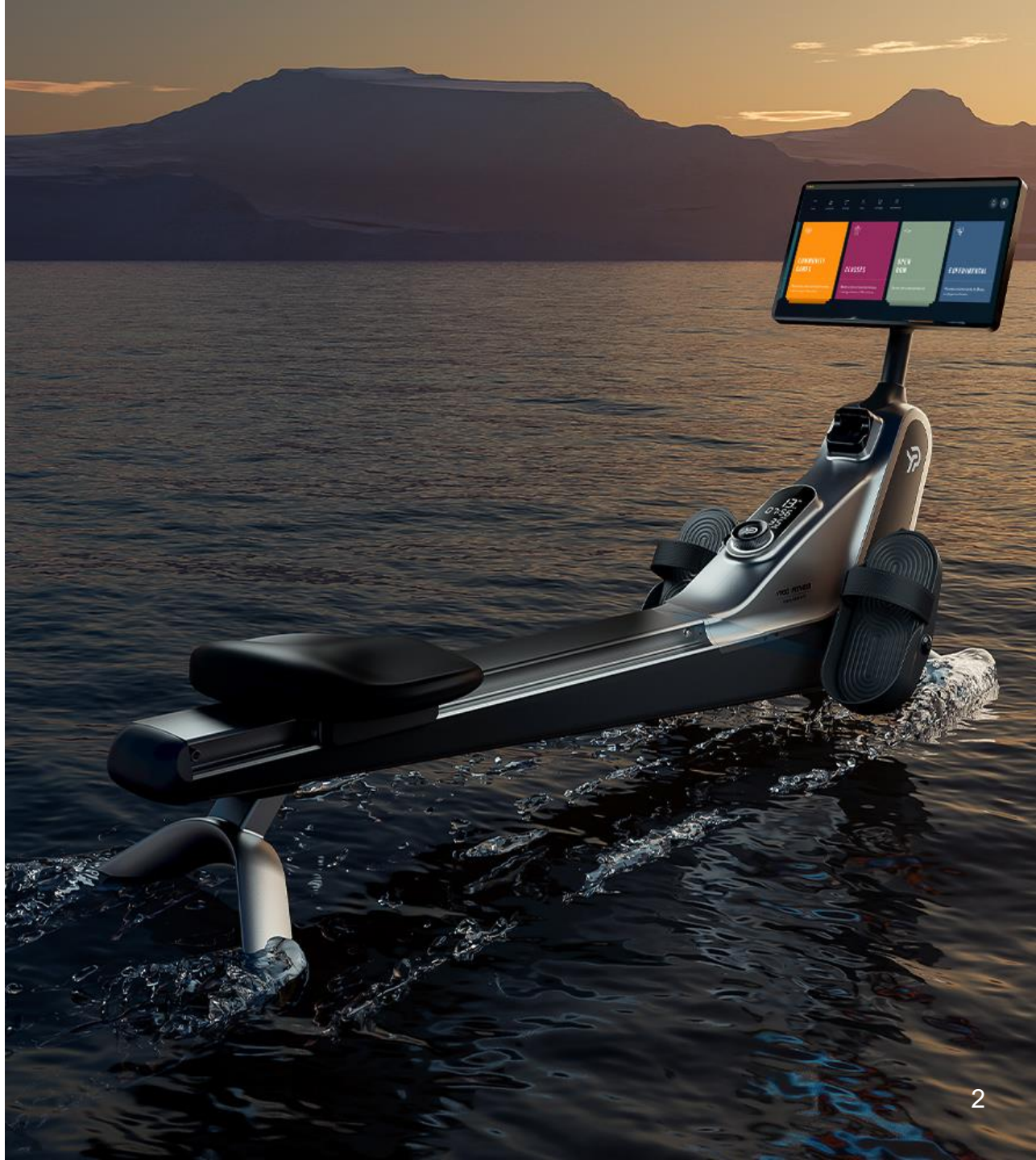
Rudder Nonsense:
Steering Smart Rowers Off Course

Presented by Shane Kell, Senior Security Consultant

10.25.24

The Rower

- Smart rowing machine
- Android-powered user interface
- Gamified workout experience requires a subscription
- Rower is usable without subscription as a normal machine



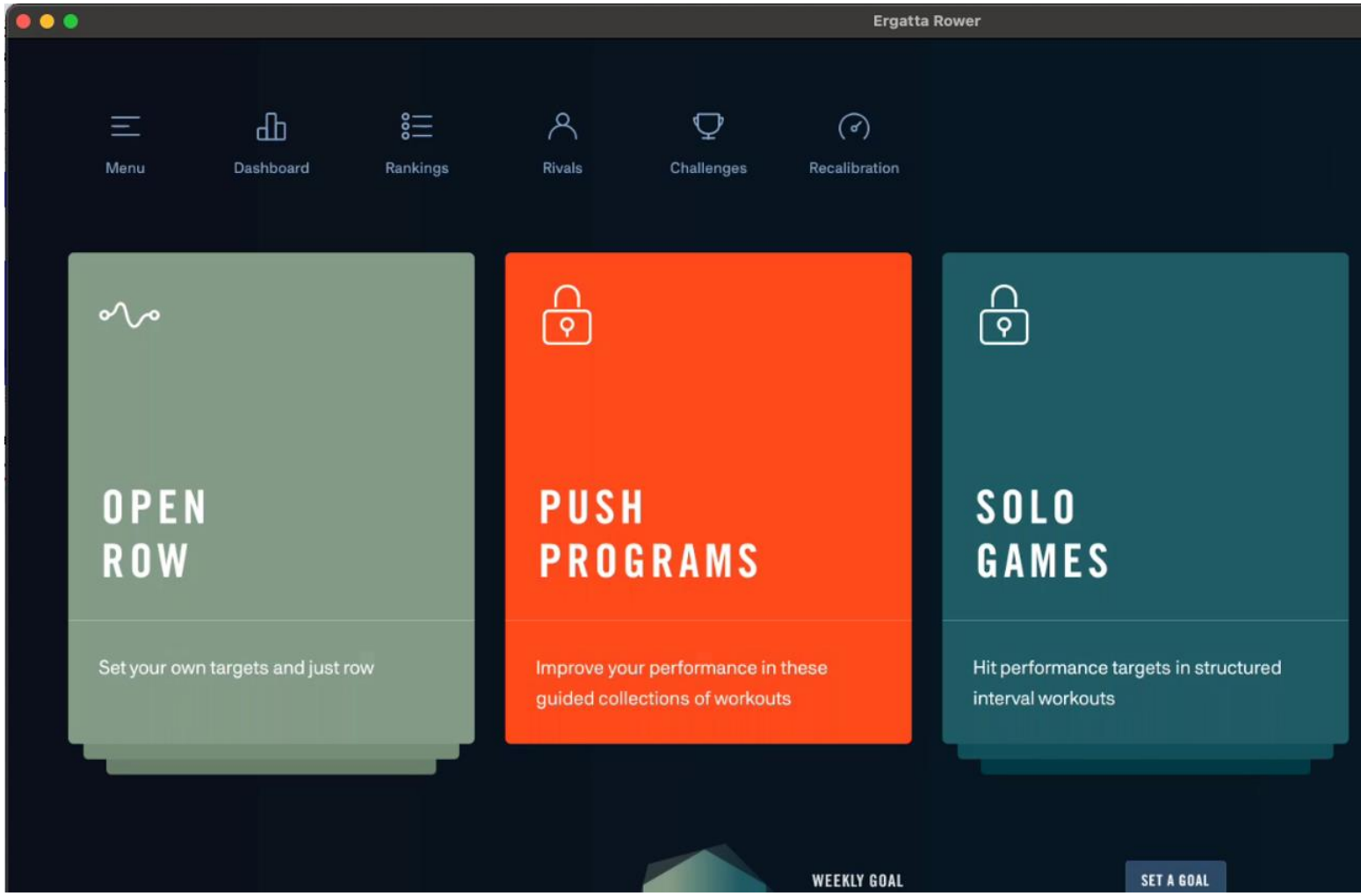
Phase 1:

Proxying and Manipulating Traffic

- What traffic is it sending?
- How is it handling authorizations?

The Lay of the Land

- Auto loads workout app on startup
- No virtual or physical buttons
- Locks denote modes that require a subscription
- Nmap port scan reported all ports closed

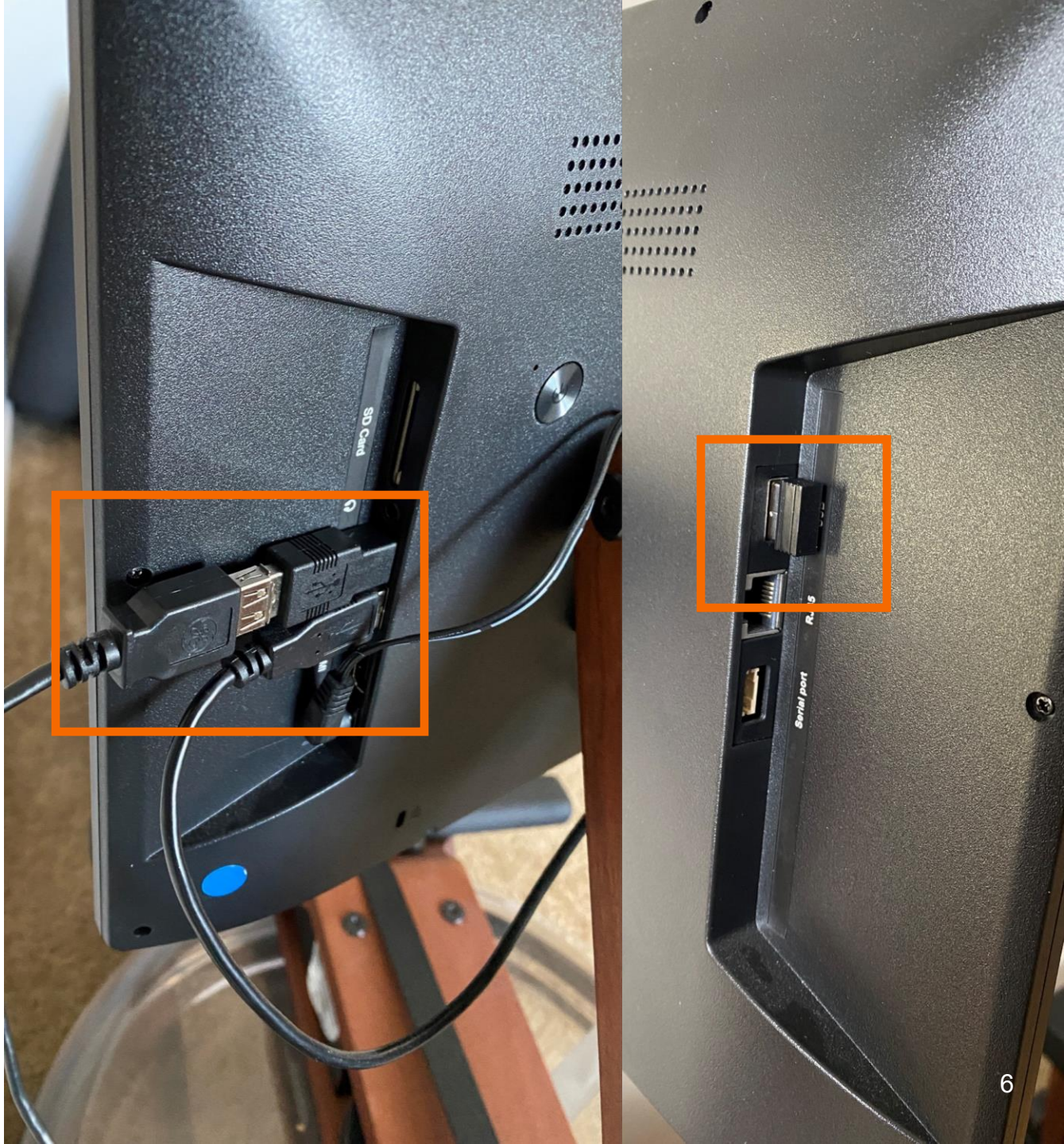


External Ports

- Two open USB ports on the back

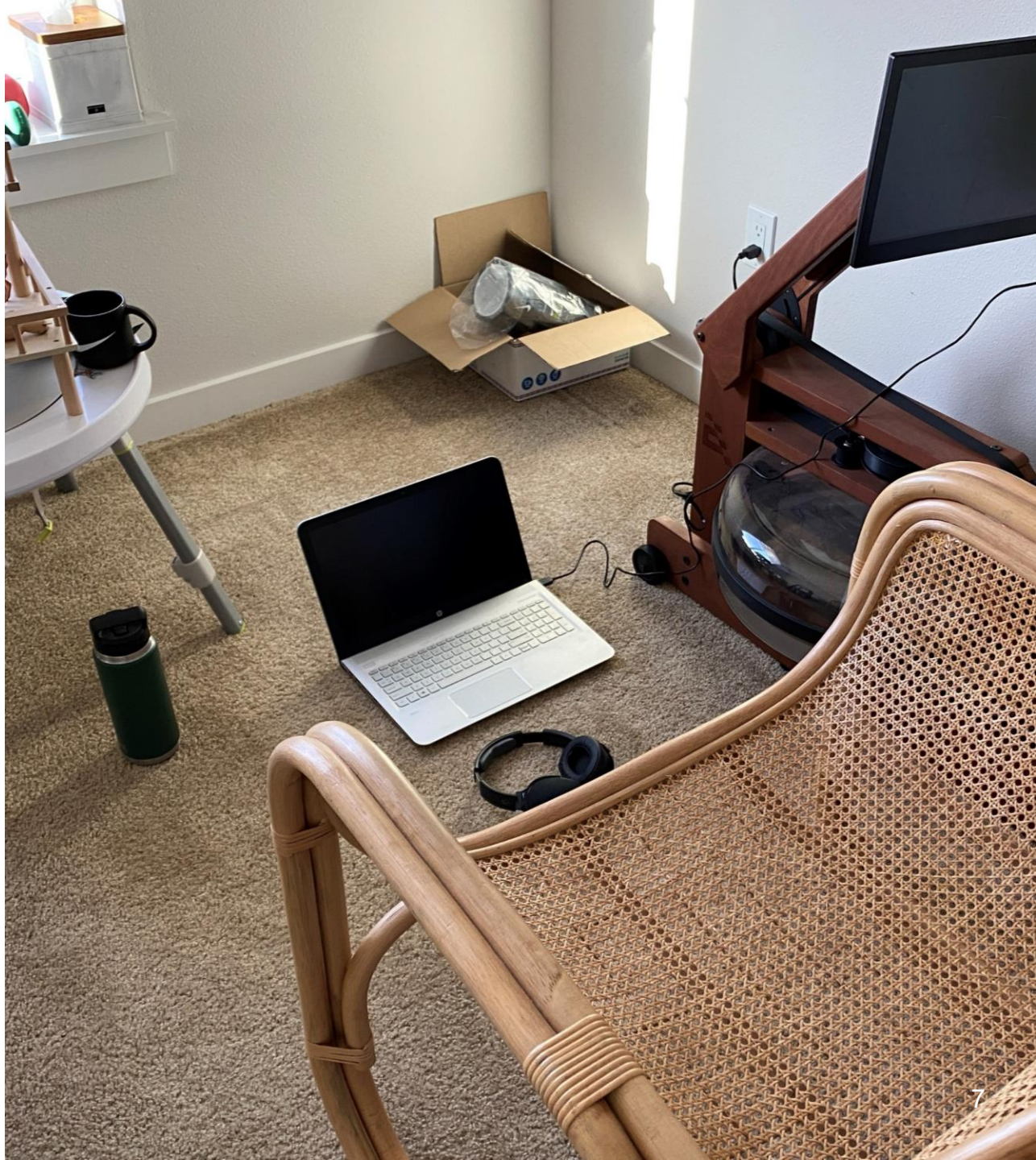
External Ports

- Two open USB ports on the back
- Keyboard dongle on the left
- Mini-USB adapter on the right connected to laptop



Next Steps

- alt+tab to switch views on the tablet display
- Loaded the settings menu
- Enabled developer options
- Enabled USB debugging



Android Debug Bridge (ADB)

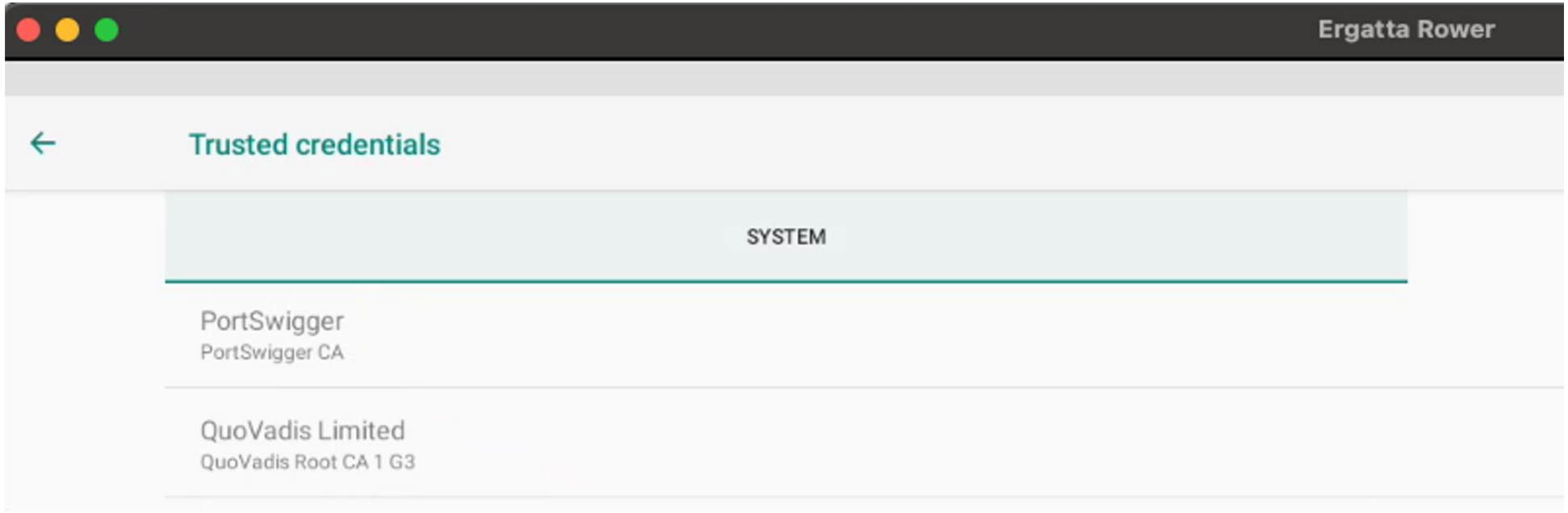
- Rename Burp cert to be compatible with Android

```
┌──(assess|US-JPV2X61YKD|z)-[2.7.18]-[~/test]
└─$ ls
cacert.der

┌──(assess|US-JPV2X61YKD|z)-[2.7.18]-[~/test]
└─$ hashed_name=`openssl x509 -inform DER -subject_hash_old -in cacert.der | head -1` && cp cacert.der $hashed_name.0

┌──(assess|US-JPV2X61YKD|z)-[2.7.18]-[~/test]
└─$ ls
9a5ba575.0  cacert.der
```

- Remount system directory to be writable
- Push the cert
- Alter cert permissions
- Reboot

```
┌──(assess|US-JPV2X61YKD|z)-[2.7.18]-[~/test]
└─$ adb root
restarting adbd as root

┌──(assess|US-JPV2X61YKD|z)-[2.7.18]-[~/test]
└─$ adb remount
remount succeeded

┌──(assess|US-JPV2X61YKD|z)-[2.7.18]-[~/test]
└─$ adb push 9a5ba575.0 /system/etc/security/cacerts
9a5ba575.0: 1 file pushed, 0 skipped. 3.7 MB/s (940 bytes in 0.000s)

┌──(assess|US-JPV2X61YKD|z)-[2.7.18]-[~/test]
└─$ adb shell chmod 664 /system/etc/security/cacerts/9a5ba575.0

┌──(assess|US-JPV2X61YKD|z)-[2.7.18]-[~/test]
└─$ adb reboot
```

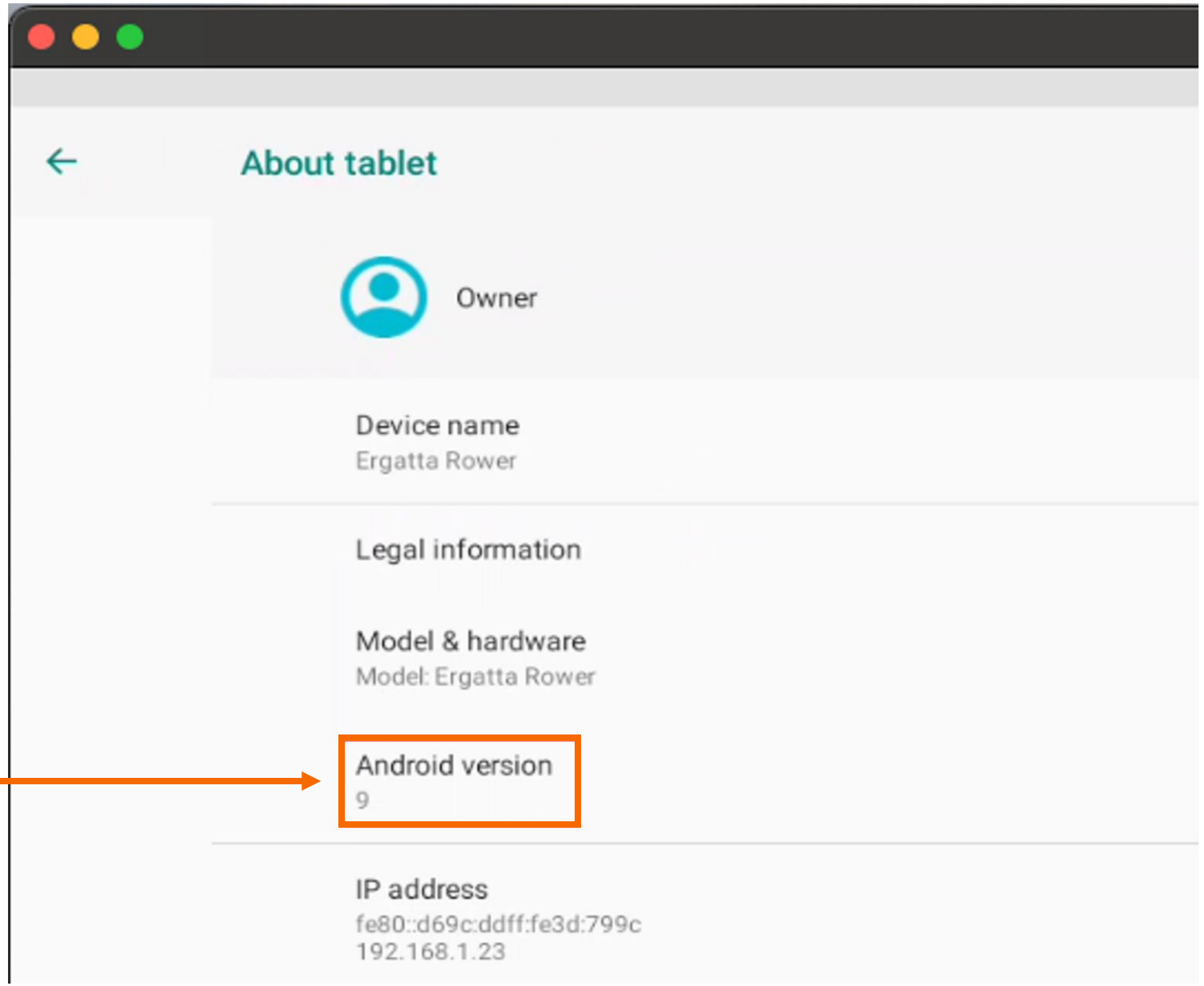
Burp CA in Trusted Credentials



Wait…

You can’t just remount system to be writable in current versions of Android.

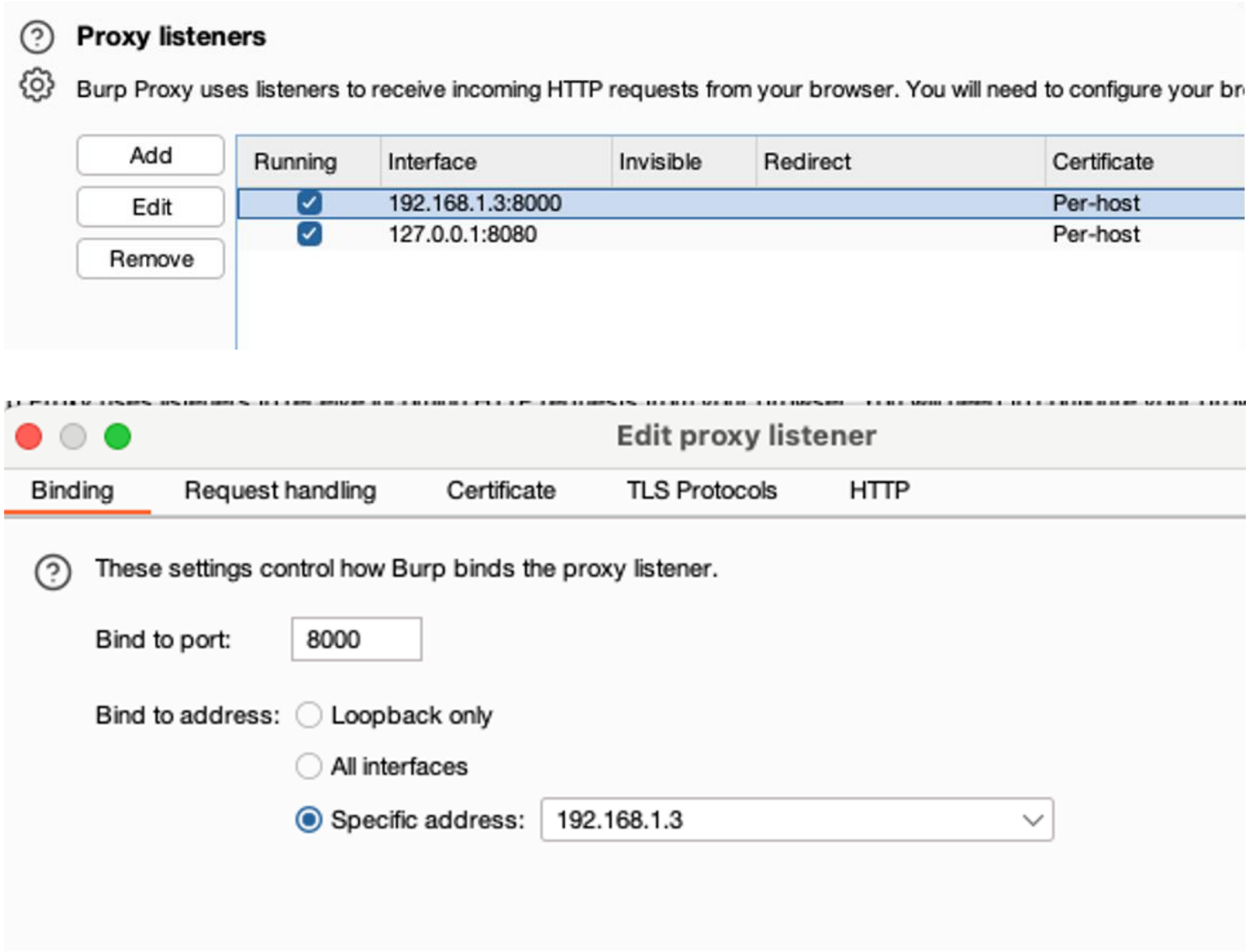
Android Version



Configure Burp Proxy Listener

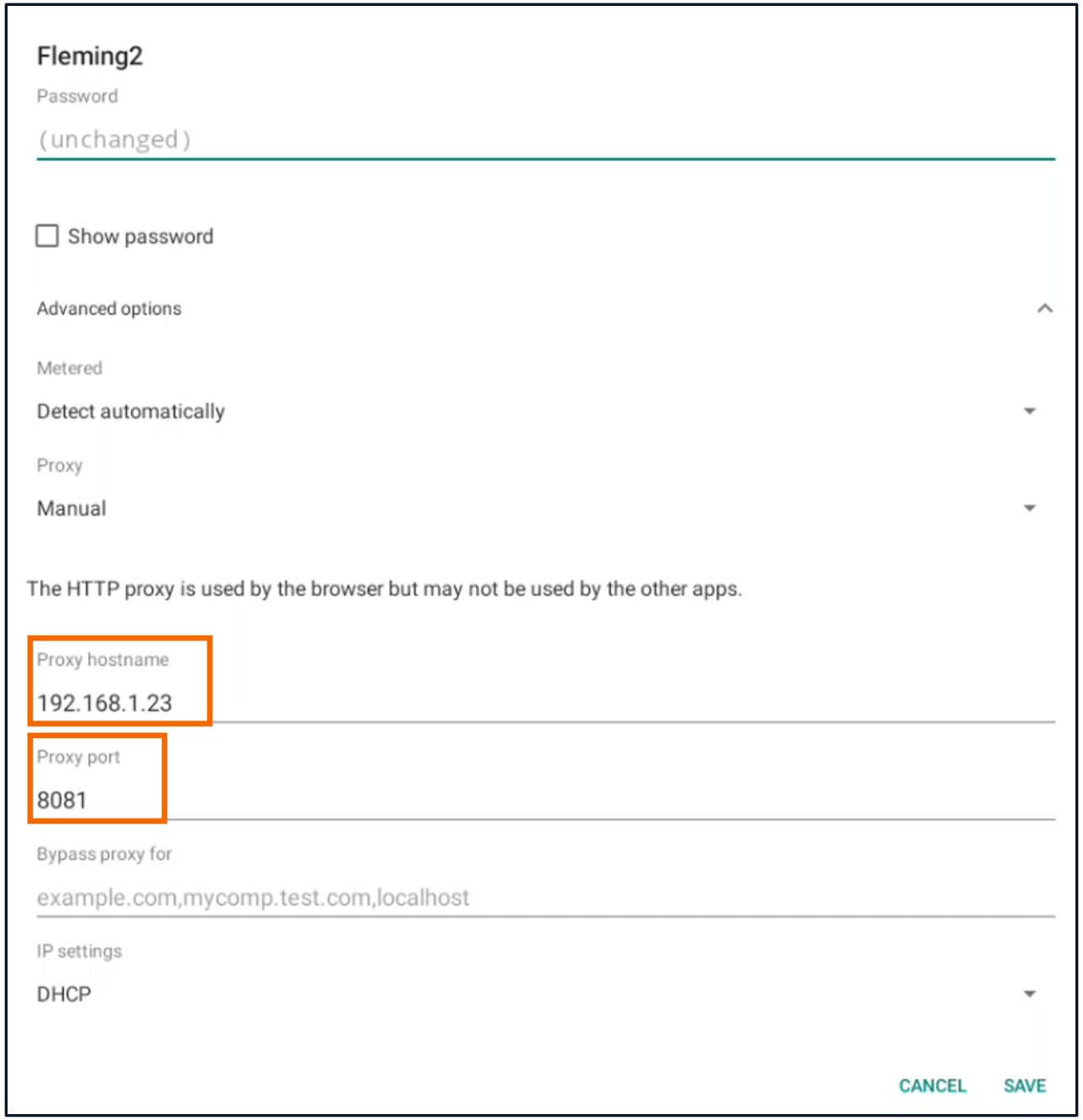
From Proxy Listeners:

- Select Edit
- Specific Address
- Then the internal network interface

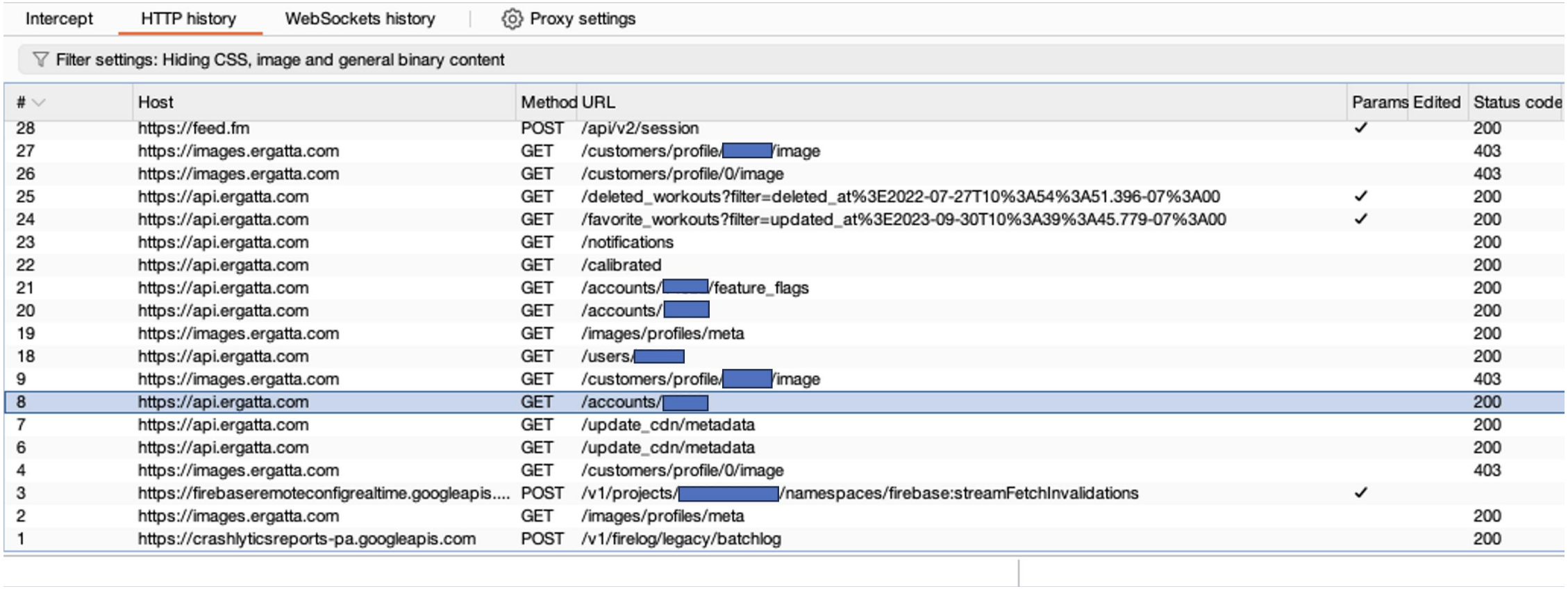


Rower Upstream Proxy

- Edit the WiFi connection
- Set the proxy to manual
- Set proxy hostname and proxy port of your Proxy Listener



Check the Proxy



Intercept | HTTP history | WebSockets history | Proxy settings

Filter settings: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code
28	https://feed.fm	POST	/api/v2/session	✓		200
27	https://images.ergatta.com	GET	/customers/profile/[redacted]/image			403
26	https://images.ergatta.com	GET	/customers/profile/0/image			403
25	https://api.ergatta.com	GET	/deleted_workouts?filter=deleted_at%3E2022-07-27T10%3A54%3A51.396-07%3A00	✓		200
24	https://api.ergatta.com	GET	/favorite_workouts?filter=updated_at%3E2023-09-30T10%3A39%3A45.779-07%3A00	✓		200
23	https://api.ergatta.com	GET	/notifications			200
22	https://api.ergatta.com	GET	/calibrated			200
21	https://api.ergatta.com	GET	/accounts/[redacted]/feature_flags			200
20	https://api.ergatta.com	GET	/accounts/[redacted]			200
19	https://images.ergatta.com	GET	/images/profiles/meta			200
18	https://api.ergatta.com	GET	/users/[redacted]			200
9	https://images.ergatta.com	GET	/customers/profile/[redacted]/image			403
8	https://api.ergatta.com	GET	/accounts/[redacted]			200
7	https://api.ergatta.com	GET	/update_cdn/metadata			200
6	https://api.ergatta.com	GET	/update_cdn/metadata			200
4	https://images.ergatta.com	GET	/customers/profile/0/image			403
3	https://firebaseremoteconfigrealtime.googleapis....	POST	/v1/projects/[redacted]/namespaces/firebase:streamFetchInvalidations	✓		
2	https://images.ergatta.com	GET	/images/profiles/meta			200
1	https://crashlyticsreports-pa.googleapis.com	POST	/v1/firelog/legacy/batchlog			200

Request for billing_state

Request

Pretty Raw Hex JSON Web Tokens

```
GET /accounts/[redacted] HTTP/2
Host: api.ergatta.com
X-Rudder-Version: 33
X-Seaborn-Flavor: standard
Authorization: Bearer
[redacted]
User-Agent: Seaborn/2024.08.01/1001
Accept-Encoding: gzip, deflate, br
```

Response

Pretty Raw Hex Render Copy Response

```
HTTP/2 200 OK
Date: Mon, 23 Sep 2024 17:52:03 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 195
Vary: Accept-Encoding

{
  "message":"ok",
  "data":{
    "billing_state":"Canceled",
    "cbSubscriptionId":"[redacted]",
    "rootUserId":[redacted],
    "countryCode":"US",
    "partner":null,
    "id":[redacted],
    "created_at":"[redacted]"
  }
}
```

Request for billing_state

Request for feature_flags

Request

Pretty Raw Hex JSON Web Tokens

```
GET /accounts/[redacted]/feature_flags HTTP/2
Host: api.ergatta.com
X-Rudder-Version: 33
X-Seaborn-Flavor: standard
Authorization: Bearer
[redacted]
User-Agent: Seaborn/2024.08.01/1001
Accept-Encoding: gzip, deflate, br
```

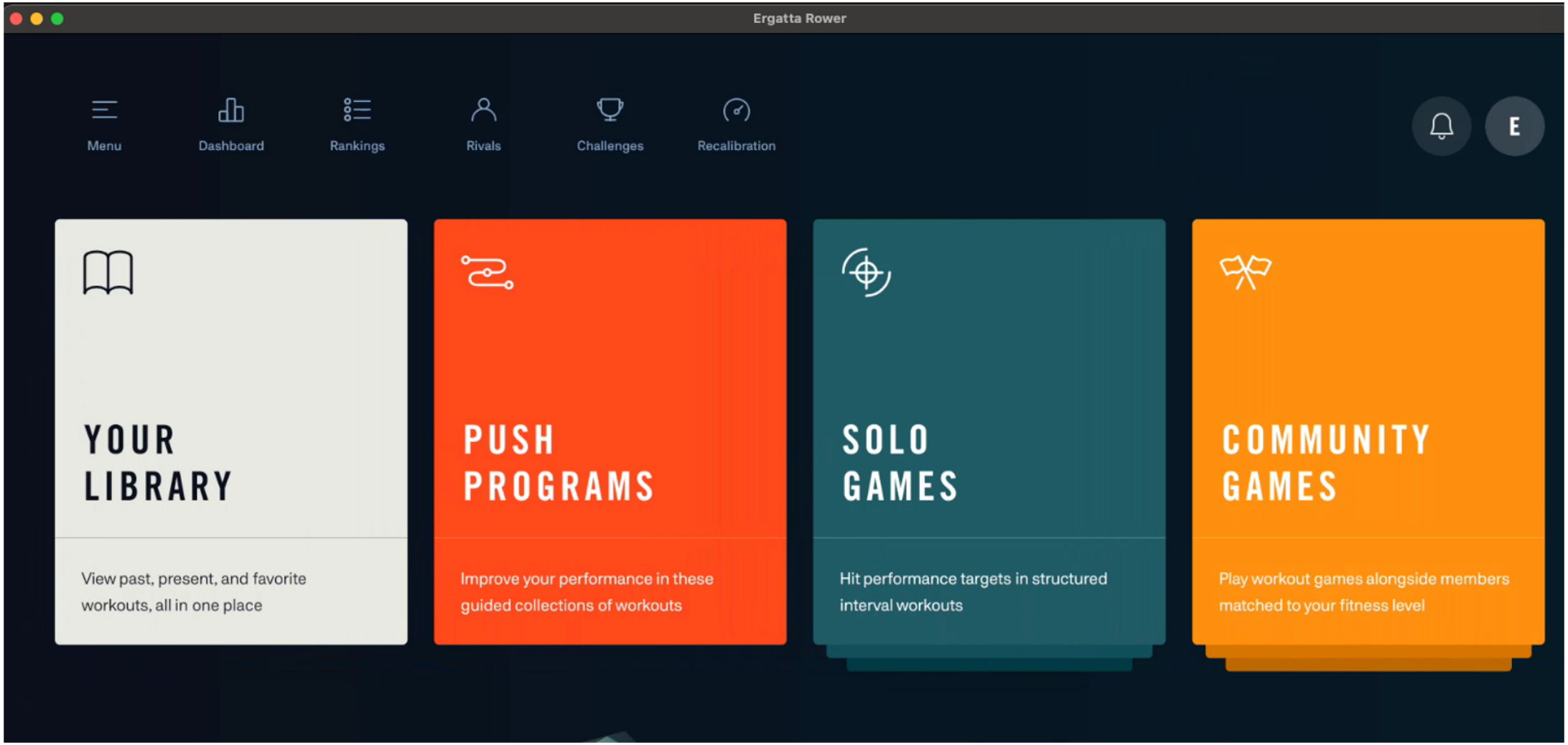
Response

Pretty Raw Hex Render Copy Response

```
HTTP/2 200 OK
Date: Mon, 23 Sep 2024 17:52:03 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 238
Vary: Accept-Encoding

{
  "message":"success",
  "data":{
    "accountID":[redacted],
    "canaryUpdateChannel":false,
    "liveRaceUI":false,
    "gameInfinity":false,
    "classes":true,
    "updateChannels":[
      {
        "name":"Main",
        "baseUrl":"https://appupdates.ergatta.com/Se
      }
    ]
  }
}
```

Request for feature_flags

UI after Editing Requests

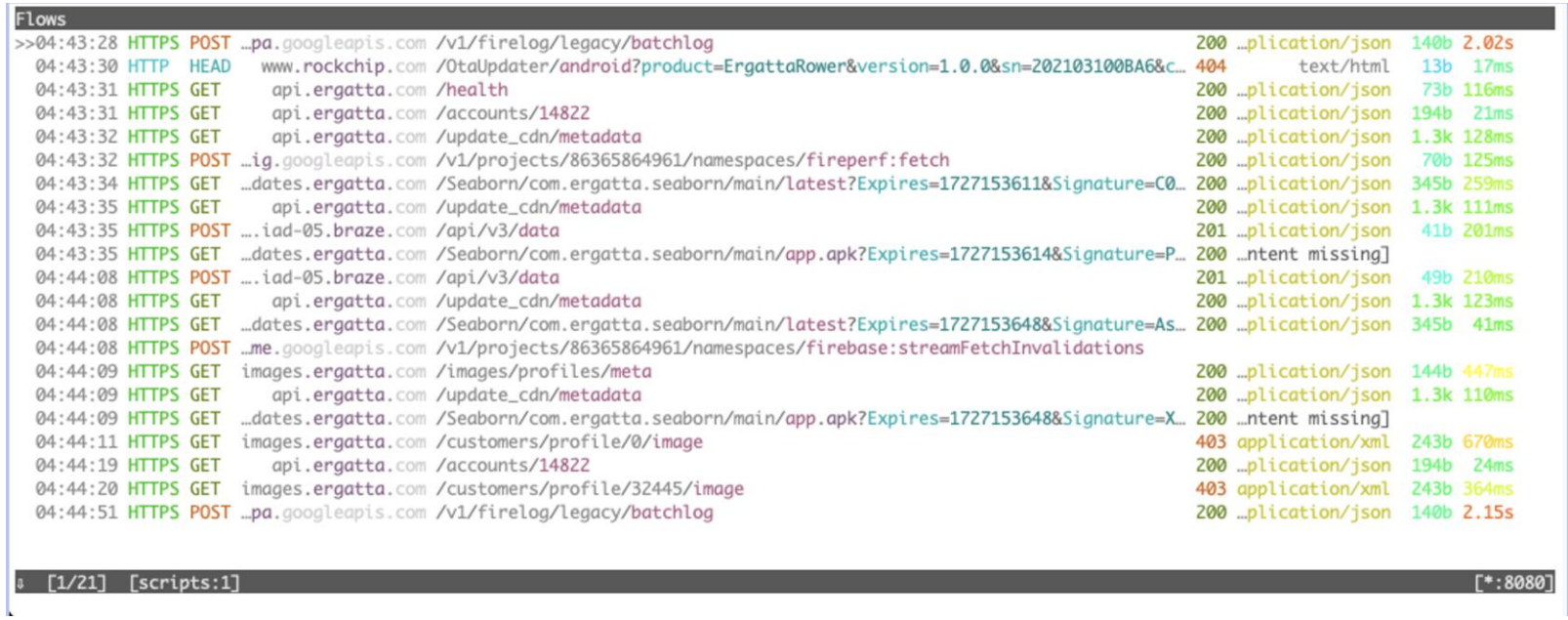


Phase 2:
Automation

- The rower will go through this flow several times while navigating the UI.
- Is there a good way to automate this?

Mitmproxy

- Intercepting proxy
- Light weight
- Open source
- Runs well from the Command Line

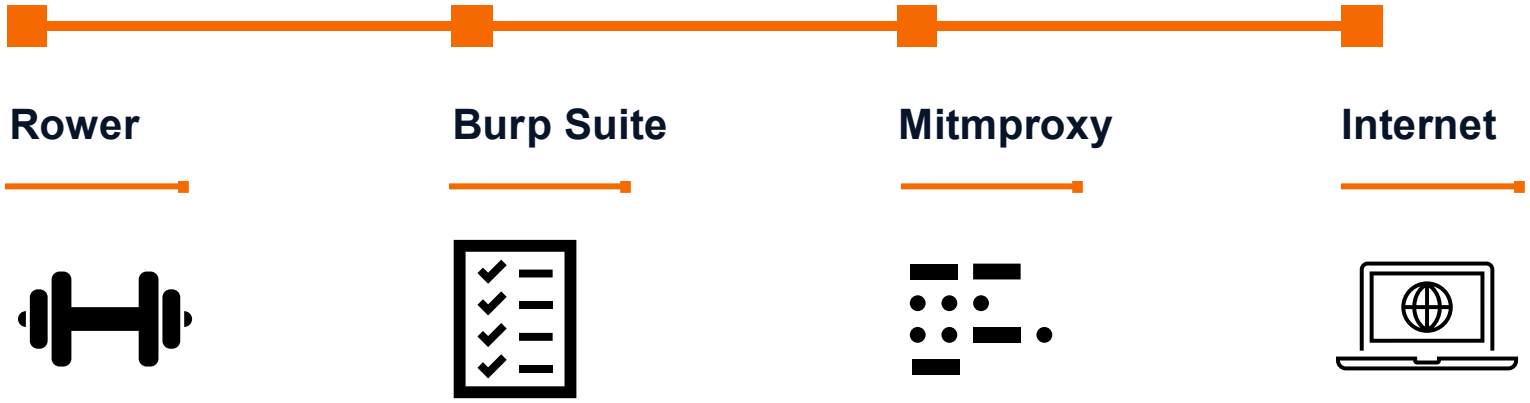


Mitproxy AddOn Structure

```python
# mitmproxy -s ./path/to/addon.py

from mitmproxy import http

class ChangeSomeTraffic:

    def __init__(self):
        pass

    # mitmproxy response event
    def response(self, flow: http.HTTPFlow) -> None:
        if 'target.host' in flow.request.host:     #if our target host is in the host for a request
            modified_response = open("some/file", "r").read()   #open file
            flow.response.text = modified_response #replace body in response with body from file
            print("Request modified!")

    # mitmproxy request event
    def request (self, flow:http.HTTPflow) -> None:
        match str(flow.request.host)
            case 'target.host': #if the host is the target host
                modified_response = open("some/file", "r").read()   #open file
                flow.response.text = modified_response #replace body in response with body from file
                print("Request modified!")

addons = [
    ChangeSubStatus()
]
```

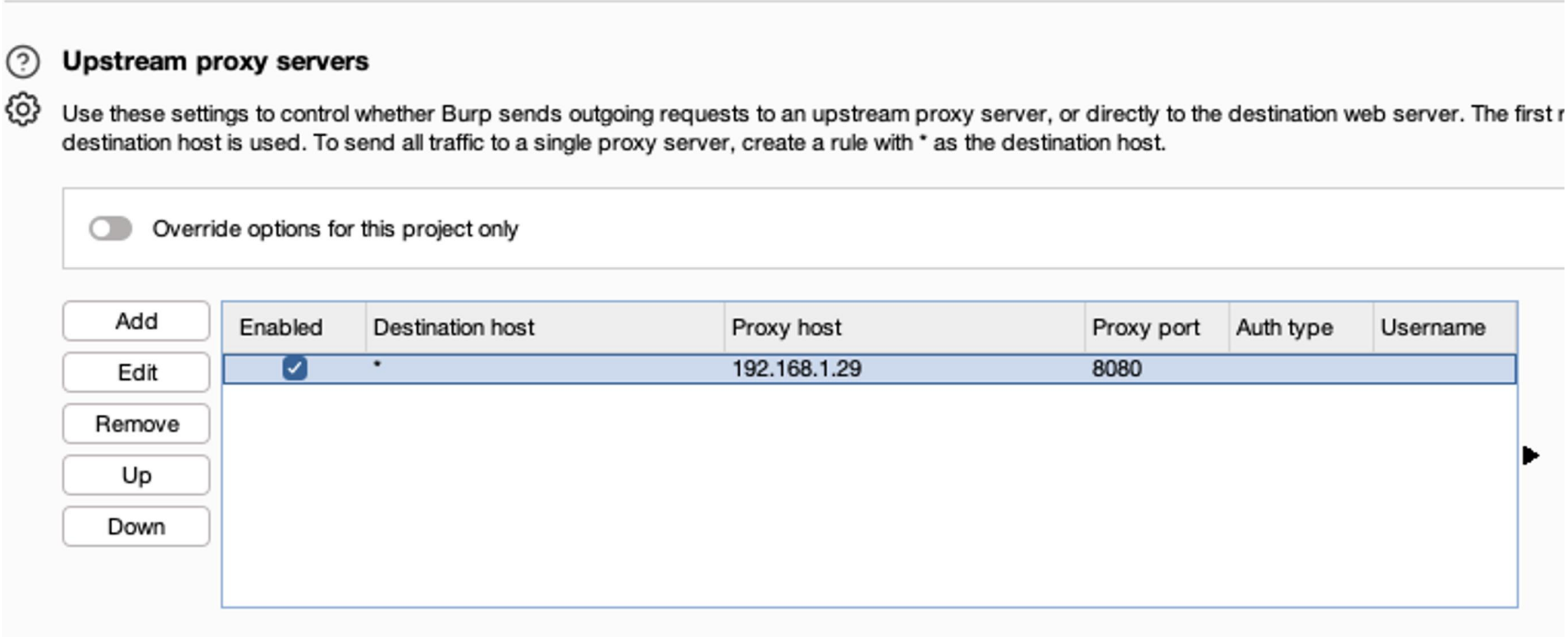
Proxy Testing

Traffic chain: Rower -> Burp -> Mitmproxy -> Internet



Burp Upstream Proxy Settings

Settings -> Network -> Connections -> Upstream proxy servers



First Mitmproxy AddOn Script

```
#mitmproxy response event
def response(self, flow: http,HTTPFlow) -> None:
    if 'api.ergatta.com' in flow.request.host:
        regex  = re.compile('/accounts/14822$')
        regex2 = re.compile('/accounts/14822/feature_flags')
        found  = regex.search(flow.request.path)
        found2 = regex2.search(flow.request.path)

        if found:
            modified_response = open("_accounts_14822.json", "r").read()
            flow.response.text = modified_response
            print("---------->Changed Billing Value<----------")
        elif found2:
            modified_response = open("_accounts_14822_feature_flags.json", "r").read()
            flow.response.text = modified_response
            print("---------->Changed Flag Value<----------")
```

Body Files for AddOn

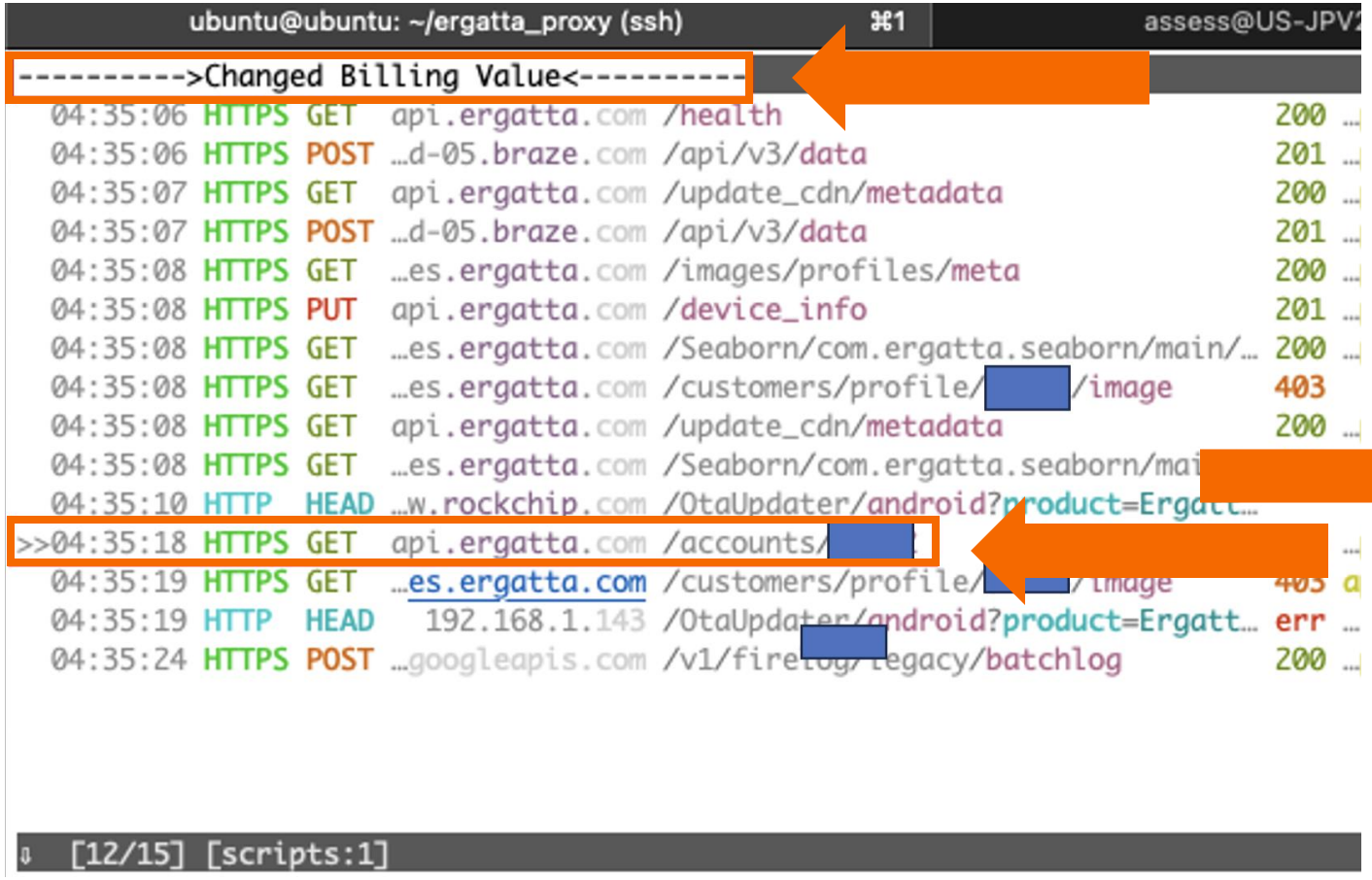
Account

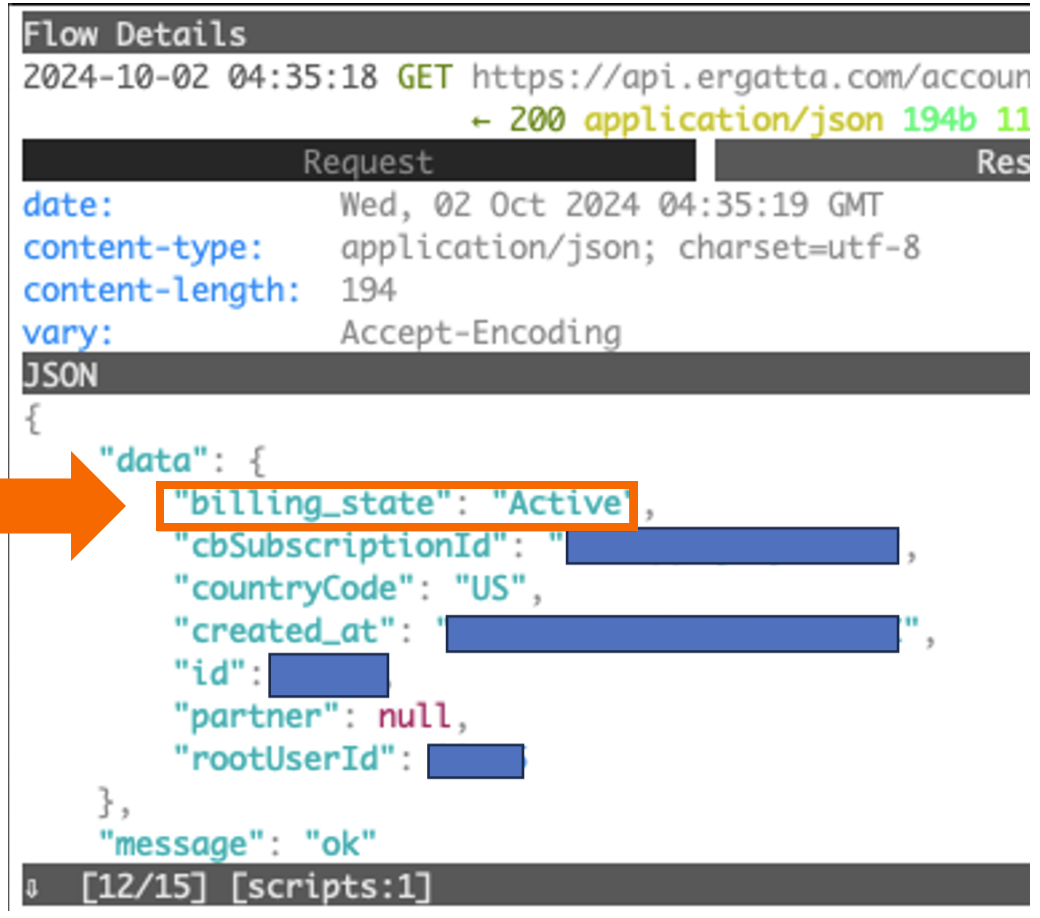

```
{"message":"ok","data":{"billing_state":"Active","cbSubscriptionId":"16CVqQSijd1y2EAVX","rootUserId":32445,
"countryCode":"US","partner":null,"id":14822,"created_at":"2021-09-11T20:06:55.618Z"}}
~
```


Feature Flags

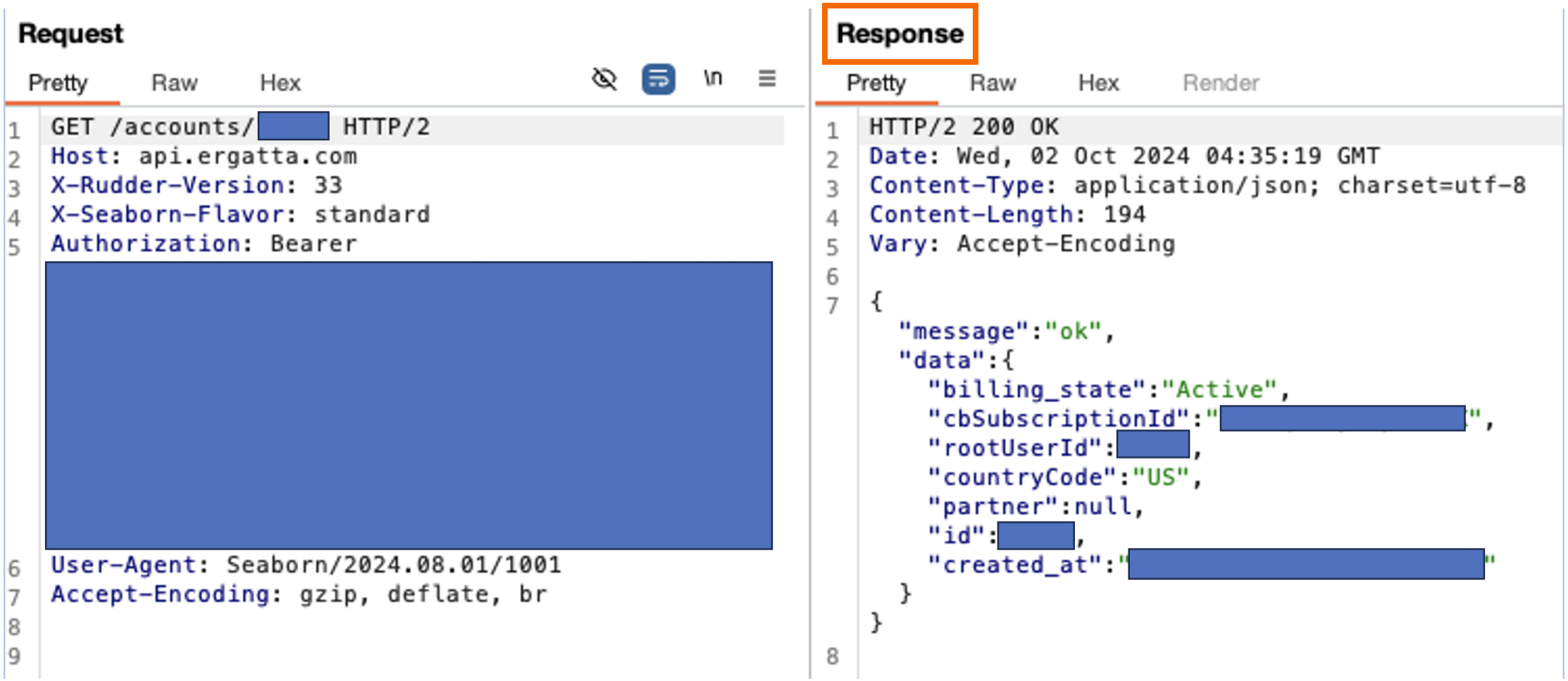

```
{"message":"success","data":{"accountID":14822,"canaryUpdateChannel":true,"liveRaceUI":true,"gameInfinity":
true,"classes":true,"updateChannels":[{"name":"Main","baseUrl":"https://appupdates.ergatta.com/Seaborn/$APP
LICATION_ID/main"}]}}
```

Test Run – It works!

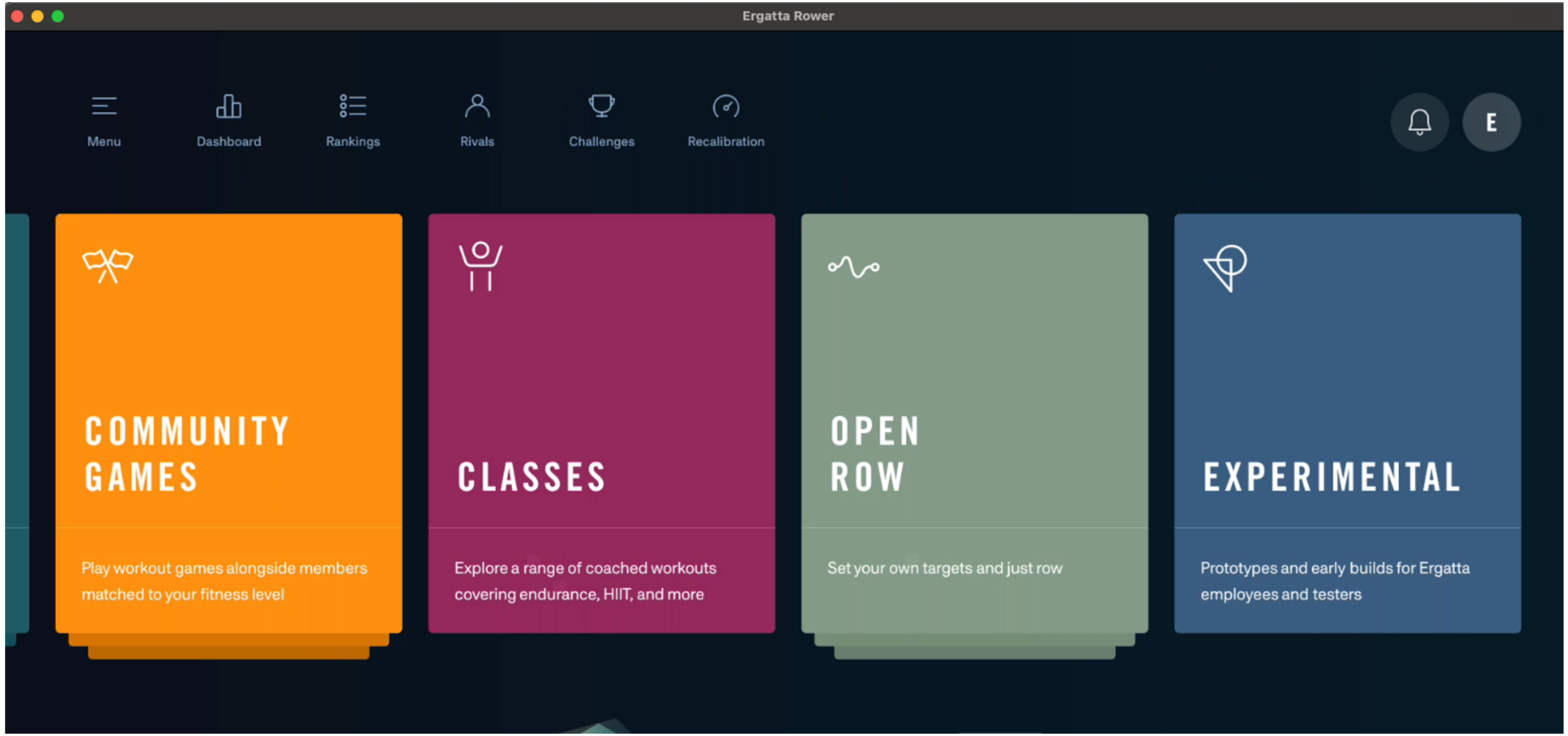




In Burp… No “Edited” Marker



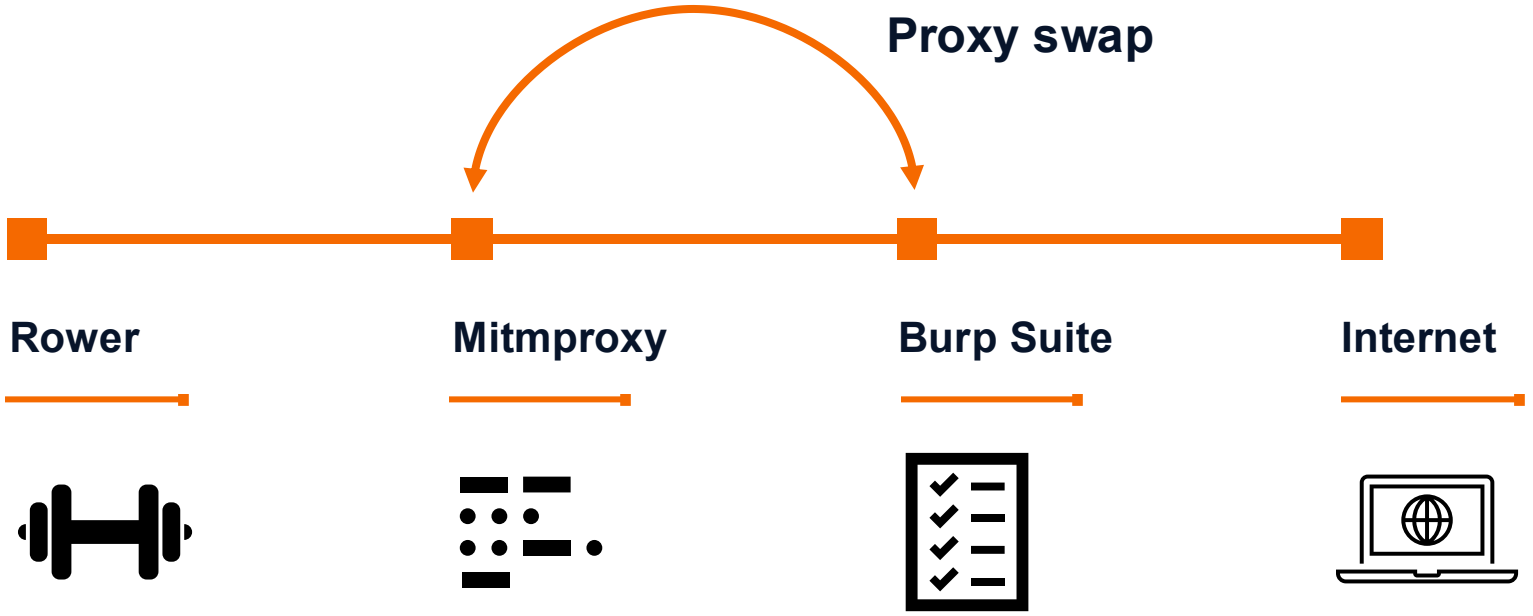
Confirming In-App – Extra Functions



Phase 3:
Handle All Responses

- Mitmproxy has an example that states it does not send any data to the server
- Confirm this function works as documented
- Mitmproxy service running on Raspberry Pi
- Flesh out the addon to handle all the requests coming from the rowing machine

Proxy Testing



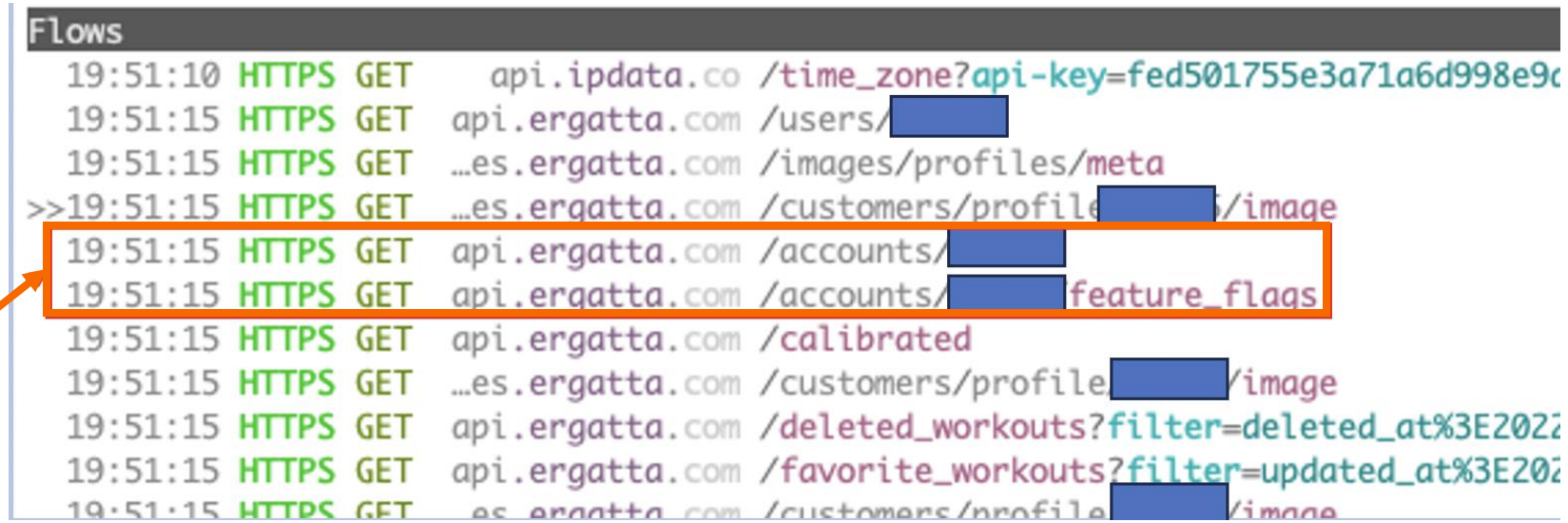
Update to AddOn

We are now acting on the request, not waiting for a response.

```
def request(self, flow: http.HTTPFlow) -> None:
    match str(flow.request.host):
        case 'api.ergatta.com': # if the host is a target host.
            path = str(flow.request.path)
            path = path.replace("/", "_")       #change / to _ to filter characters we can't use to name files (sanitize input)
            filename = find_file_regex.find(path) #search if we have a .json file that matches the path

            if filename:                        #if we have a match, fabricate a response using the .json body from the file
                response_body = open(filename, "r").read()
                flow.response = http.Response.make(
                    200,                            # (optional) status code
                    response_body,                  # (optional) content
                    {"Content-Type": "application/json; charset-utf-8"},  # (optional) headers
                    )
                print("---------------->"+filename+" supplied<-----------------")
```

Comparing Proxies

Note that these requests should have been here, but they are not.



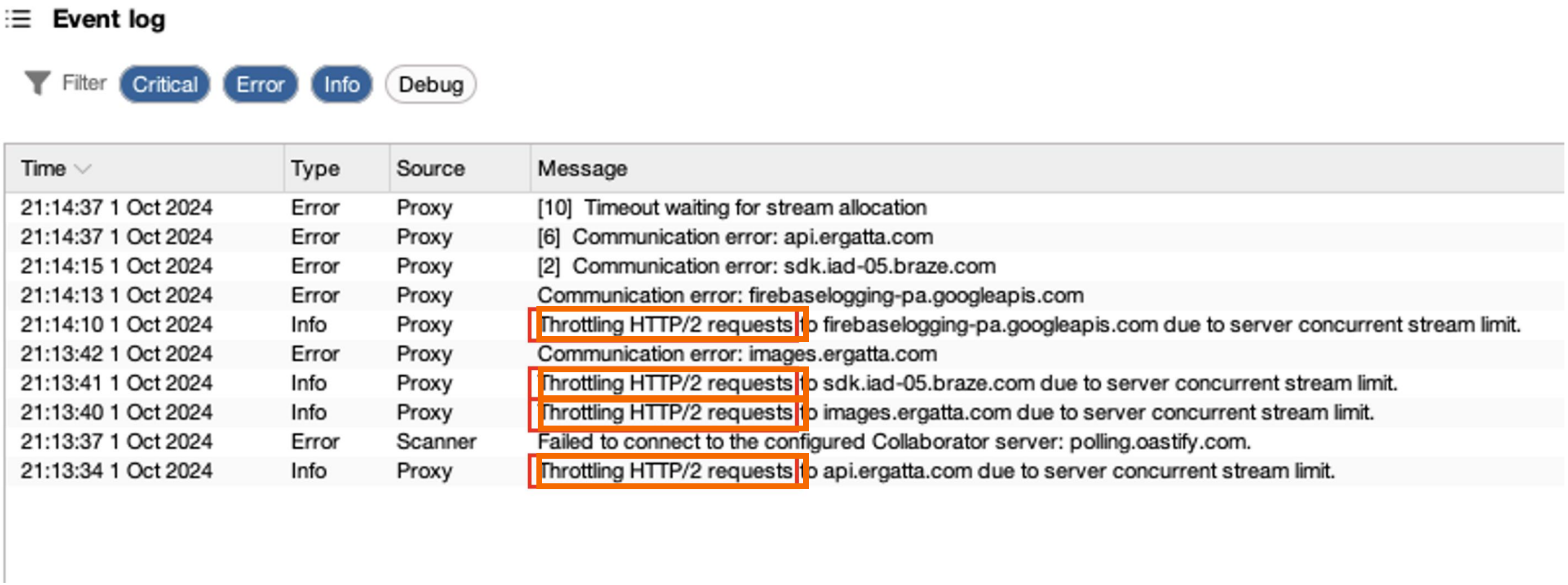
Time	Host	Metho	URL
12:51:15 3 Oct 2024	https://api.ergatta.com	GET	/users/
12:51:15 3 Oct 2024	https://images.ergatta.com	GET	/images/profiles/meta
12:51:15 3 Oct 2024	https://images.ergatta.com	GET	/customers/profile/ image
12:51:15 3 Oct 2024	https://api.ergatta.com	GET	/calibrated
12:51:15 3 Oct 2024	https://images.ergatta.com	GET	/customers/profile/ mage
12:51:15 3 Oct 2024	https://api.ergatta.com	GET	/deleted_workouts?filter=deleted_at%3E20
12:51:15 3 Oct 2024	https://api.ergatta.com	GET	/favorite_workouts?filter=updated_at%3E2

Command used to start mitmproxy:

mitmproxy -s ./AddOn.py --mode upstream:http://<ip-address>:<port> --ssl-insecure

Pitfall

Don’t point your proxies at each other or this will happen to you



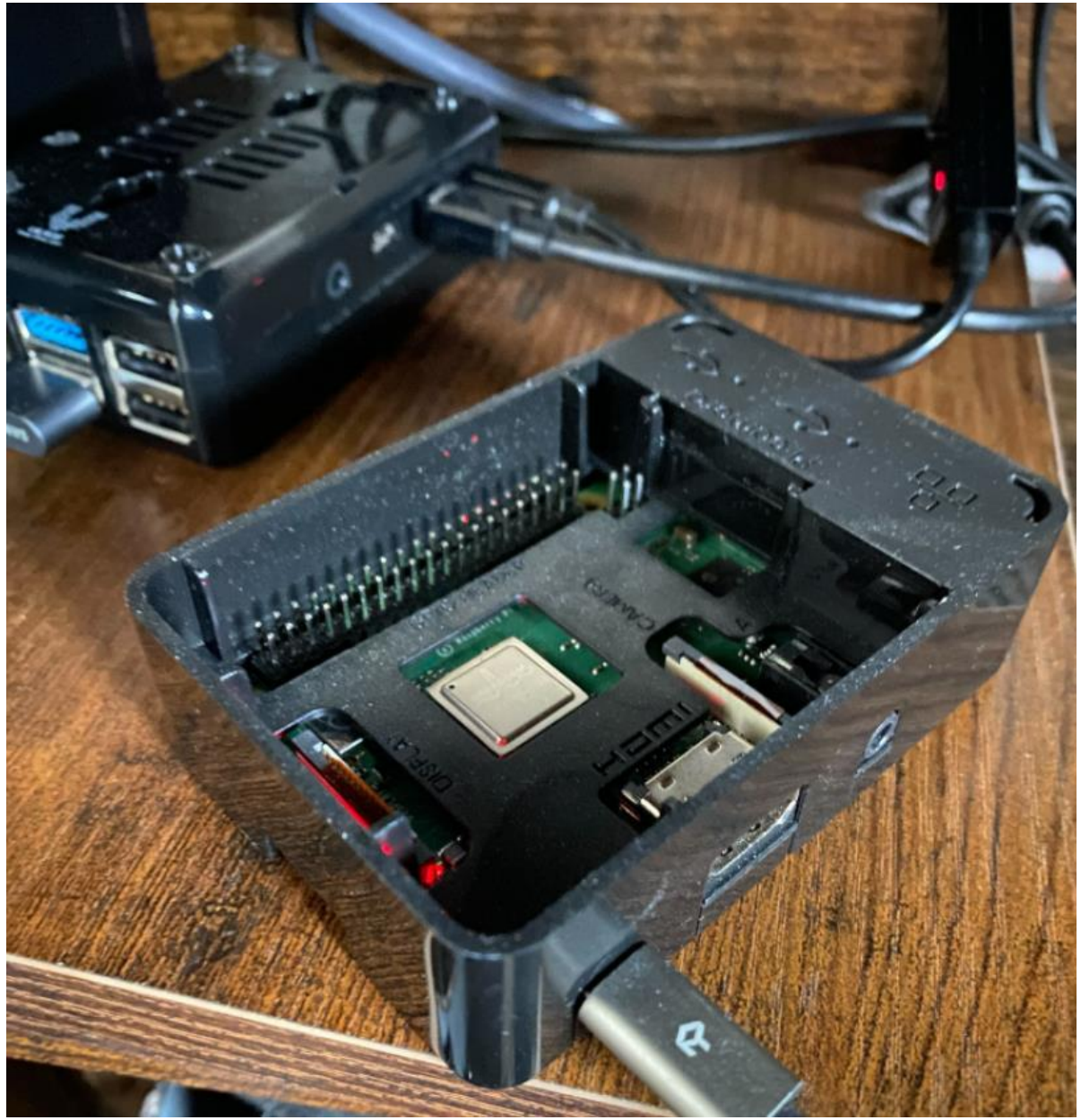
Event log

Filter Critical Error Info Debug

Time	Type	Source	Message
21:14:37 1 Oct 2024	Error	Proxy	[10] Timeout waiting for stream allocation
21:14:37 1 Oct 2024	Error	Proxy	[6] Communication error: api.ergatta.com
21:14:15 1 Oct 2024	Error	Proxy	[2] Communication error: sdk.iad-05.braze.com
21:14:13 1 Oct 2024	Error	Proxy	Communication error: firebaselogging-pa.googleapis.com
21:14:10 1 Oct 2024	Info	Proxy	Throttling HTTP/2 requests to firebaselogging-pa.googleapis.com due to server concurrent stream limit.
21:13:42 1 Oct 2024	Error	Proxy	Communication error: images.ergatta.com
21:13:41 1 Oct 2024	Info	Proxy	Throttling HTTP/2 requests to sdk.iad-05.braze.com due to server concurrent stream limit.
21:13:40 1 Oct 2024	Info	Proxy	Throttling HTTP/2 requests to images.ergatta.com due to server concurrent stream limit.
21:13:37 1 Oct 2024	Error	Scanner	Failed to connect to the configured Collaborator server: polling.oastify.com.
21:13:34 1 Oct 2024	Info	Proxy	Throttling HTTP/2 requests to api.ergatta.com due to server concurrent stream limit.

Raspberry Pi 4 Model B

- Ubuntu Server Headless
- WiFi Connected



Mitmdump

- Even more CLI than Mitmproxy

```
ubuntu@ubuntu:~/ergatta_proxy$ mitmdump -s ./ergatta-proxy.py
Loading script ./ergatta-proxy.py
Proxy server listening at *:8080
192.168.1.16:53820: client connect
192.168.1.16:53822: client connect
192.168.1.16:53822: server connect firebaselogging-pa.googleapis.com:443 (74.125.1
192.168.1.16:53824: client connect
192.168.1.16:53820: server connect api.ergatta.com:443 (3.141.73.29:443)
192.168.1.16:53824: error establishing server connection: Multiple exceptions: [Er
 111] Connect call failed ('::', 443, 0, 0)
192.168.1.16:53824: CONNECT app-measurement.com:443
                 << 502 Bad Gateway 163b
192.168.1.16:53824: client disconnect
192.168.1.16:53828: client connect
192.168.1.16:53828: error establishing server connection: Multiple exceptions: [Er
 111] Connect call failed ('::', 443, 0, 0)
192.168.1.16:53828: CONNECT app-measurement.com:443
                 << 502 Bad Gateway 163b
192.168.1.16:53828: client disconnect
192.168.1.16:53820: GET https://api.ergatta.com/health HTTP/2.0
        << HTTP/2.0 200 OK 73b
---------------->_accounts_[redacted].json supplied<----------------
192.168.1.16:53820: GET https://api.ergatta.com/accounts/[redacted] HTTP/2.0
        << HTTP/1.1 200 OK 194b
```

ergatta.service config

- Add <someservice>.service to /etc/system/system
- Note the WorkingDirectory parameter

```
[Unit]
Description=Ergatta mitmproxy addon service
After=network.target
[Service]
Type=simple
Restart=always
RestartSec=3
User=ubuntu
WorkingDirectory=/home/ubuntu/ergatta_proxy/
ExecStart=/home/ubuntu/ergatta_proxy/ergatta-script.sh

[Install]
WantedBy=multi-user.target
```

Ergatta-script.sh

```
#!/bin/bash
/usr/bin/mitmdump --ssl-insecure --set confdir=/home/ubuntu/.mitmproxy -s /home/ubuntu/ergatta_proxy/ergatta-proxy.py
 &>> /home/ubuntu/ergatta_proxy/mitmdump.log
~
```

Start ergatta.service

- If all goes well it should be active (running) status
- Use systemctl enable <someservice>.service to have it start automatically on boot

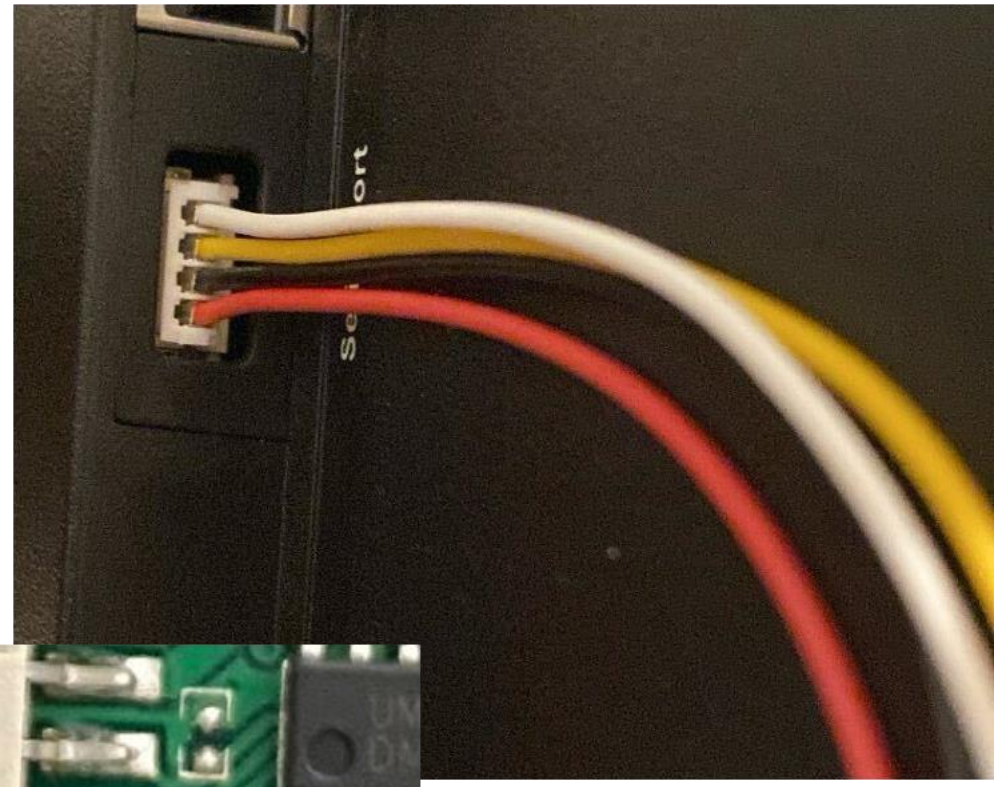
```
ubuntu@ubuntu:~/ergatta_proxy$ sudo systemctl daemon-reload
ubuntu@ubuntu:~/ergatta_proxy$ sudo systemctl start ergatta.service
ubuntu@ubuntu:~/ergatta_proxy$ sudo systemctl status ergatta.service
● ergatta.service - Ergatta mitmproxy addon service
     Loaded: loaded (/etc/systemd/system/ergatta.service; enabled; preset: enabled)
     Active: active (running) since Thu 2024-10-03 21:02:48 UTC; 9s ago
   Main PID: 18340 (ergatta-script.)
      Tasks: 3 (limit: 853)
     Memory: 44.5M (peak: 44.9M)
        CPU: 4.703s
     CGroup: /system.slice/ergatta.service
             ├─18340 /bin/bash /home/ubuntu/ergatta_proxy/ergatta-script.sh
             └─18341 /usr/bin/python3 /usr/bin/mitmdump --ssl-insecure --set confdir=/home/ubuntu/

Oct 03 21:02:48 ubuntu systemd[1]: Started ergatta.service - Ergatta mitmproxy addon service.
```

Installation & Current State

- Glued to the back of the monitor
- Powers up normally
- Powers down when rower does
- Might have some power warnings, but otherwise still seems to function

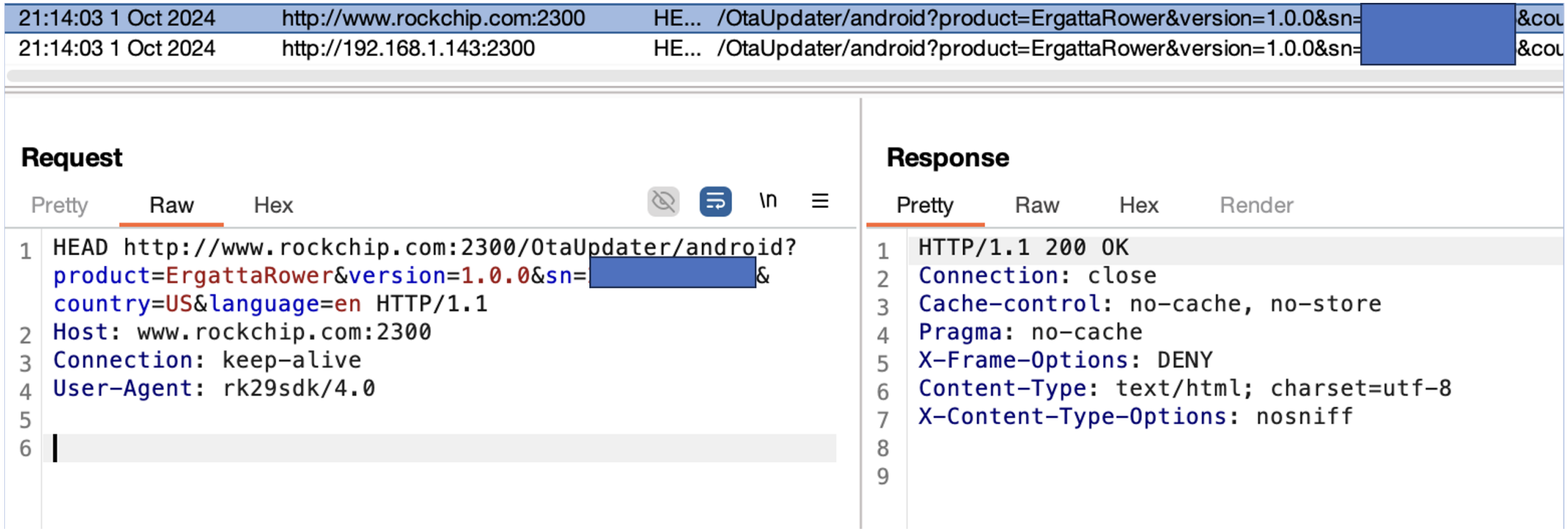
Next Steps

- 4 pin 2mm pitch PH connector w/ pigtail
- Looks like a UART port
- Common baud rates failed
- Logic analyzer to identify baud rate
- Sigrok rabbit hole



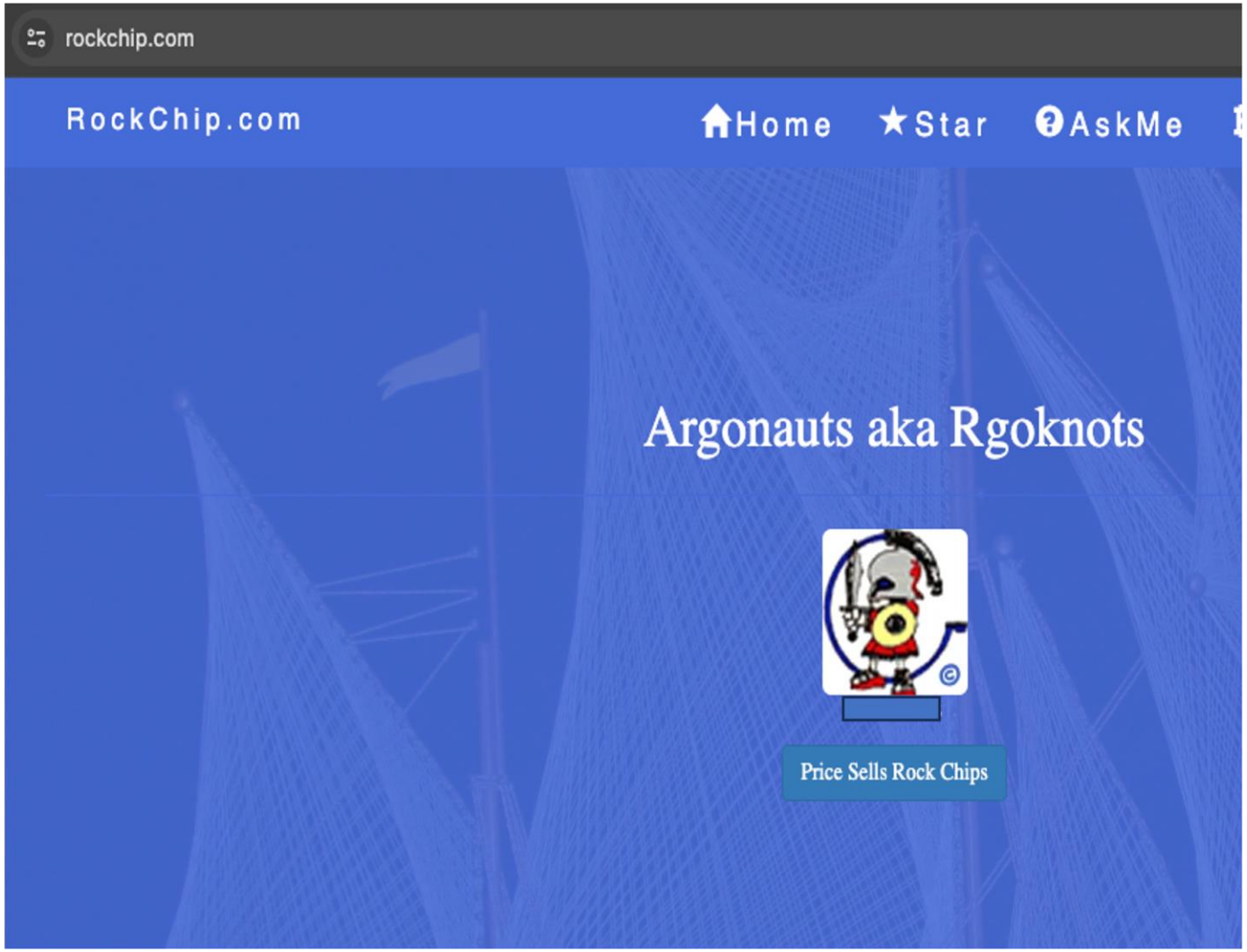
Firmware Update Domains

- Periodically reaches out to 2 domains for firmware updates
- Head requests are identical except for the Host



Rockchip.com

- Might not be a squatter, but…
- Doesn't look like a firmware distribution website to me

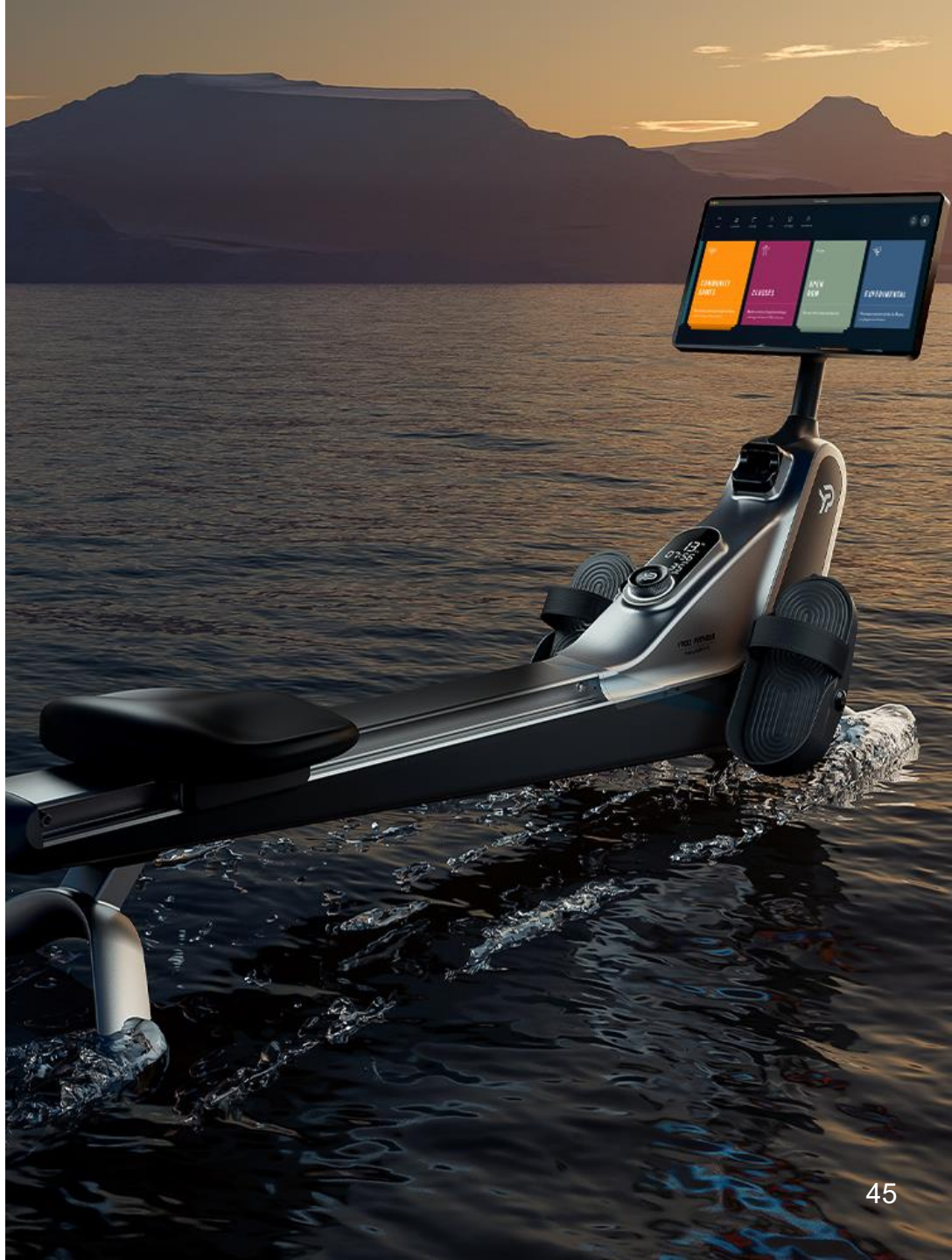


Rock-chips.com

- New domain, new you
- Tried replacing the domain for the firmware update, no dice

Disclosure Schedule

- **July 19th**
  - LinkedIn contact attempts.
- **July 25th**
  - Called customer support then emailed wireshark capture for the firmware updates to them and their CTO Prasanna Swaminathan.
- **September 20th**
  - Email follow up, and first mention of authZ issues.
- **September 23rd**
  - Prasanna responded and asked for details. Shared screenshots and high-level findings.
- **September 27th**
  - Zoom meeting with Prasanna to talk through the authZ findings. 2 month patching window.



Questions?

Shane Kell

Senior Security Consultant

shane.kell@netspi.com

241 N 5th Ave Suite 1200

Minneapolis, MN 55401

netspi.com