DNS Cache Poisoning Vulnerability
Explanation and Remedies

Viareggio, Italy
October 2008

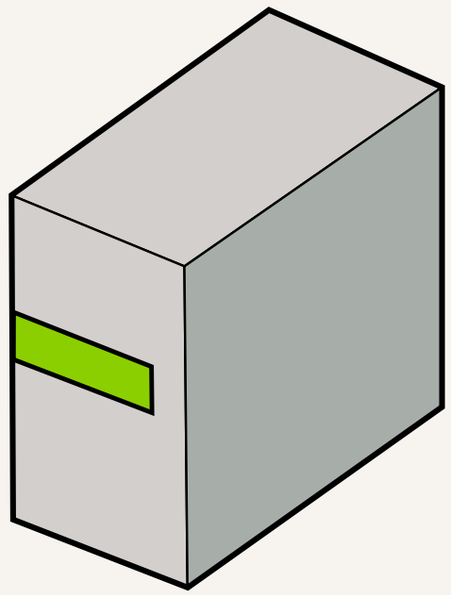
Kim Davies
Internet Assigned Numbers Authority

Internet Corporation for
Assigned Names & Numbers

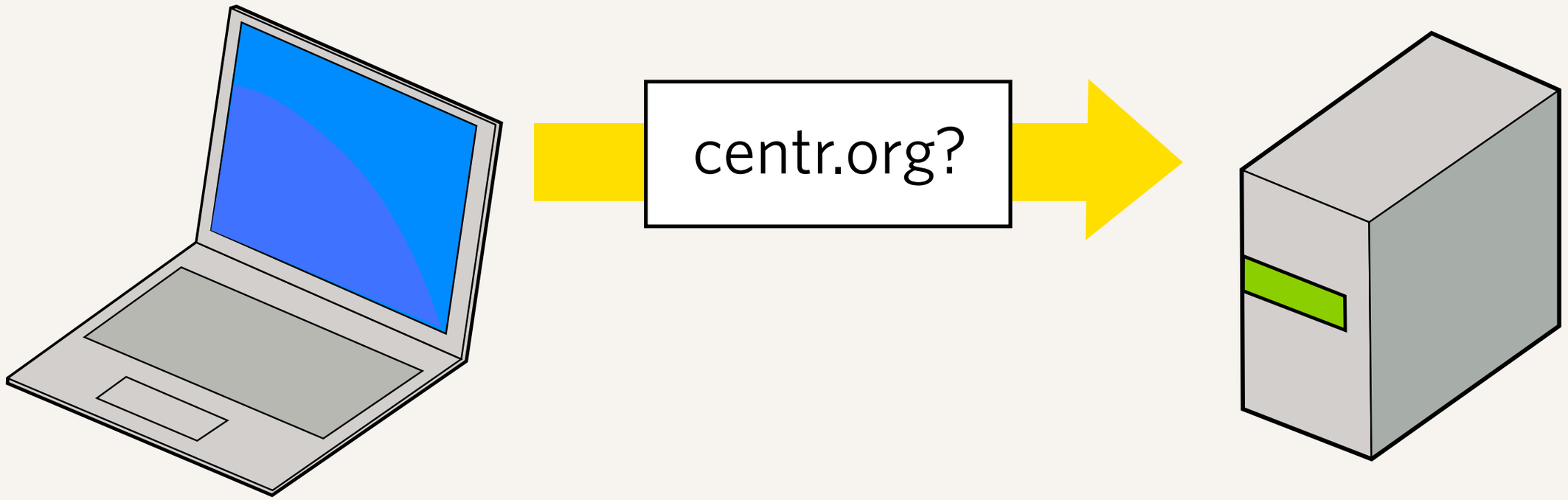
Agenda

- How do you attack the DNS?
- What has been discovered?
- Short term solutions
- Long term solutions
- Work ICANN has done to help

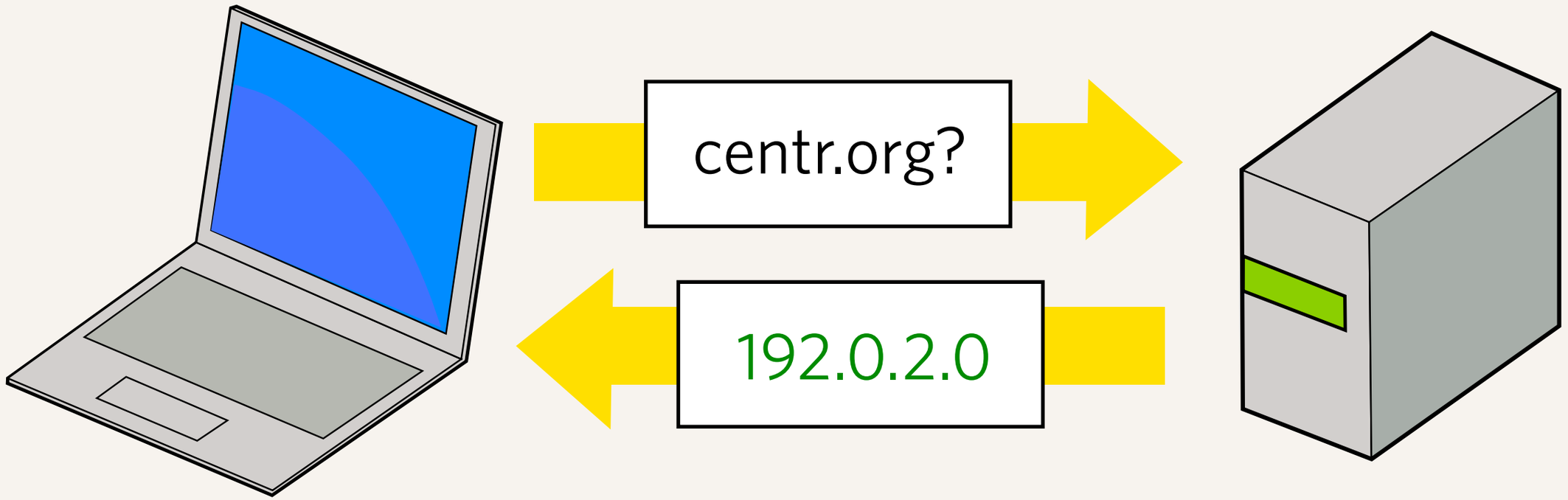
How do you attack the DNS?



A typical DNS query



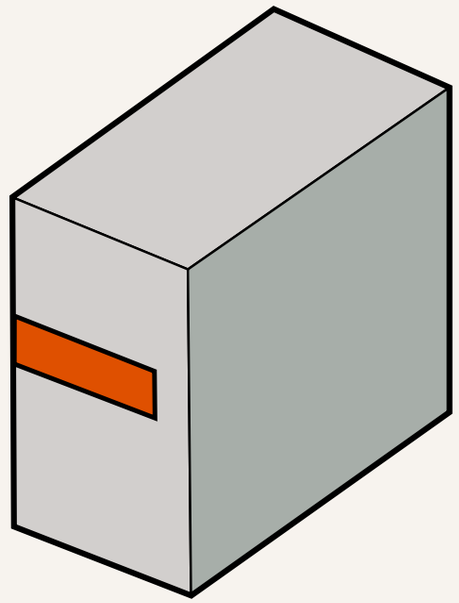
A typical DNS query



A typical DNS query

The DNS is not secure

- A computer sends a "question" to a DNS server, asking a question like "What is the IP address for aftld.org?"
- The computer gets an answer, and if the answer appears to match the question it asked, completely trusts that it is correct.
- There are multiple ways that traffic on the Internet can be intercepted and rerouted, or impersonated, so that the answer given is false.

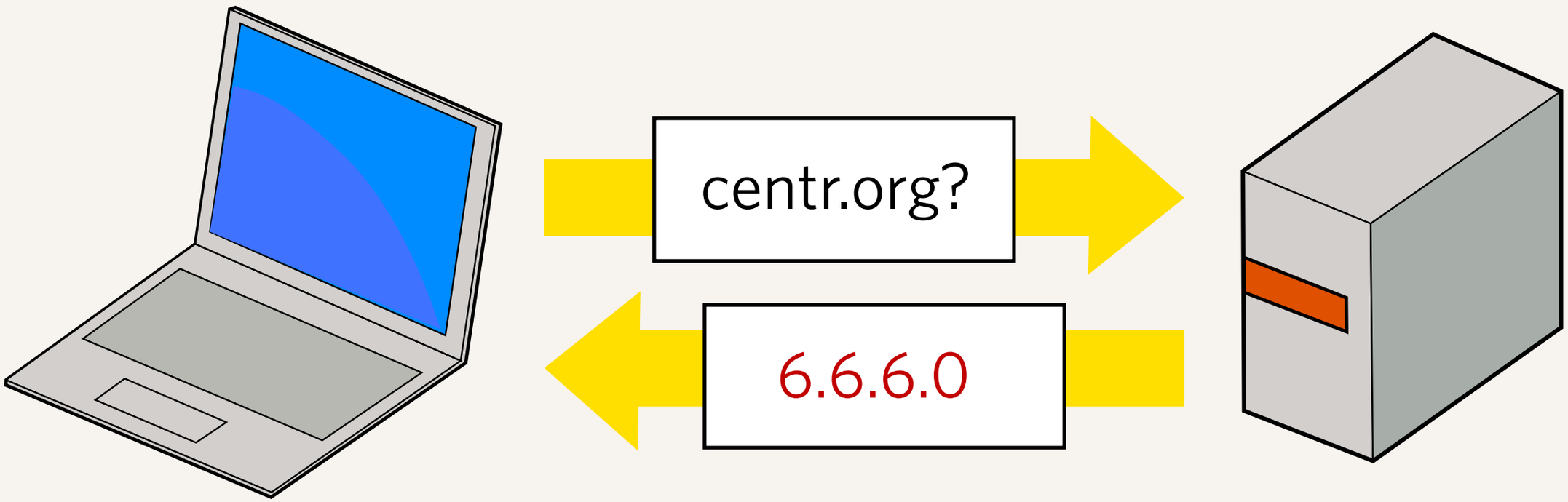


Receiving the wrong answer

- Something in the network between the computer and the server has intercepted or redirected the traffic.

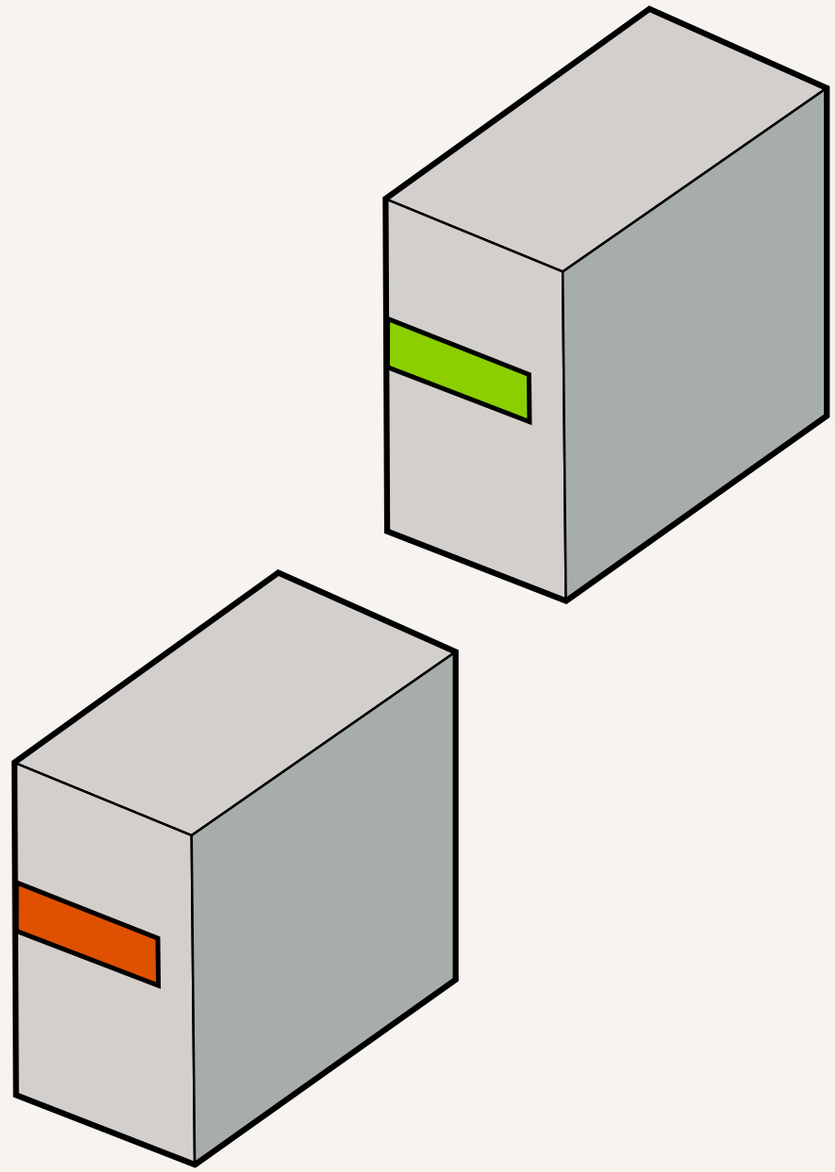
Receiving the wrong answer

- Something in the network between the computer and the server has intercepted or redirected the traffic.



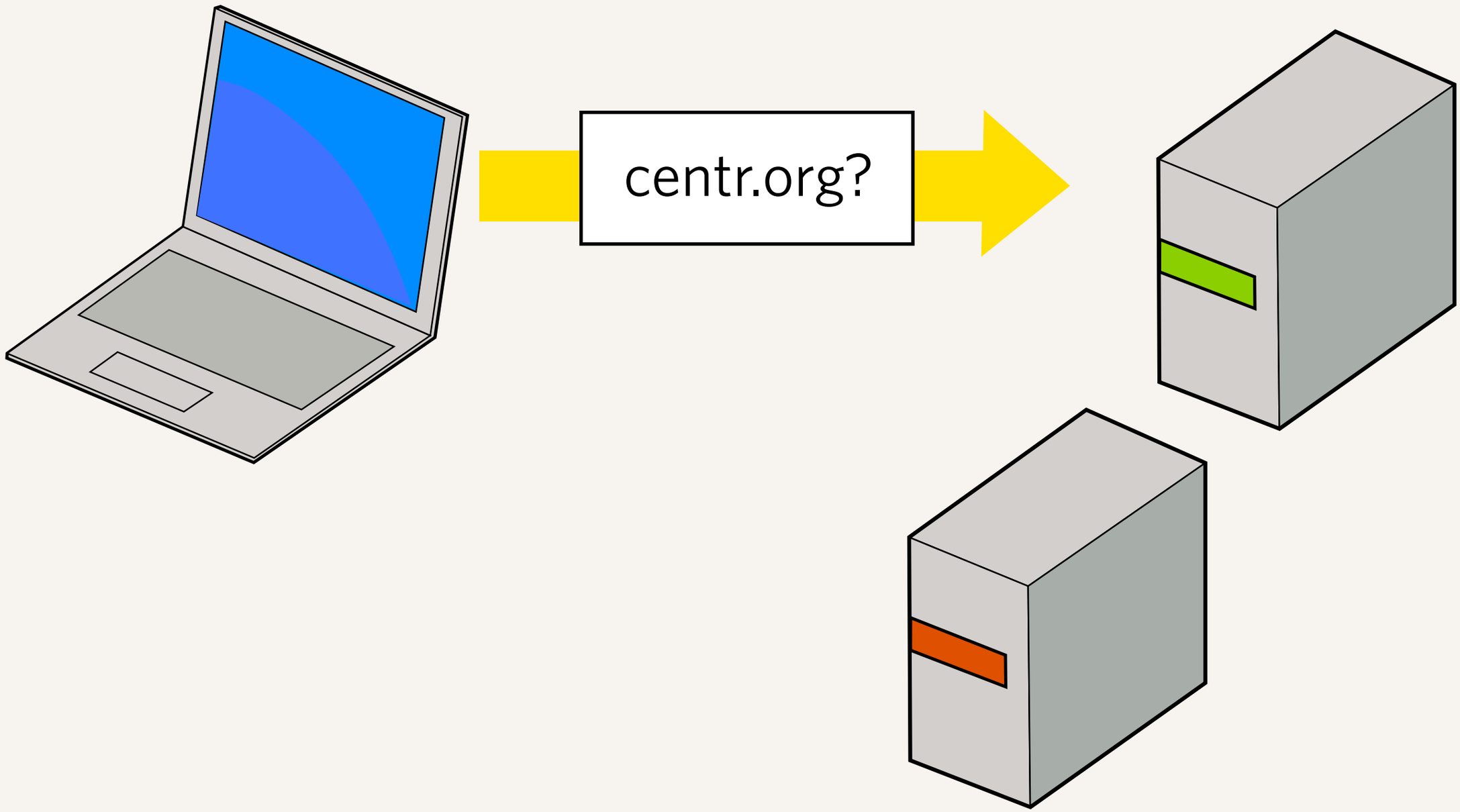
Receiving the wrong answer

- Something in the network between the computer and the server has intercepted or redirected the traffic.



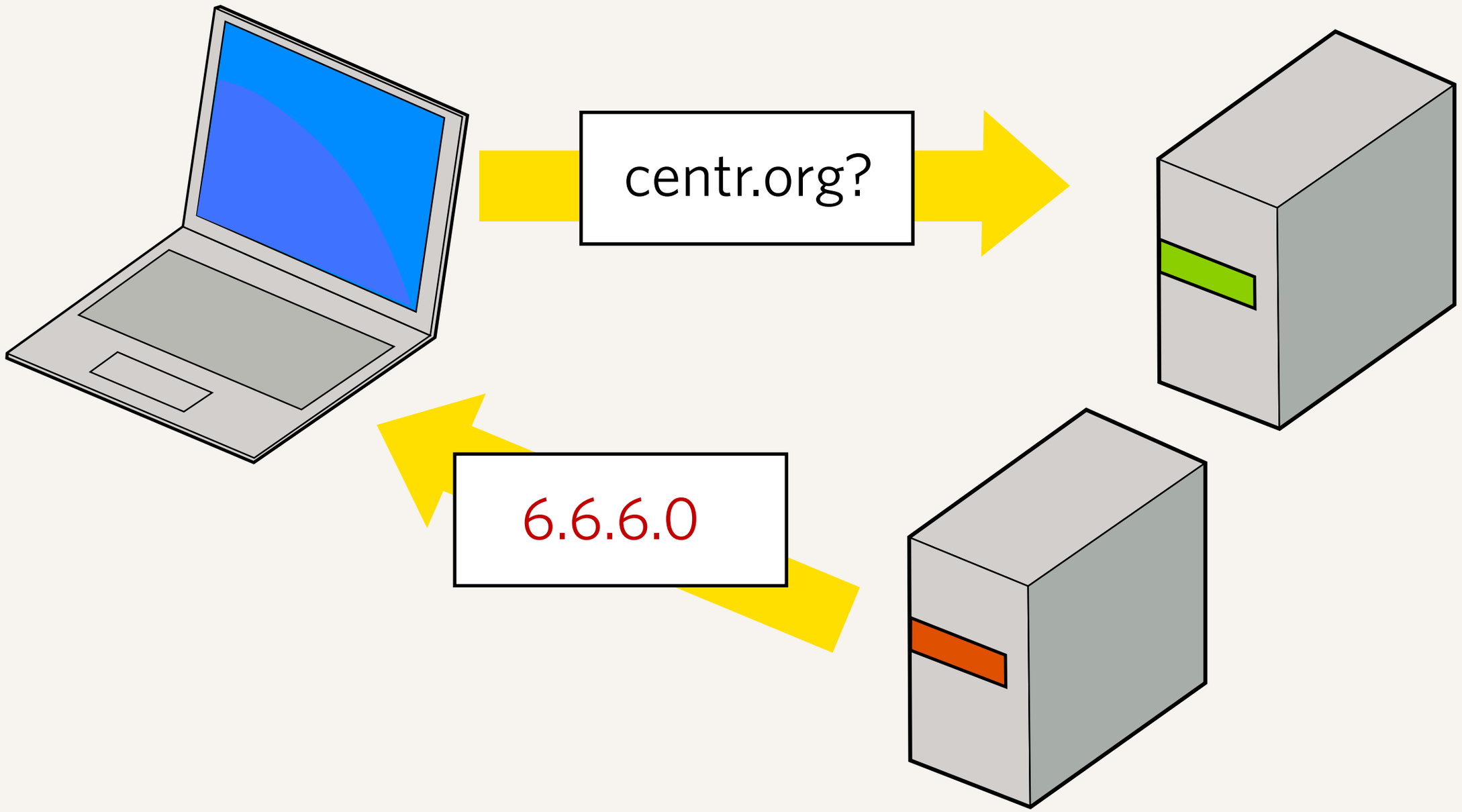
Receiving the wrong answer

- A server on the network responds with the wrong answer, quicker than the correct server can give the right answer.



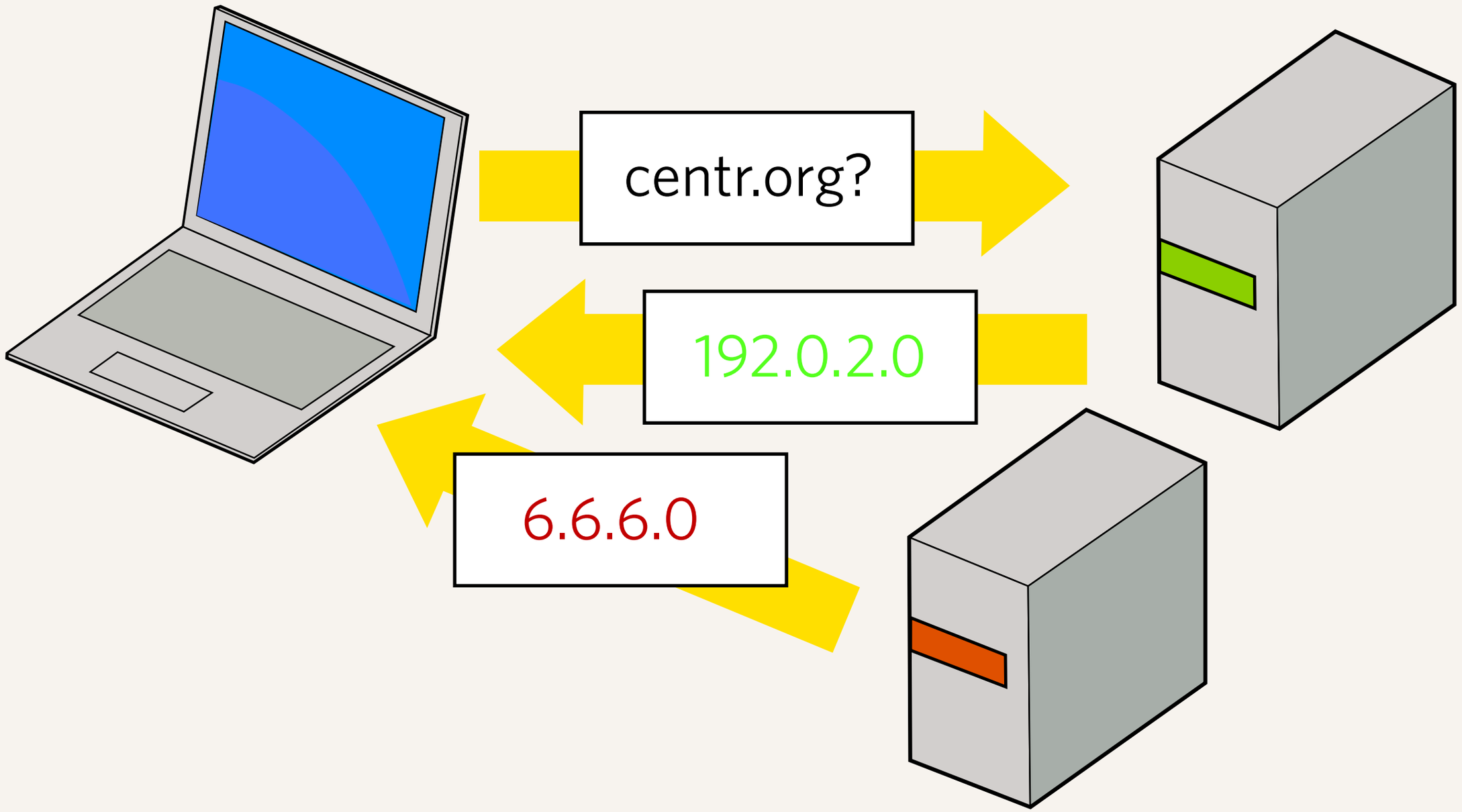
Receiving the wrong answer

- A server on the network responds with the wrong answer, quicker than the correct server can give the right answer.



Receiving the wrong answer

- A server on the network responds with the wrong answer, quicker than the correct server can give the right answer.



Receiving the wrong answer

- A server on the network responds with the wrong answer, quicker than the correct server can give the right answer.

Cache poisoning

- To improve efficiency, DNS servers typically store results in a cache to speed further lookups.
  - This is the typical configuration at ISPs, etc.
- If the wrong answer gets remembered it will be served to future lookups.
  - One successful cache poisoning attack can therefore affect many users.

How does one spoof a response

- A question is sent out, and the querying computer waits for an answer to return
- It knows it has received the answer to its question when several attributes in the answer match the question it asked
  - It comes back to the same IP address it was sent from
  - It comes back to the same port number is was sent from
  - The question matches the question asked
  - A unique transaction number matches what was sent

To spoof a response

- You need to get all these attributes the same in your forged answer packet
  - The IP address needs to match. If you know the IP address of the recursive name server this is known by the attacker, and does not need to be guessed.
  - The question needs to match. The attacker will know what this is, because they will be injecting their own questions into the recursive server.
  - What remains to guess is the transaction number and the port number

But...

But...

- Everything I have told you so far has been known for years.

What has been discovered recently?

This attack is highly effective

- Dan Kaminsky identified there is a straightforward way to flood the recursive server with lots of answers, so that the right combination would be sent very quickly (a few seconds)
- It was also identified that the two identifiers the attacker needs to guess are not fully random (or not random at all)

Why is this attack concerning to TLDs?

- If a name server provides both recursive and authoritative name service, a successful attack on the recursive portion can store bad data that is given to computers that want authoritative answers.
- The net result is one could insert or modify domain data inside a TLD.

Short term solutions

1. Maximise the amount of randomness

- Most implementations use randomised transaction numbers already. (The risk with that was discovered years ago, and fixed in most software)
- Most implementations do NOT randomise the port number. In fact most always used the same port number (53, the port number IANA has assigned for DNS)
- The patches that have been released in the last few months work by randomising the source port for the recursive server.

2. Disable open recusive name servers

- The attack is not effective if the attacker can not send question packets to the name server.
- If you must run a recursive name server, limit access to only those computers that need it. (e.g. your customers). The will still be able to execute the attack, but the exposure is constrained.
- Turning off open recursive name servers is a good idea anyway, because they can be used for other types of attack (denial of service)

Long term solution

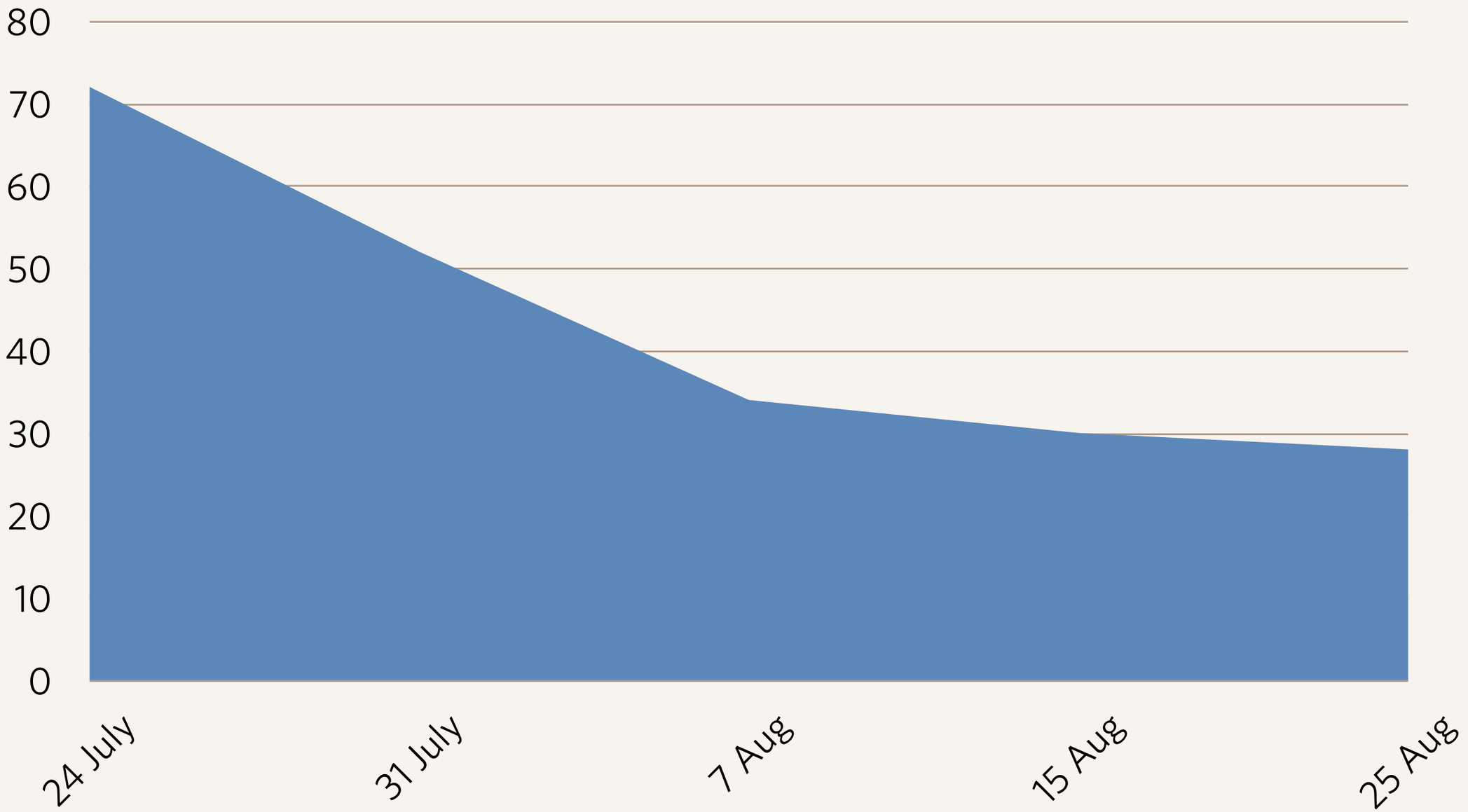
Introduce security to the DNS

- The DNS is insecure. Upgrade the DNS for security.
- DNSSEC is the current answer to this problem.
- This attack provides clear incentive to deploy a solution like DNSSEC, because without security the DNS will continue to be vulnerable to cache poisoning attacks.

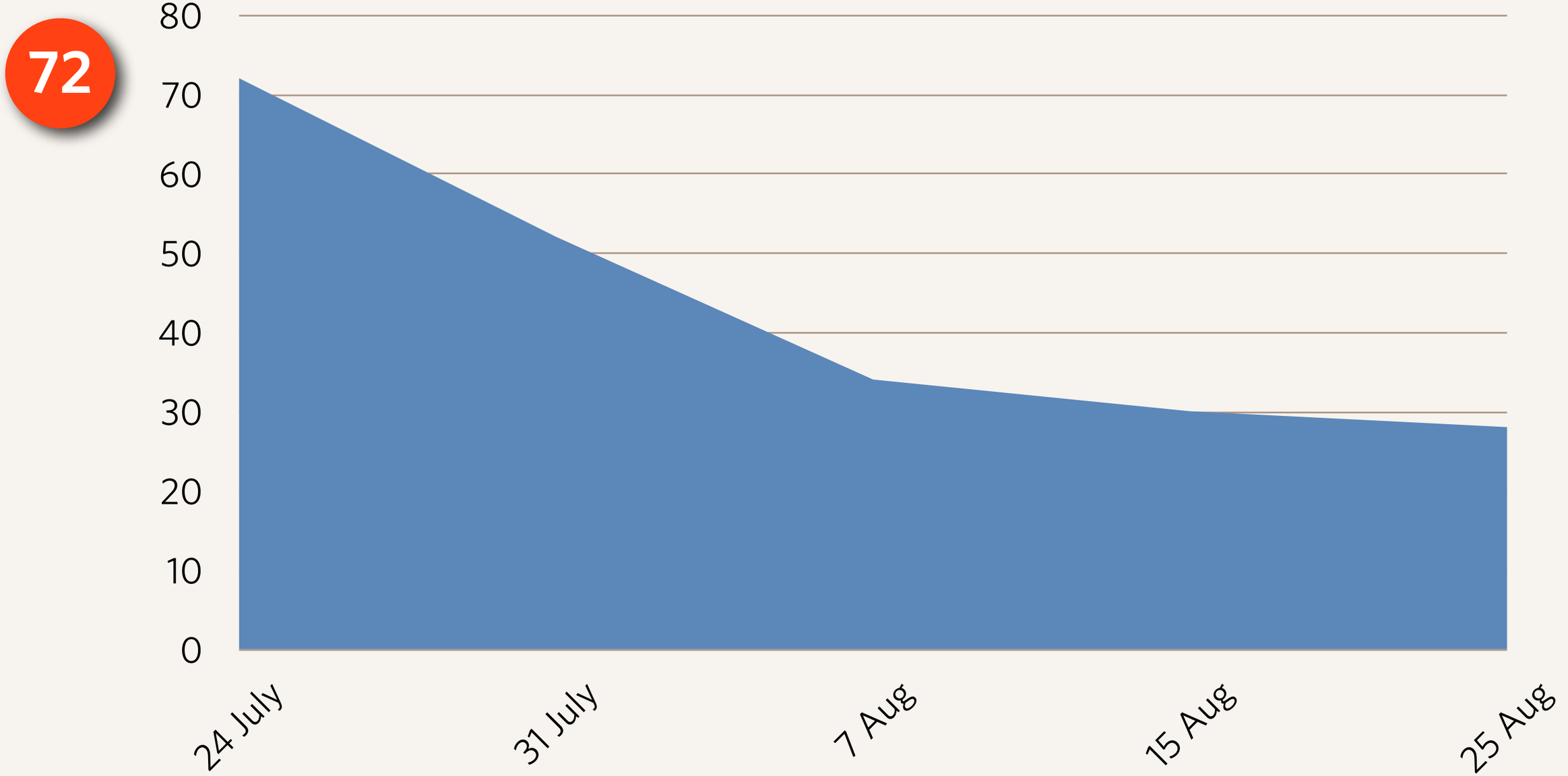
What has ICANN done

Impact on TLDs

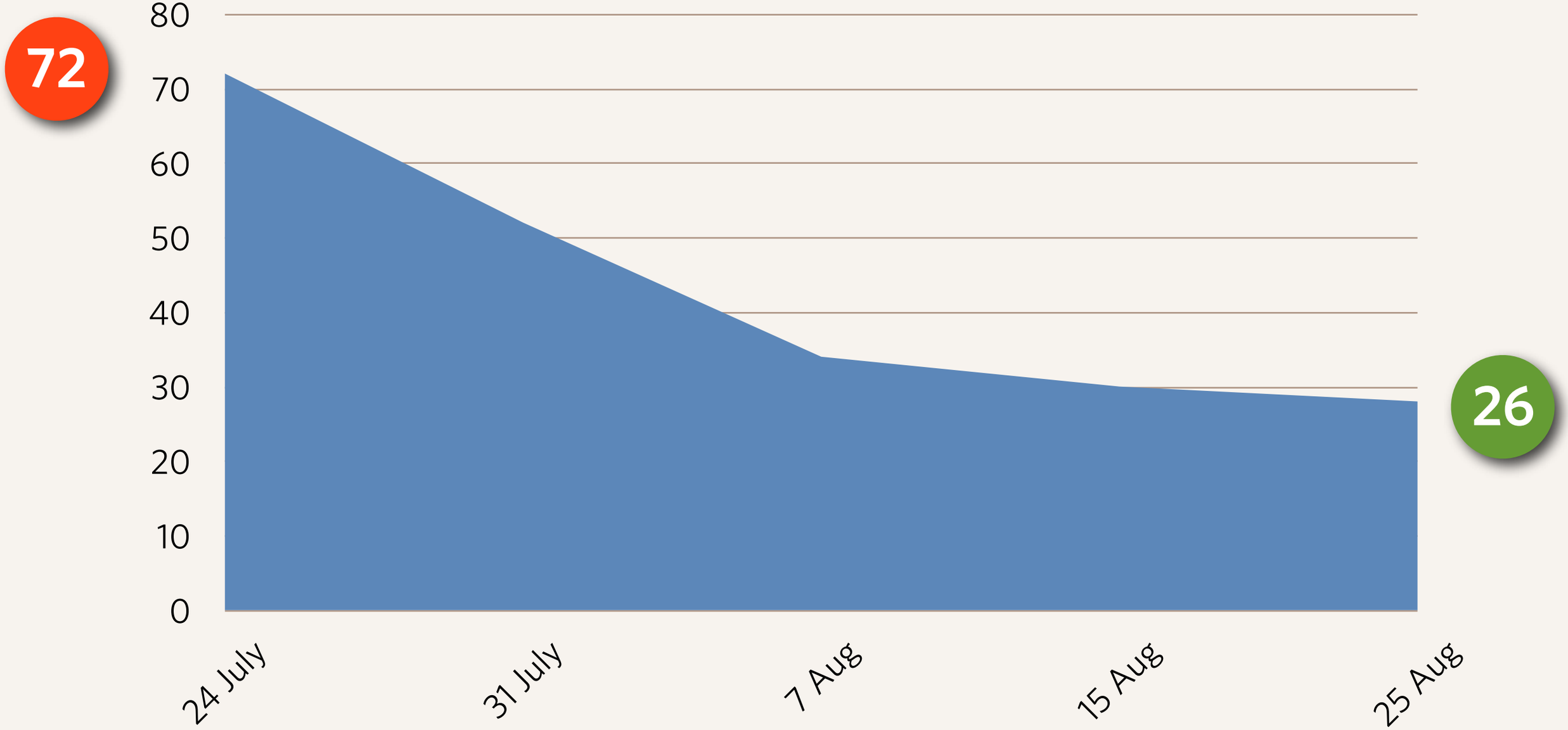
- At the time the vulnerability because known, a survey of TLD operators found that 72 TLDs had authorities that were providing open recursive service.
- ICANN contacted all TLDs affected
  - Explained the situation, and the urgency to fix it
  - Provided advice on how to reconfigure name servers
  - Expedited root zone change requests, if required



TLDs affected



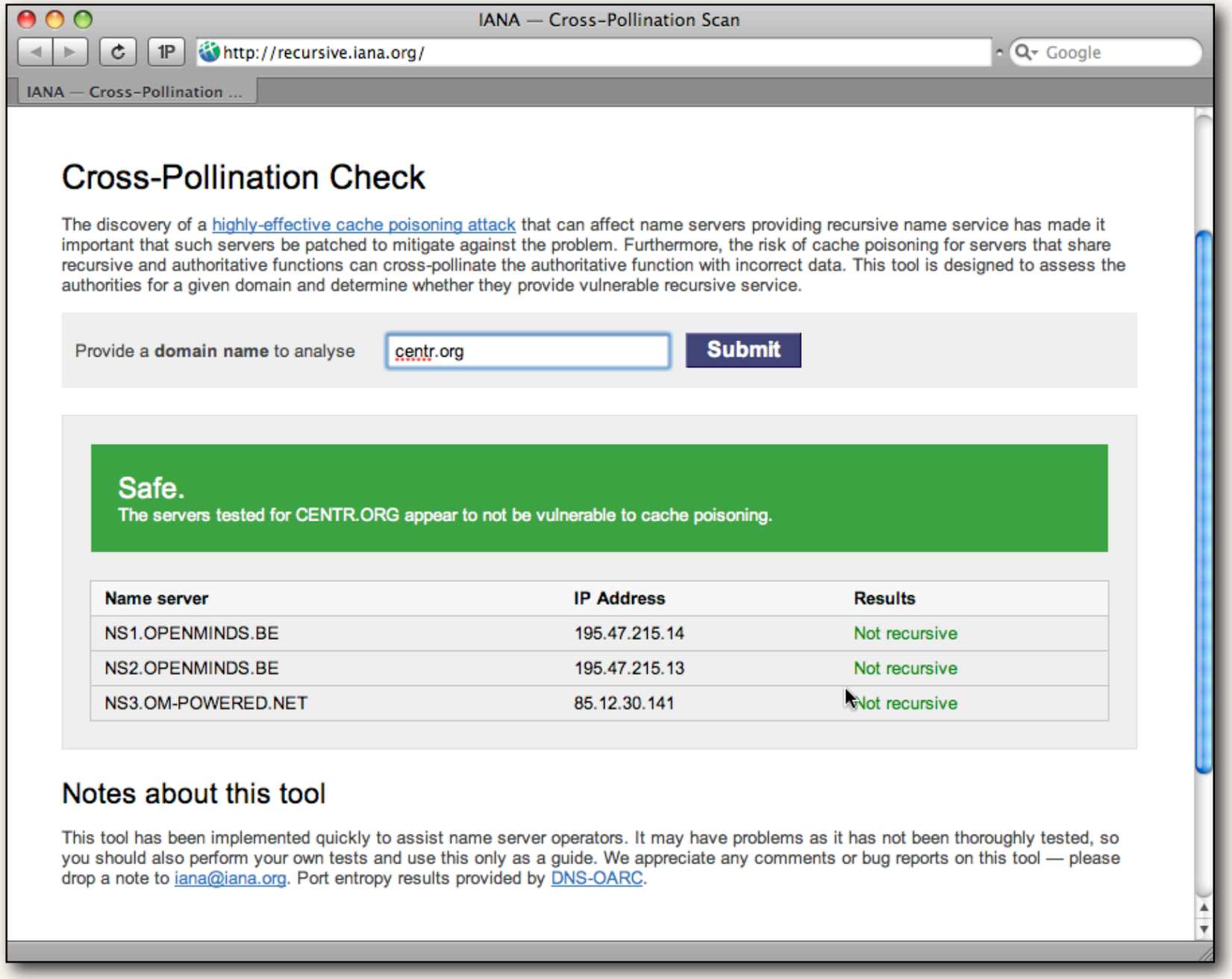
TLDs affected



TLDs affected

Checking tool

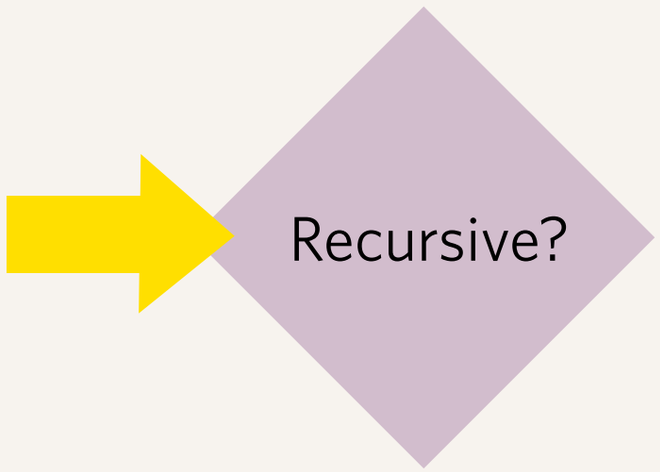
- We developed a tool which we ran daily against TLDs, and shared results with affected TLDs.
- It became clear a web-based tool where TLD operators could self-test would be useful, so it was reimplemented this way.
- The tool is not TLD specific, and works with any domain name.
- It is at http://recursive.iana.org/



Vulnerability checking tool

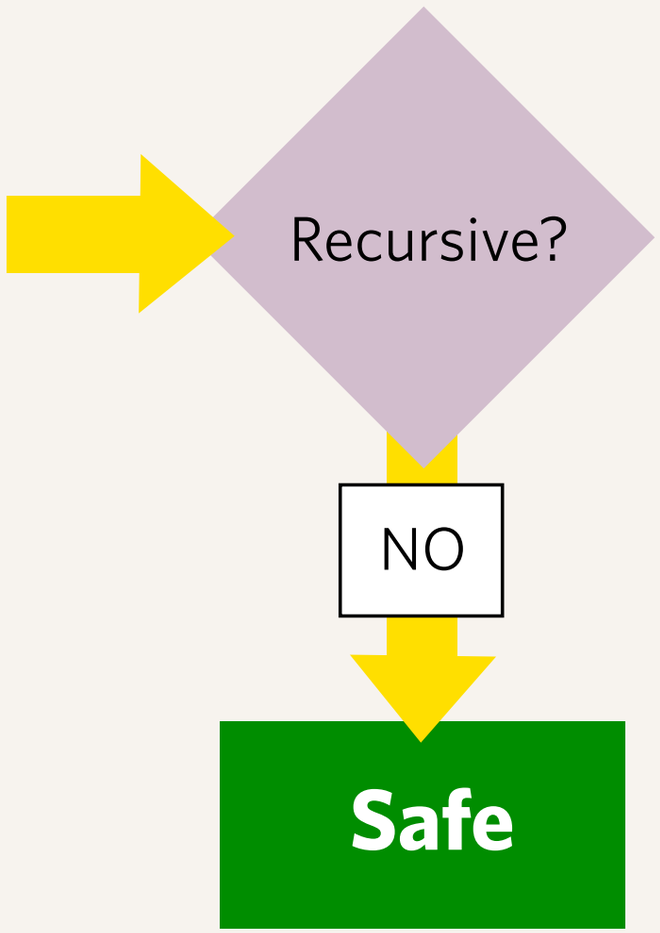
How the tool works

- The tool checks for the two aspects that enable the attack



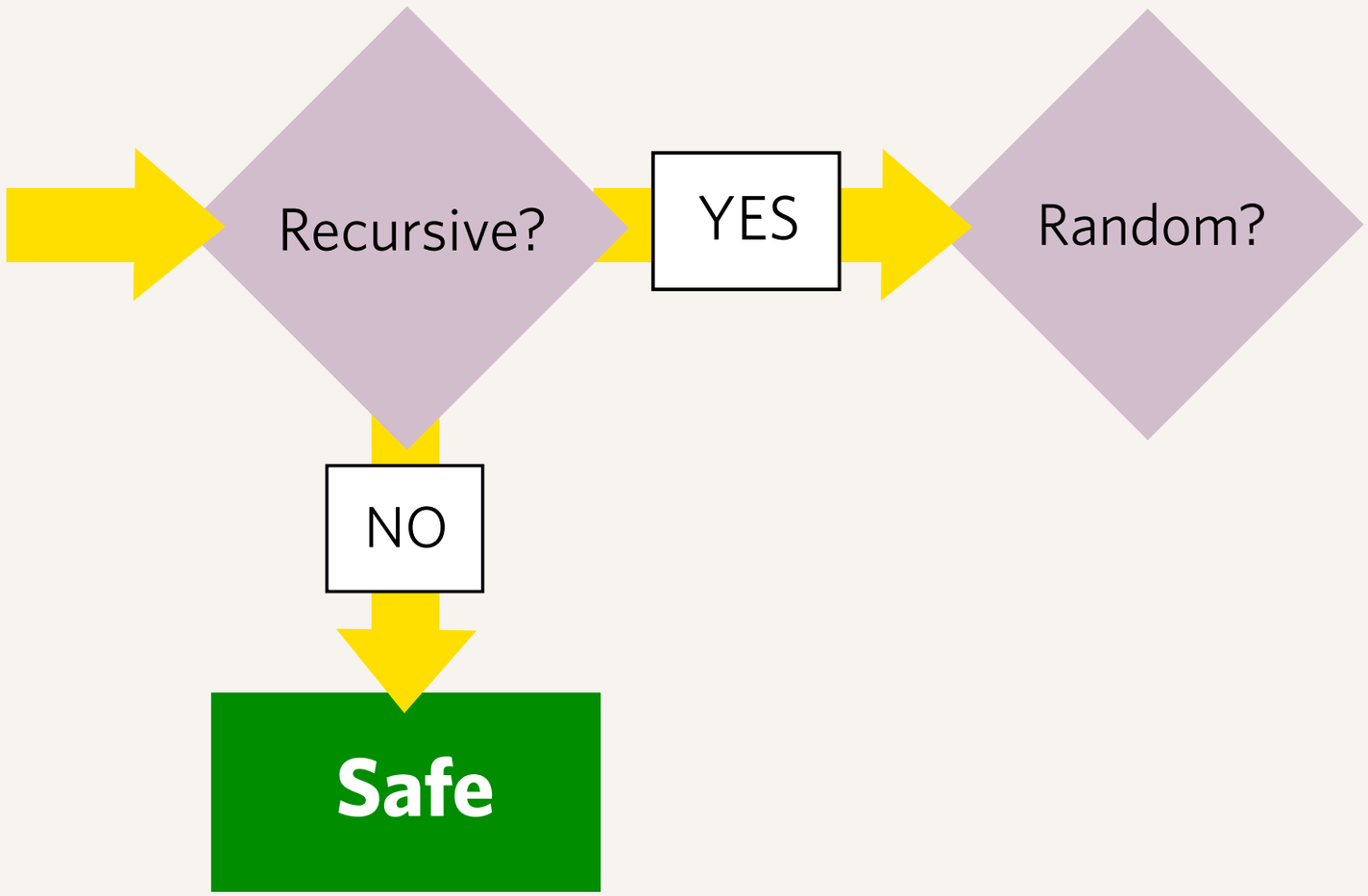
How the tool works

- The tool checks for the two aspects that enable the attack



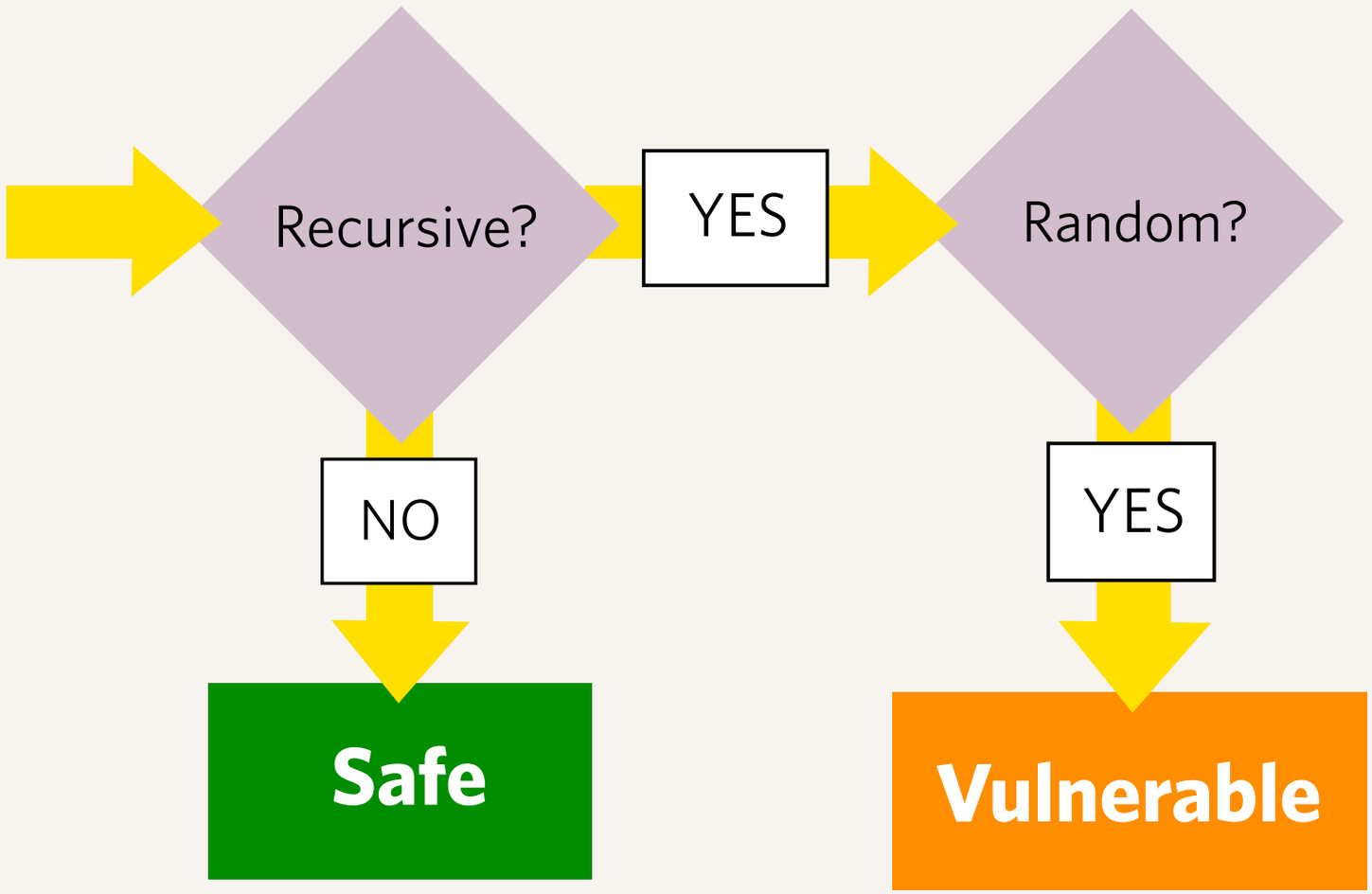
How the tool works

- The tool checks for the two aspects that enable the attack



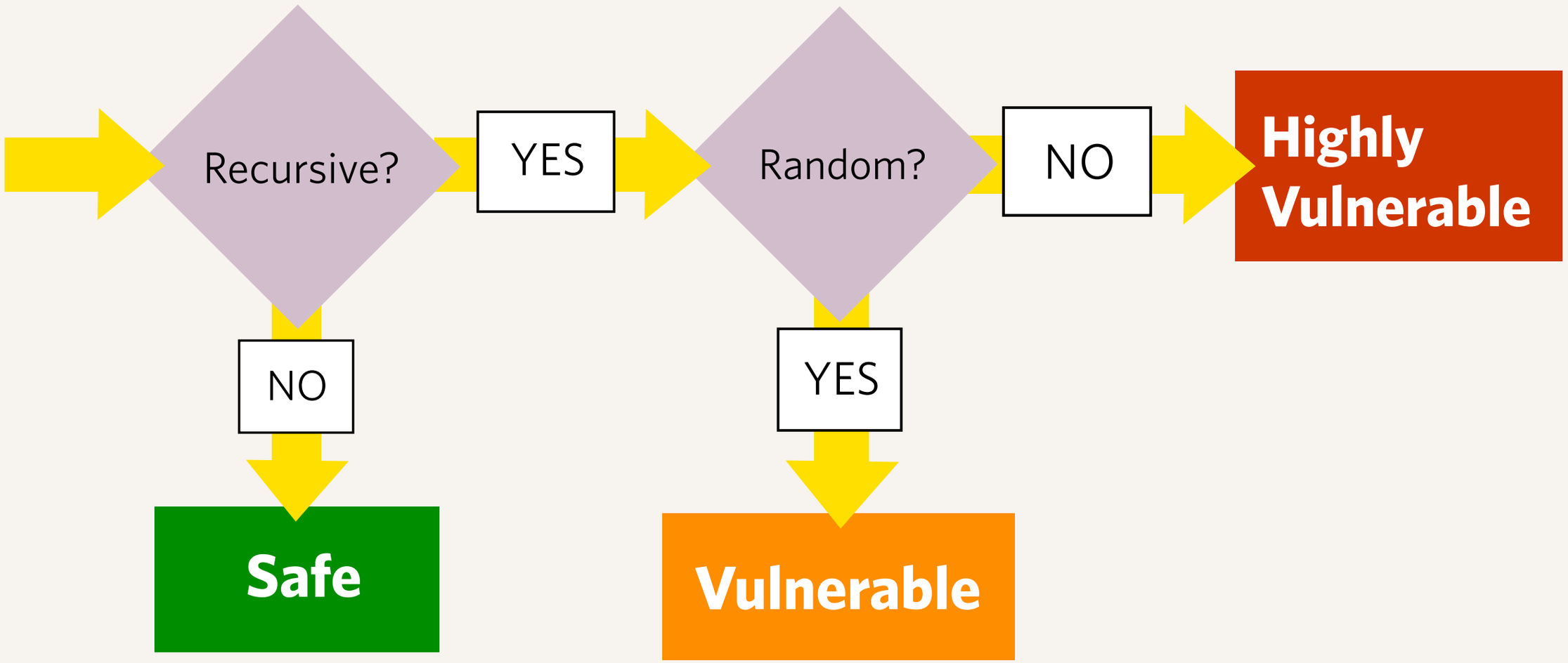
How the tool works

- The tool checks for the two aspects that enable the attack



How the tool works

- The tool checks for the two aspects that enable the attack



How the tool works

- The tool checks for the two aspects that enable the attack

over **100,000** domains tested

Work continues

- We are still working with the last remaining TLDs that are affected. Our goal is to reduce the number to zero.
- It is anticipated a ban on open recursive name servers will be instituted as a formal IANA requirement on future root zone changes.
- Work on DNSSEC, and signing the root, to facilitate a longer term solution

Thanks!

kim.davies@icann.org