 [ACK] Seq=30823 Ack=36083
```

2.3.1 : I used the filter ‘ip.dst==3.148.216.232’ that filters all traffic going to the ip destination of the server since being run locally on second machine.

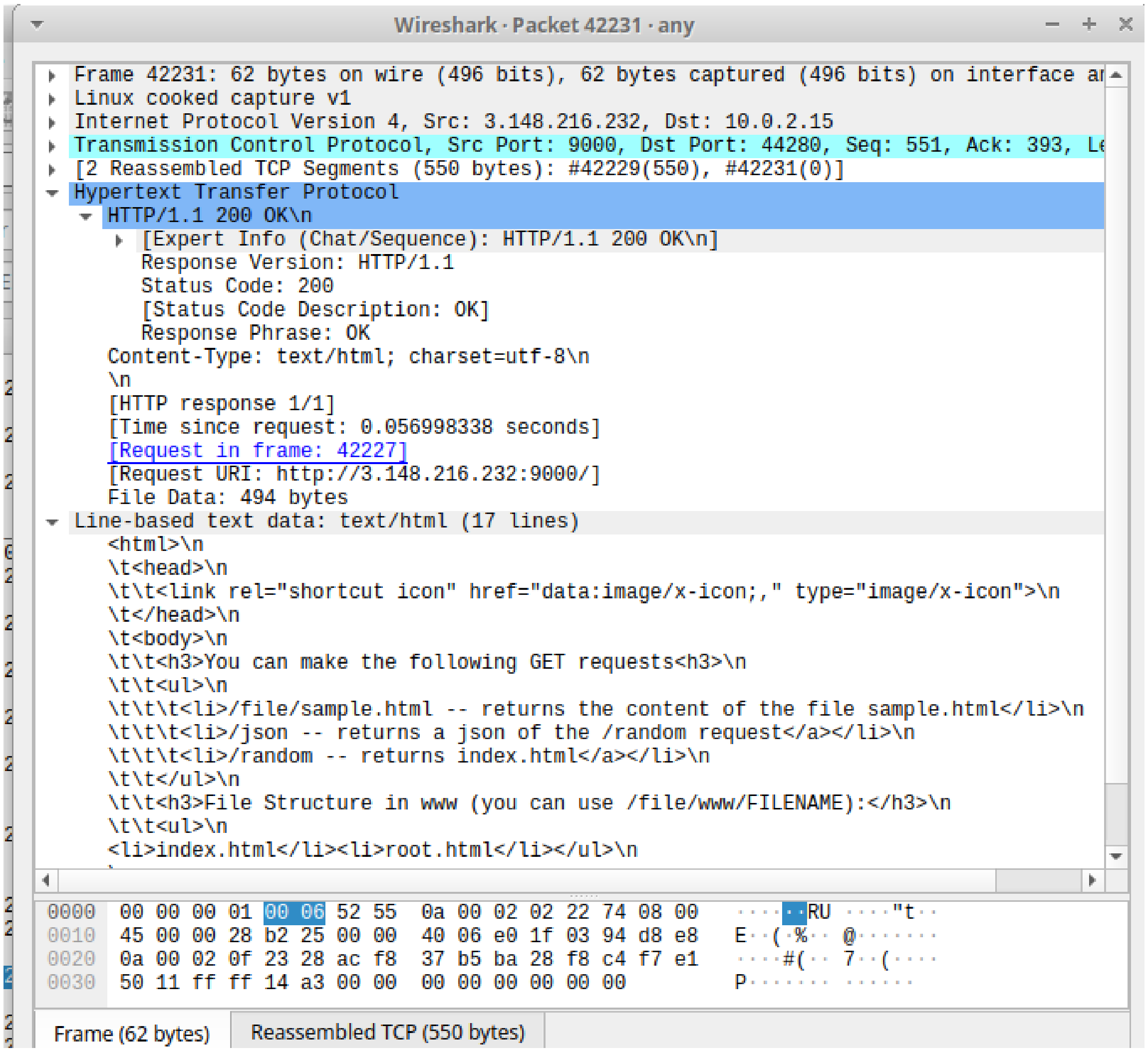
2.3.2 : When on the /random page and clicking the Random button, the page generates either ‘Streets’ with the associated image or ‘Bread’ with the associated image. In comparison refreshing the browser also automatically generates a random choice between Bread and Streets.

2.3.3 : Through different requests the server generates multiple response codes (200, 400, 404).

2.3.4 : 200 when a valid request is processed i.e. /random. 400 when a request is unrecognized or improper i.e. no GET in request header. 404 when a request doesn’t exist i.e. GET /random/bird. The codes 200, 400, and 404 are all defined within funHttpServer/WebServer.java.

2.3.5 : Yes, when loading initially the HTTP protocol is present within wireshark

Nathaniel Davis-Perez
3/31/25
SER 321



```
Frame 42231: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface ar
Linux cooked capture v1
Internet Protocol Version 4, Src: 3.148.216.232, Dst: 10.0.2.15
Transmission Control Protocol, Src Port: 9000, Dst Port: 44280, Seq: 551, Ack: 393, Le
[2 Reassembled TCP Segments (550 bytes): #42229(550), #42231(0)]
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\n]
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
  Content-Type: text/html; charset=utf-8\n
  \n
  [HTTP response 1/1]
  [Time since request: 0.056998338 seconds]
  [Request in frame: 42227]
  [Request URI: http://3.148.216.232:9000/]
  File Data: 494 bytes
Line-based text data: text/html (17 lines)
  <html>\n
  \t<head>\n
  \t\t<link rel="shortcut icon" href="data:image/x-icon;," type="image/x-icon">\n
  \t</head>\n
  \t<body>\n
  \t\t<h3>You can make the following GET requests<h3>\n
  \t\t<ul>\n
  \t\t\t<li>/file/sample.html -- returns the content of the file sample.html</li>\n
  \t\t\t<li>/json -- returns a json of the /random request</a></li>\n
  \t\t\t<li>/random -- returns index.html</a></li>\n
  \t\t</ul>\n
  \t\t<h3>File Structure in www (you can use /file/www/FILENAME):</h3>\n
  \t\t<ul>\n
  <li>index.html</li><li>root.html</li></ul>\n

0000  00 00 00 01 00 06 52 55  0a 00 02 02 22 74 08 00   ······RU ····"t··
0010  45 00 00 28 b2 25 00 00  40 06 e0 1f 03 94 d8 e8   E··(·%·· @·······
0020  0a 00 02 0f 23 28 ac f8  37 b5 ba 28 f8 c4 f7 e1   ····#(·· 7··(····
0030  50 11 ff ff 14 a3 00 00  00 00 00 00 00 00         P······· ······
```

2.3.6 : HTTPS is now more common than HTTP due in part to the fact that there are less vulnerabilities that are inherent to HTTP such as having the ability to peep at data through WireShark or even the manipulation of data.

2.3.7 : In this case the port the server listen's to for HTTP requests is 9000 which is manually hardcoded into WebServer.java, but the most common port for HTTP is 80.

2.3.8 : For this specific example when running the server, my local port used when sending different requests is (41133) from (79.46.23.207:41133)

2.4.1 : Now able to use the IP without specifying a port instead of :9000 since nginx listens on port 80.

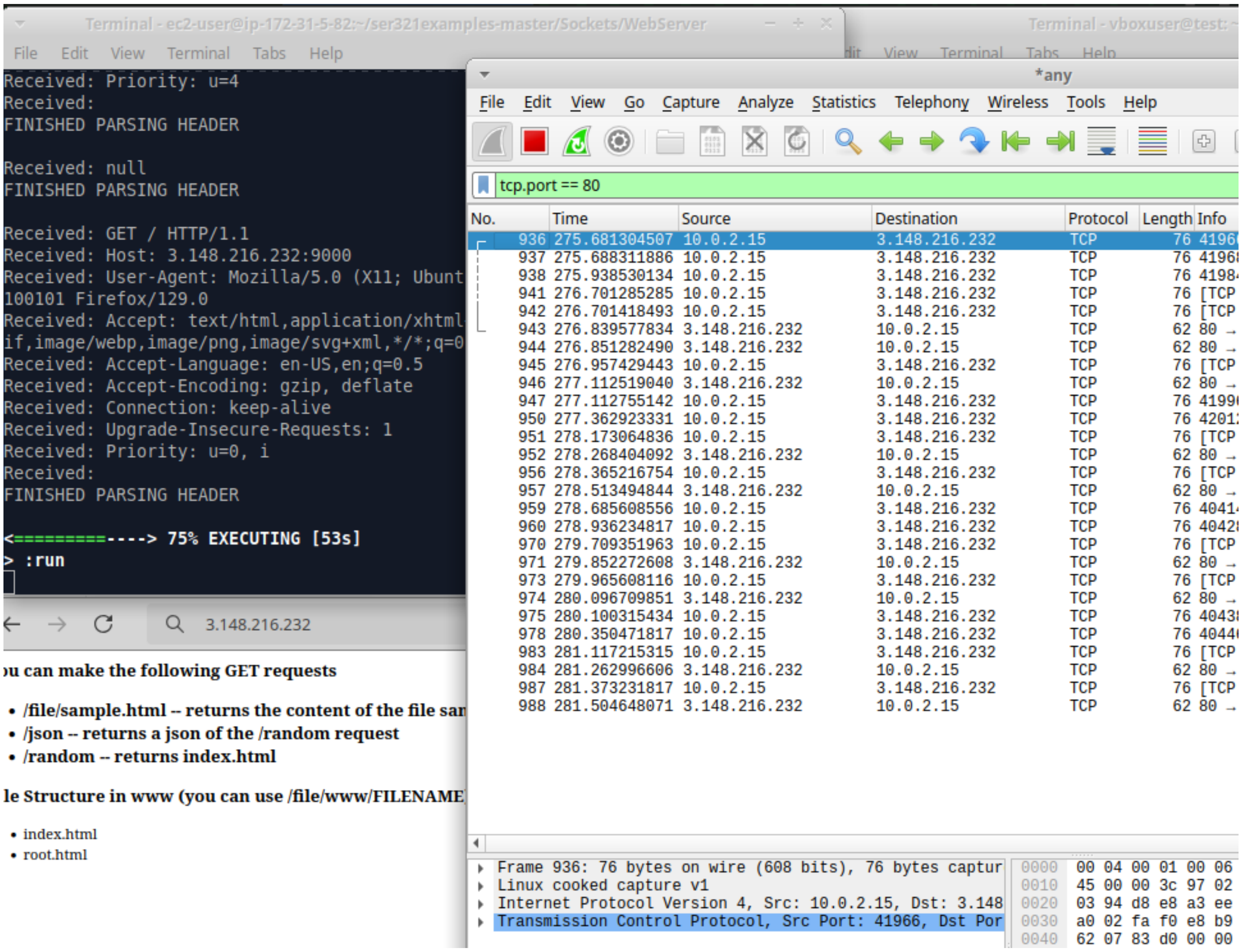
2.4.2 : The port that the traffic is now going to is 80 which is manually set in nginx.conf. It isn't the same as before and should be different, the internal port 9000 still remains unchanged

Nathaniel Davis-Perez
3/31/25
SER 321

2.4.3 : HTTP is still being used since nginx listens on port 80.

2.4.4 : Yes AWS security settings could be changed.

2.4.5 :


Terminal - ec2-user@ip-172-31-5-82:~/ser321examples-master/Sockets/WebServer
File Edit View Terminal Tabs Help
Terminal - vboxuser@test: ~
dit View Terminal Tabs Help
*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
Received: Priority: u=4
Received:
FINISHED PARSING HEADER

Received: null
FINISHED PARSING HEADER

Received: GET / HTTP/1.1
Received: Host: 3.148.216.232:9000
Received: User-Agent: Mozilla/5.0 (X11; Ubunt
100101 Firefox/129.0
Received: Accept: text/html,application/xhtml
if,image/webp,image/png,image/svg+xml,*/*;q=0
Received: Accept-Language: en-US,en;q=0.5
Received: Accept-Encoding: gzip, deflate
Received: Connection: keep-alive
Received: Upgrade-Insecure-Requests: 1
Received: Priority: u=0, i
Received:
FINISHED PARSING HEADER

<=========----> 75% EXECUTING [53s]
> :run
```

3.148.216.232

u can make the following GET requests

- /file/sample.html -- returns the content of the file sar
- /json -- returns a json of the /random request
- /random -- returns index.html

le Structure in www (you can use /file/www/FILENAME

- index.html
- root.html

tcp.port == 80

```
No.  Time          Source          Destination     Protocol Length Info
936  275.681304507 10.0.2.15       3.148.216.232   TCP      76 4196
937  275.688311886 10.0.2.15       3.148.216.232   TCP      76 4196
938  275.938530134 10.0.2.15       3.148.216.232   TCP      76 4198
941  276.701285285 10.0.2.15       3.148.216.232   TCP      76 [TCP
942  276.701418493 10.0.2.15       3.148.216.232   TCP      76 [TCP
943  276.839577834 3.148.216.232   10.0.2.15       TCP      62 80 →
944  276.851282490 3.148.216.232   10.0.2.15       TCP      62 80 →
945  276.957429443 10.0.2.15       3.148.216.232   TCP      76 [TCP
946  277.112519040 3.148.216.232   10.0.2.15       TCP      62 80 →
947  277.112755142 10.0.2.15       3.148.216.232   TCP      76 4199
950  277.362923331 10.0.2.15       3.148.216.232   TCP      76 4201
951  278.173064836 10.0.2.15       3.148.216.232   TCP      76 [TCP
952  278.268404092 3.148.216.232   10.0.2.15       TCP      62 80 →
956  278.365216754 10.0.2.15       3.148.216.232   TCP      76 [TCP
957  278.513494844 3.148.216.232   10.0.2.15       TCP      62 80 →
959  278.685608556 10.0.2.15       3.148.216.232   TCP      76 4041
960  278.936234817 10.0.2.15       3.148.216.232   TCP      76 4042
970  279.709351963 10.0.2.15       3.148.216.232   TCP      76 [TCP
971  279.852272608 3.148.216.232   10.0.2.15       TCP      62 80 →
973  279.965608116 10.0.2.15       3.148.216.232   TCP      76 [TCP
974  280.096709851 3.148.216.232   10.0.2.15       TCP      62 80 →
975  280.100315434 10.0.2.15       3.148.216.232   TCP      76 4043
978  280.350471817 10.0.2.15       3.148.216.232   TCP      76 4044
983  281.117215315 10.0.2.15       3.148.216.232   TCP      76 [TCP
984  281.262996606 3.148.216.232   10.0.2.15       TCP      62 80 →
987  281.373231817 10.0.2.15       3.148.216.232   TCP      76 [TCP
988  281.504648071 3.148.216.232   10.0.2.15       TCP      62 80 →
```

```
Frame 936: 76 bytes on wire (608 bits), 76 bytes captur    0000  00 04 00 01 00 06
Linux cooked capture v1                                    0010  45 00 00 3c 97 02
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 3.148    0020  03 94 d8 e8 a3 ee
Transmission Control Protocol, Src Port: 41966, Dst Por    0030  a0 02 fa f0 e8 b9
                                                           0040  62 07 83 d0 00 00
```

2.5 Extra Credit

2.6.1 : Updated ‘multiply?’ to check for initial existence of num1 & num2 (error code 400), attempt to perform math function on parsed nums (success response 200) else (error code 400) in the case of an exception.

2.6.3 : BMI & Concatenation

Multiply : usage (/multiply?num1=X&num2=Y)

Ex. [curl "http://3.148.216.232:9000/multiply?num1=5&num2=3"]

GitHub : usage (/github?query=API_PATH)

Ex. [curl "http://3.148.216.232:9000/github?query=users/amehlhase316/repos"]

Nathaniel Davis-Perez
3/31/25
SER 321

BMI Calc : usage (/bmi?weight=LB&height=IN)

Ex. [curl "http://3.148.216.232:9000/bmi?weight=150&height=65"]

String Concatenation : usage (/concat?str1=TEXT&str2=TEXT)

Ex. [curl "http://3.148.216.232:9000/concat?str1=Hello&str2=World"]