2008 DNS Cache Poisoning Vulnerability

Cairo, Egypt
November 2008

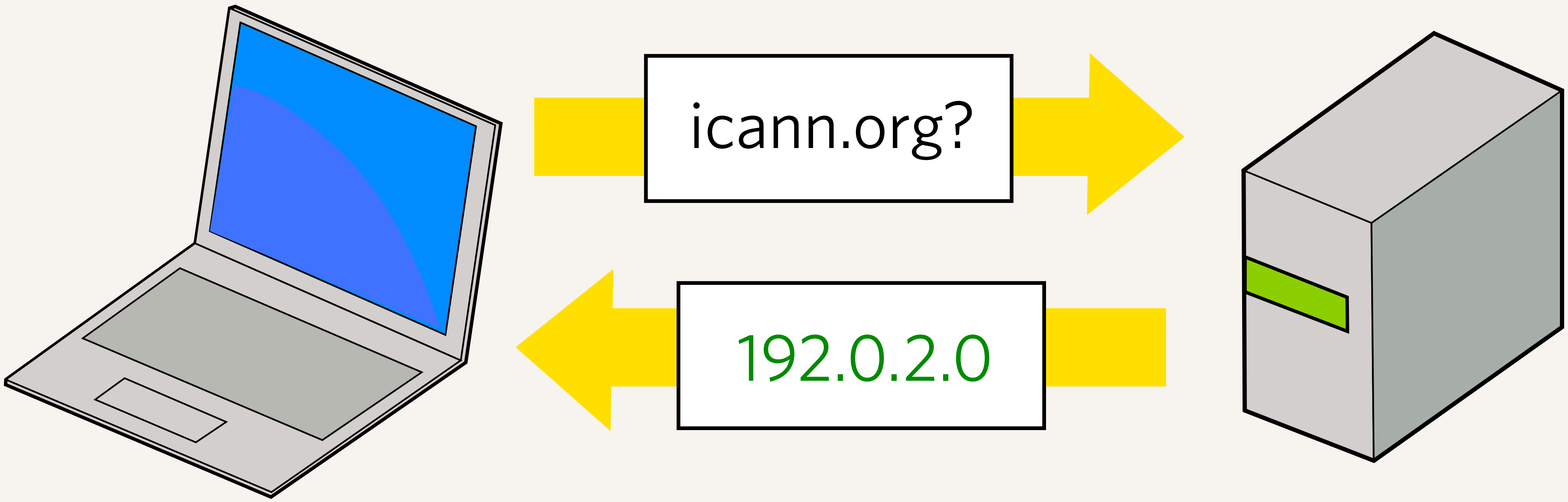
Kim Davies
Manager, Root Zone Services

ICANN
Internet Corporation for
Assigned Names & Numbers

How does the DNS work?

A typical DNS query

The DNS protocol revolves around sending questions, and sending back answers to those questions.



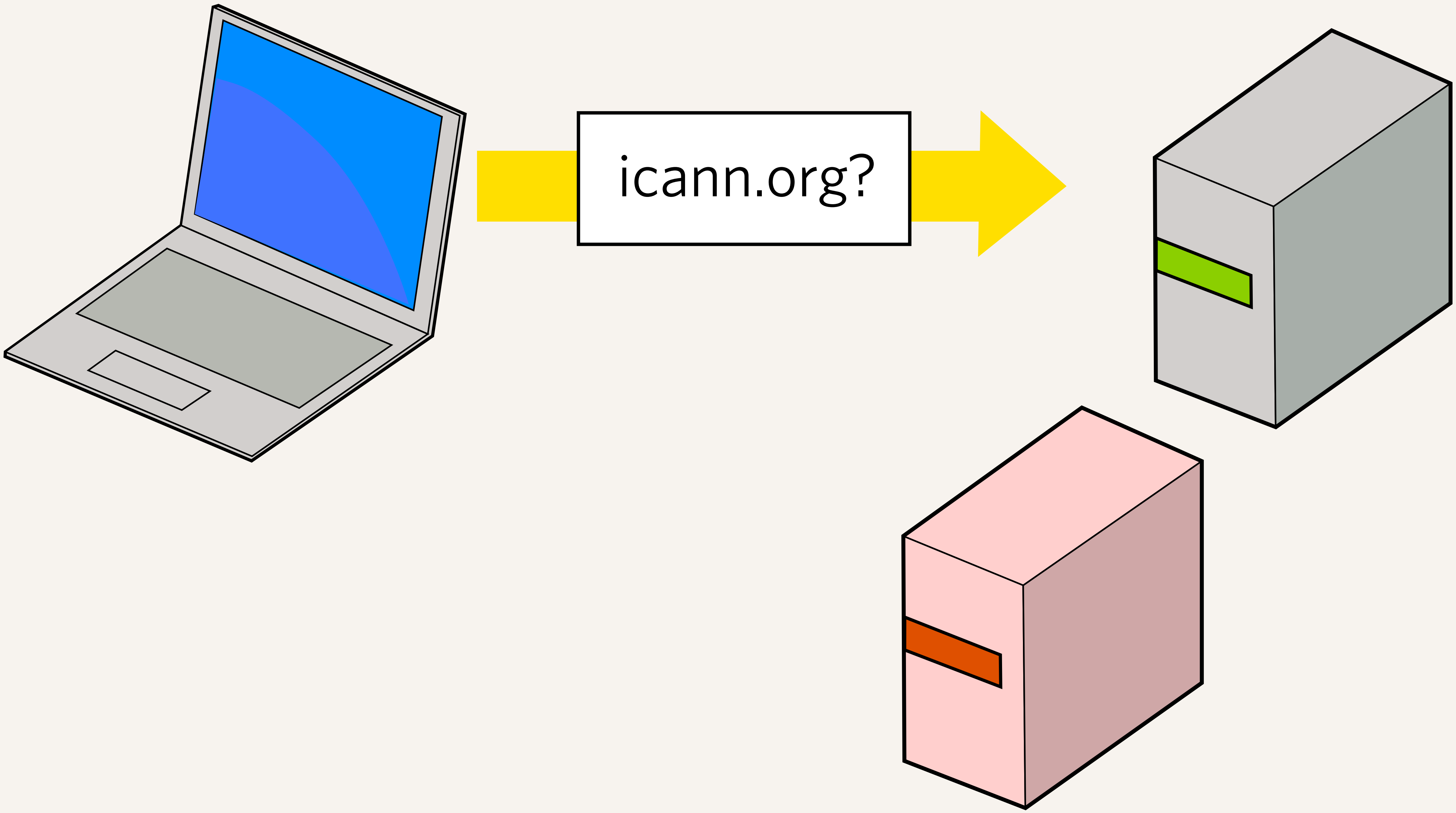
A typical DNS query

The DNS protocol revolves around sending questions, and sending back answers to those questions.

How do you attack the DNS?

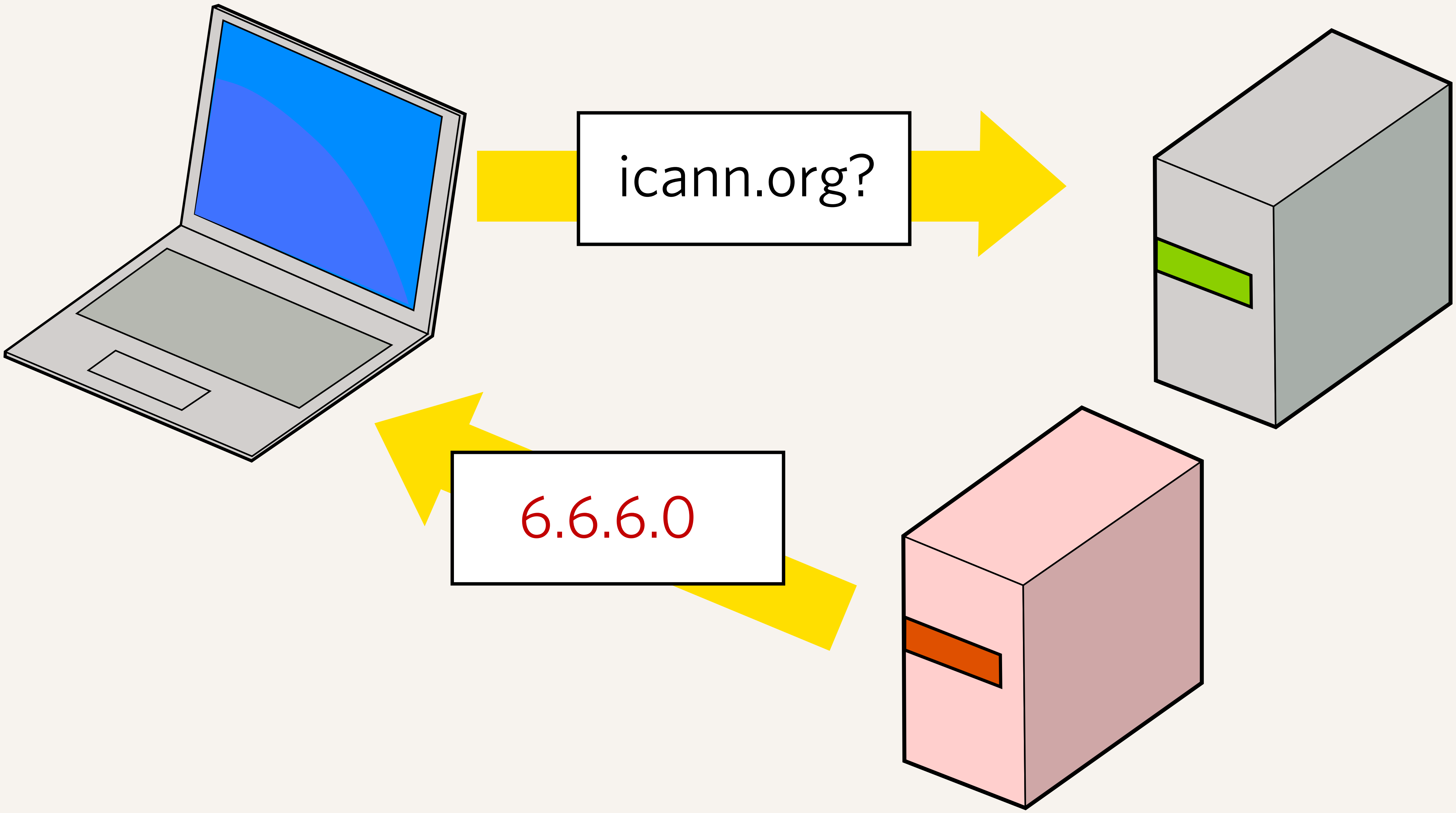
The DNS is not secure

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



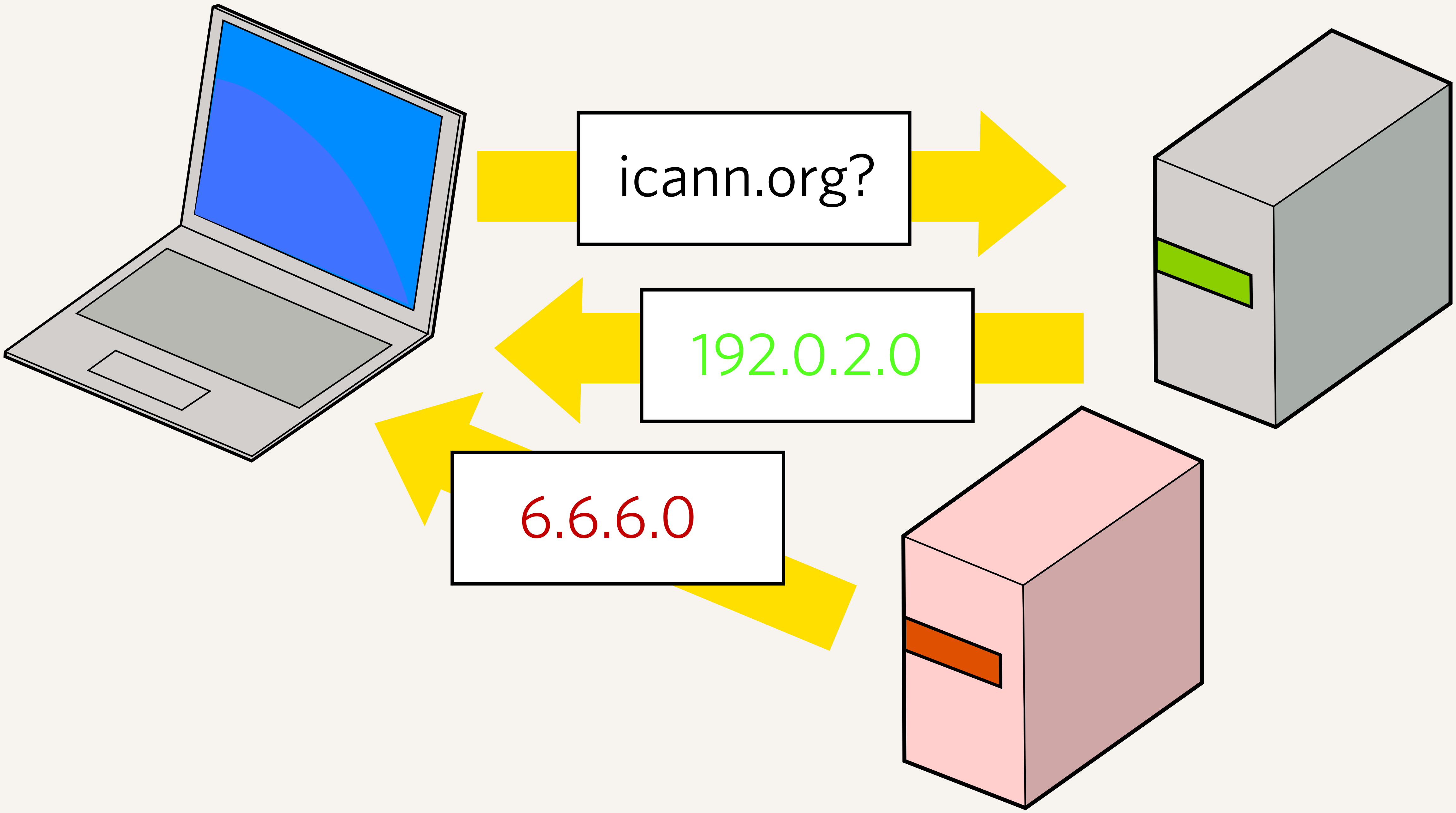
Winning the race

Exploits rely on the server providing the false answer responding quicker than the correct server can give the right answer.



Winning the race

Exploits rely on the server providing the false answer responding quicker than the correct server can give the right answer.

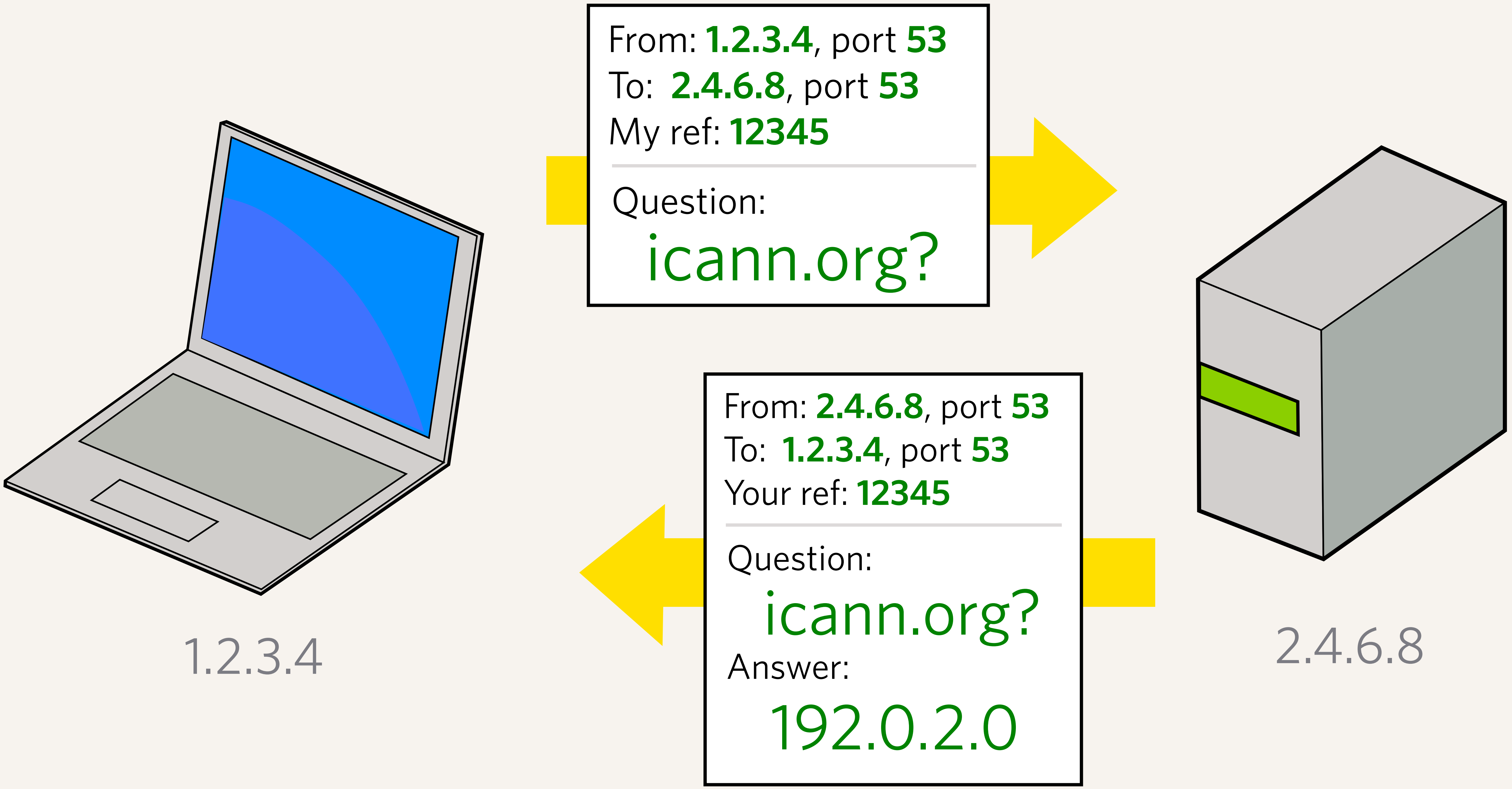


Winning the race

Exploits rely on the server providing the false answer responding quicker than the correct server can give the right answer.

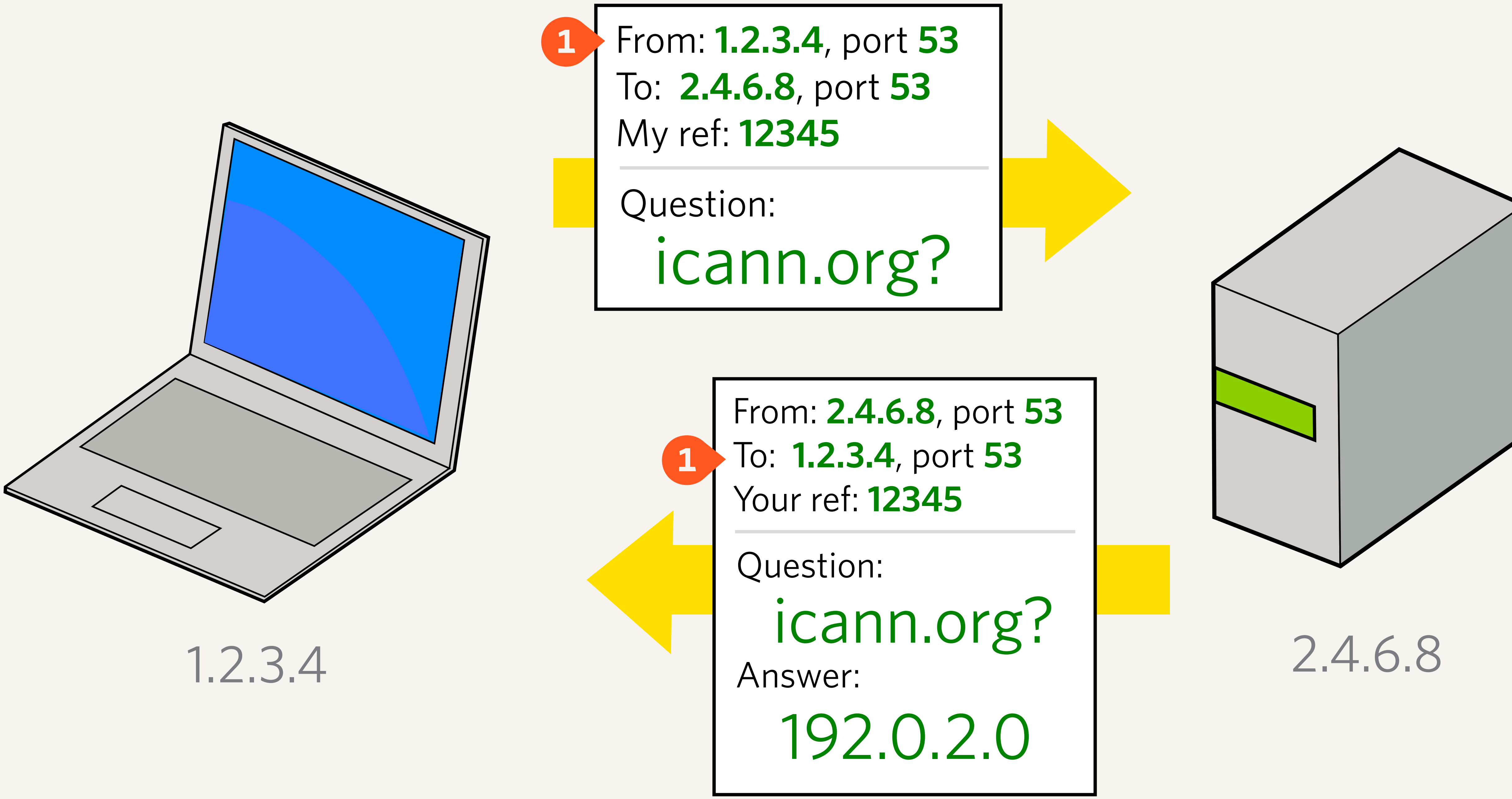
Cache poisoning

- The previous example scenario is a successful attack against just one computer.
- To improve efficiency, intermediate DNS servers typically store results in a cache to speed further lookups.
  - This is the typical configuration at ISPs, etc.
- If an attacker can trick a server to remember a wrong answer, the server will then use it to respond to future lookups.
  - One successful attack can therefore affect many users by “poisoning” the cache.



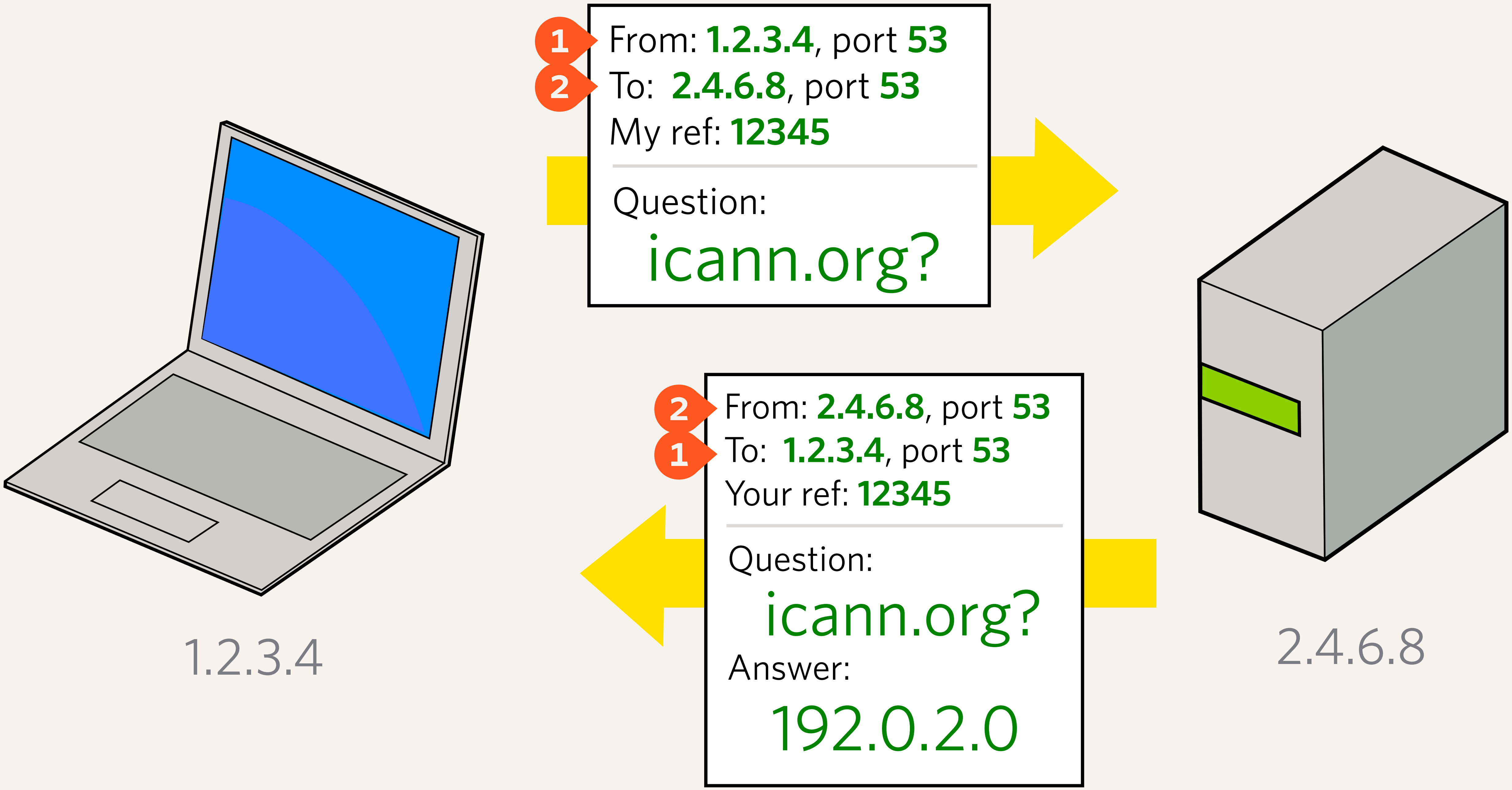
What should match in a DNS transaction

① Source address and port ② Destination address and port
③ Reference (Transaction) number ④ Question being asked



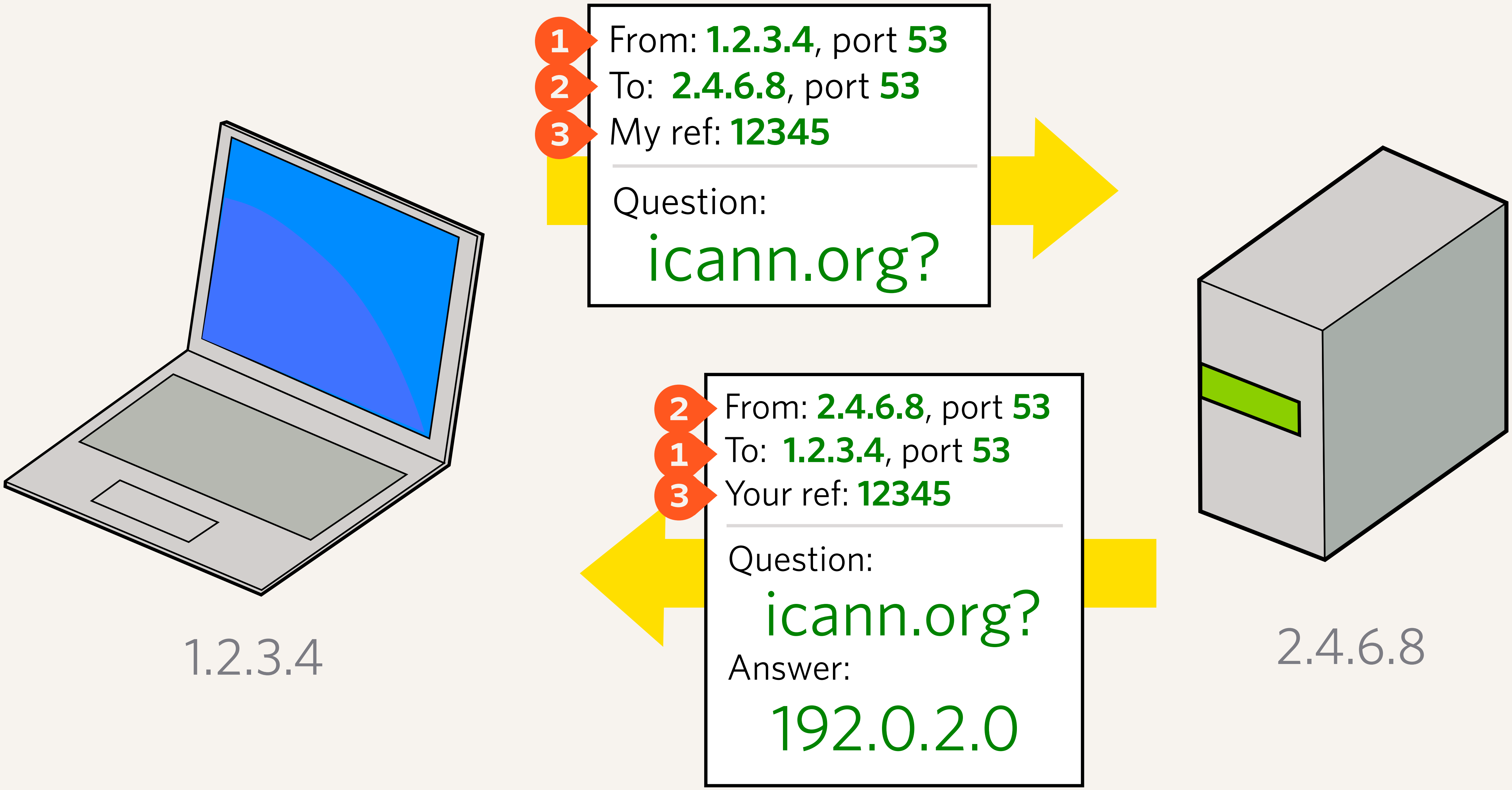
What should match in a DNS transaction

① Source address and port ② Destination address and port
③ Reference (Transaction) number ④ Question being asked



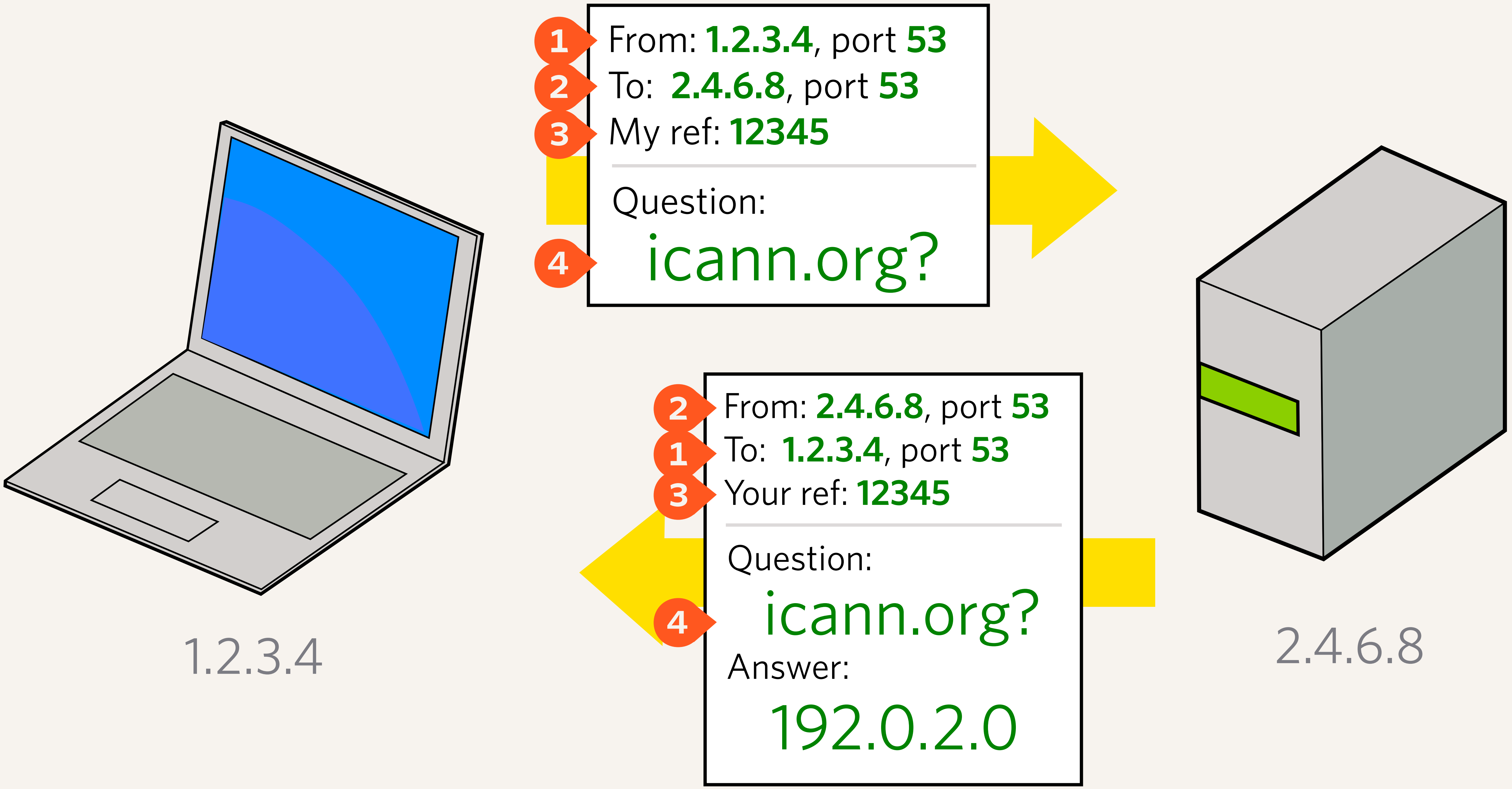
What should match in a DNS transaction

① Source address and port ② Destination address and port
③ Reference (Transaction) number ④ Question being asked



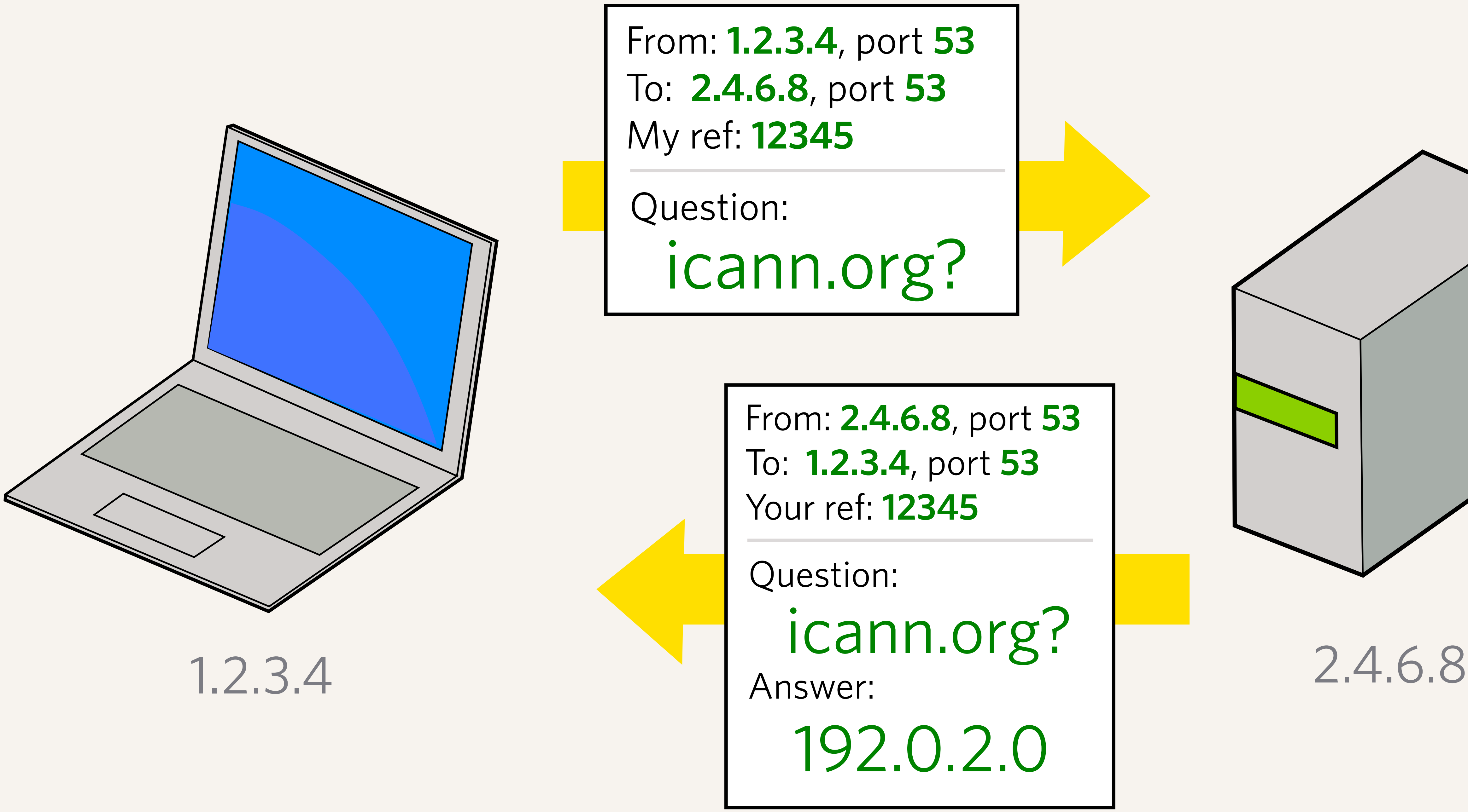
What should match in a DNS transaction

① Source address and port ② Destination address and port
③ Reference (Transaction) number ④ Question being asked



What should match in a DNS transaction

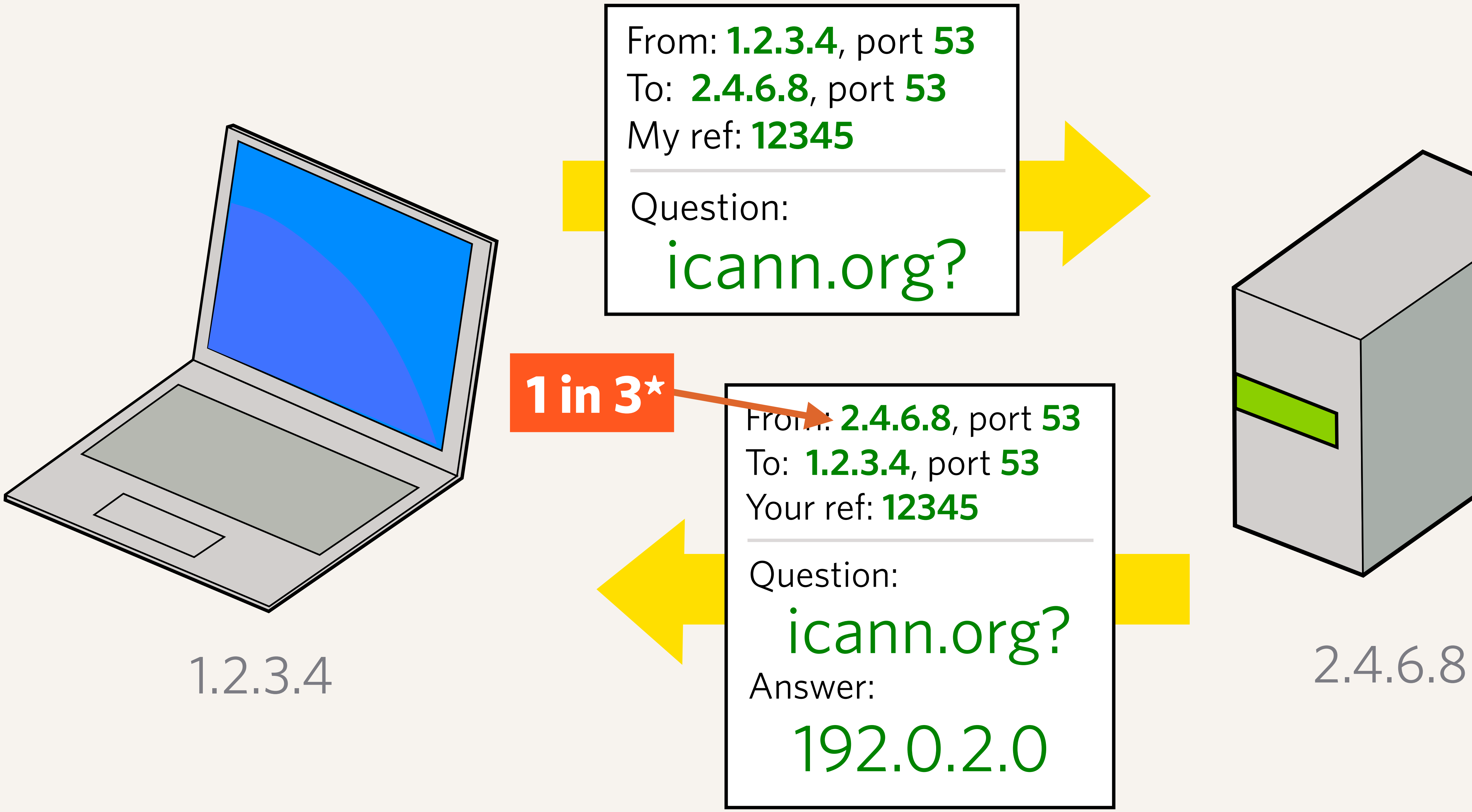
① Source address and port ② Destination address and port
③ Reference (Transaction) number ④ Question being asked



Approximate possible combinations

The key variability is in the reference number. Other values are mostly deterministic.

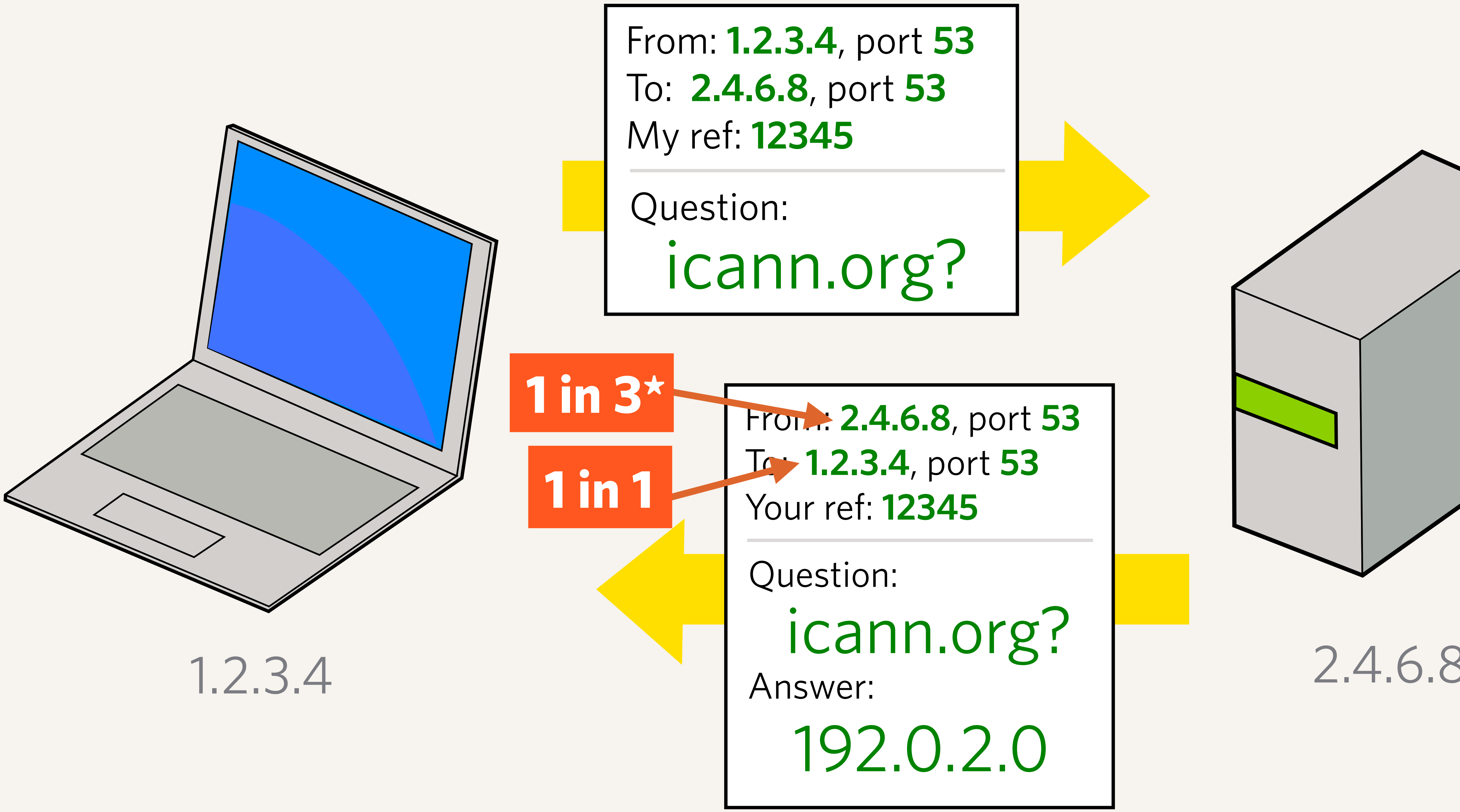
* Number of authoritative name servers for the domain (average is 2.5)



Approximate possible combinations

The key variability is in the reference number. Other values are mostly deterministic.

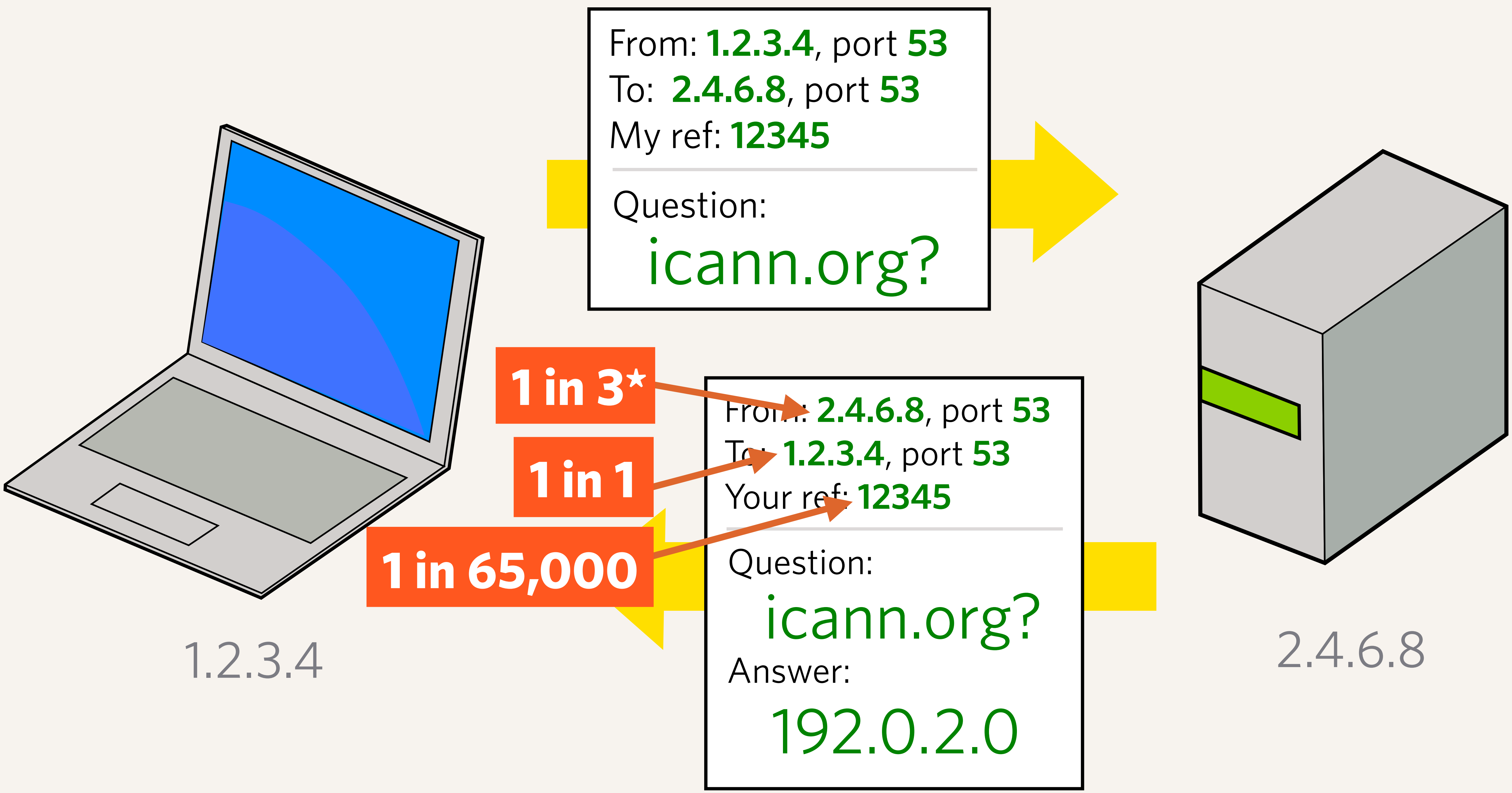
* Number of authoritative name servers for the domain (average is 2.5)



Approximate possible combinations

The key variability is in the reference number. Other values are mostly deterministic.

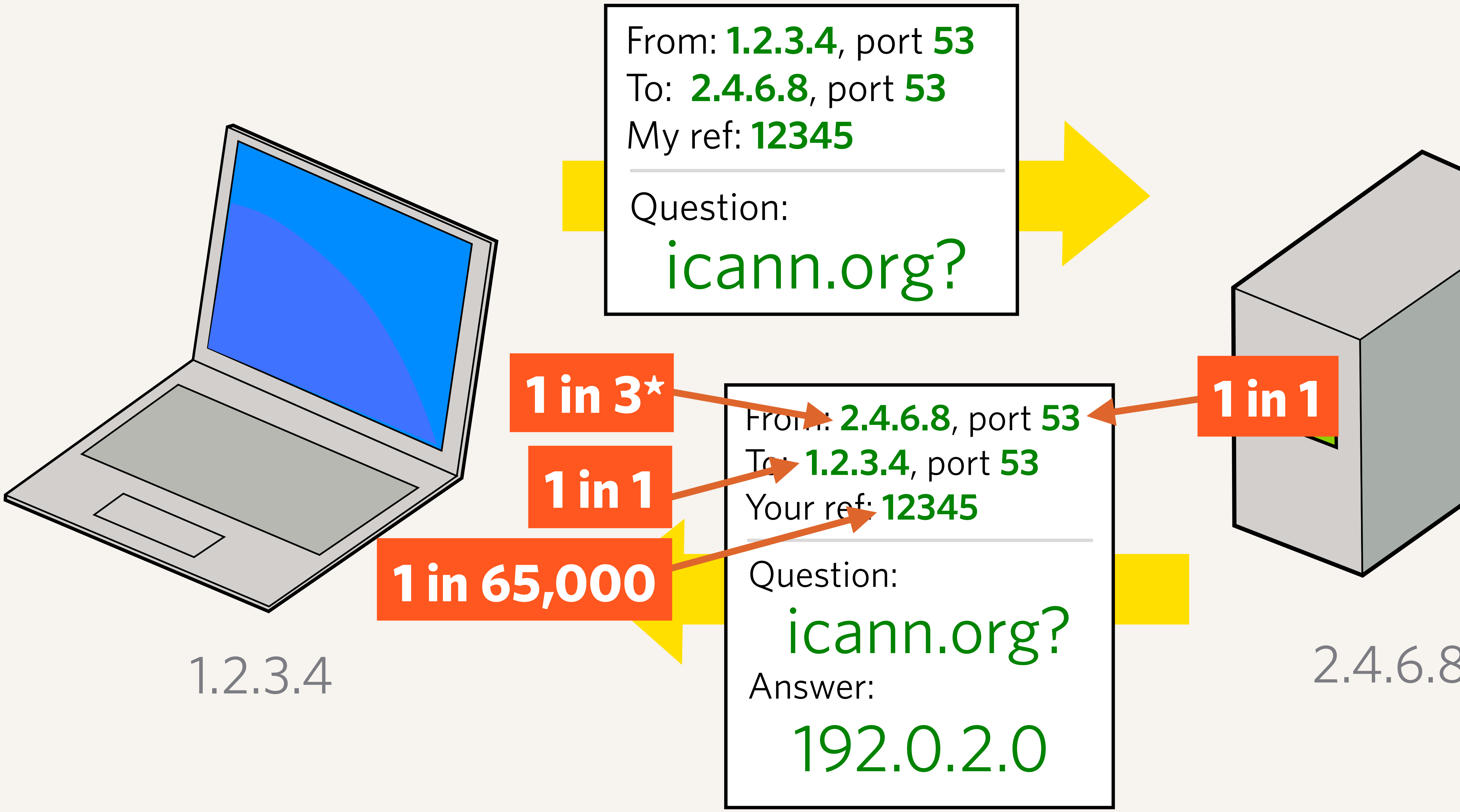
* Number of authoritative name servers for the domain (average is 2.5)



Approximate possible combinations

The key variability is in the reference number. Other values are mostly deterministic.

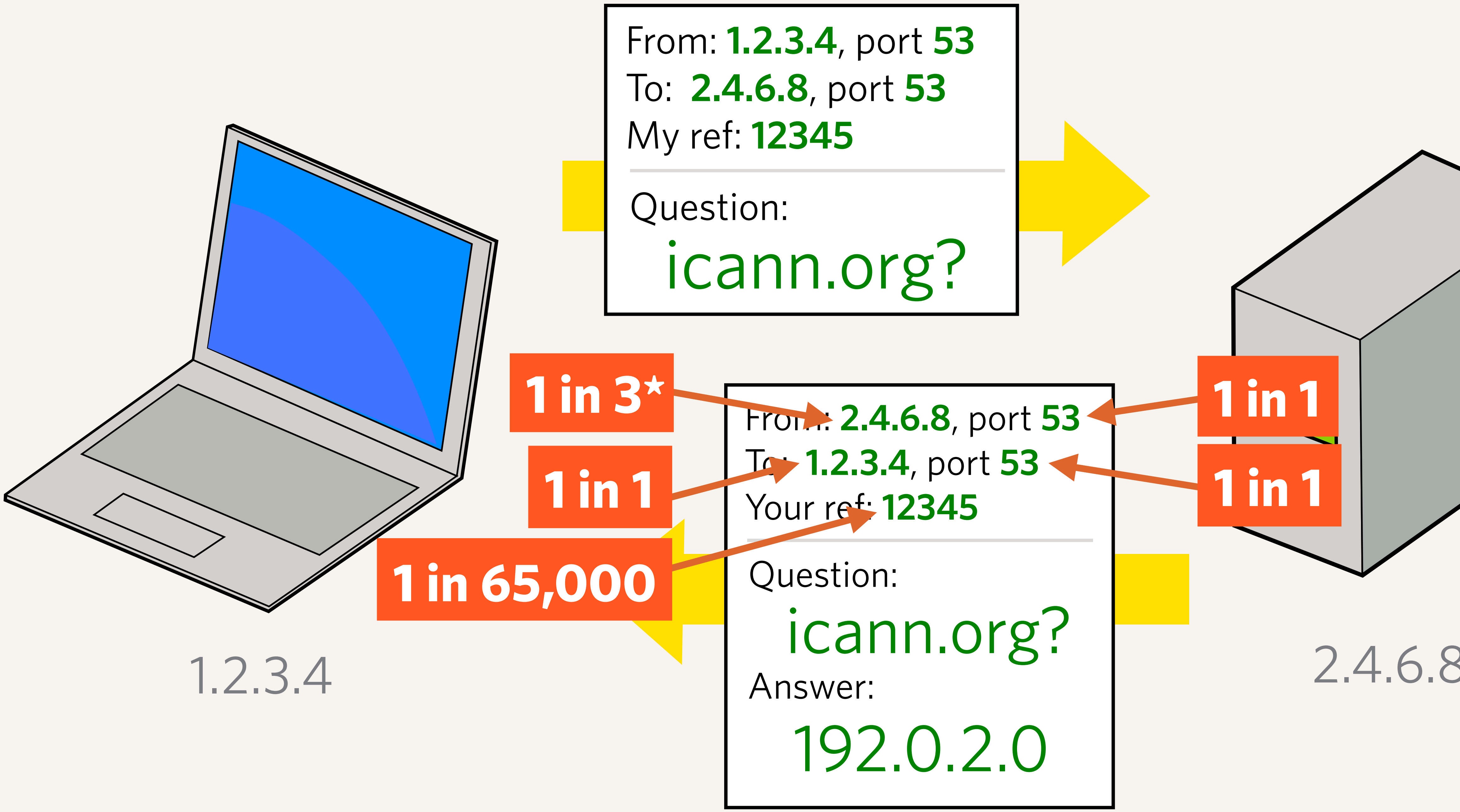
* Number of authoritative name servers for the domain (average is 2.5)



Approximate possible combinations

The key variability is in the reference number. Other values are mostly deterministic.

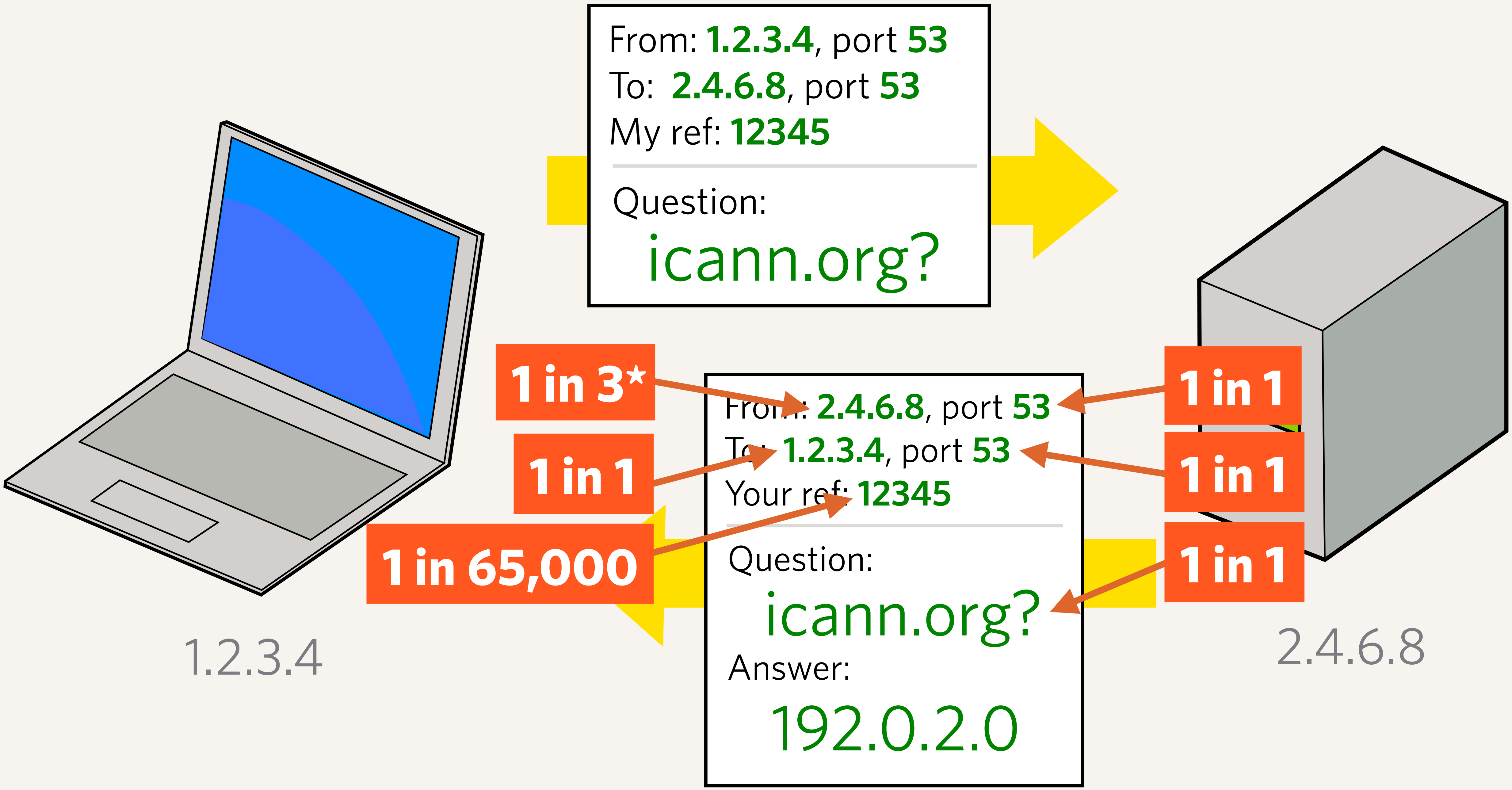
* Number of authoritative name servers for the domain (average is 2.5)



Approximate possible combinations

The key variability is in the reference number. Other values are mostly deterministic.

* Number of authoritative name servers for the domain (average is 2.5)



Approximate possible combinations

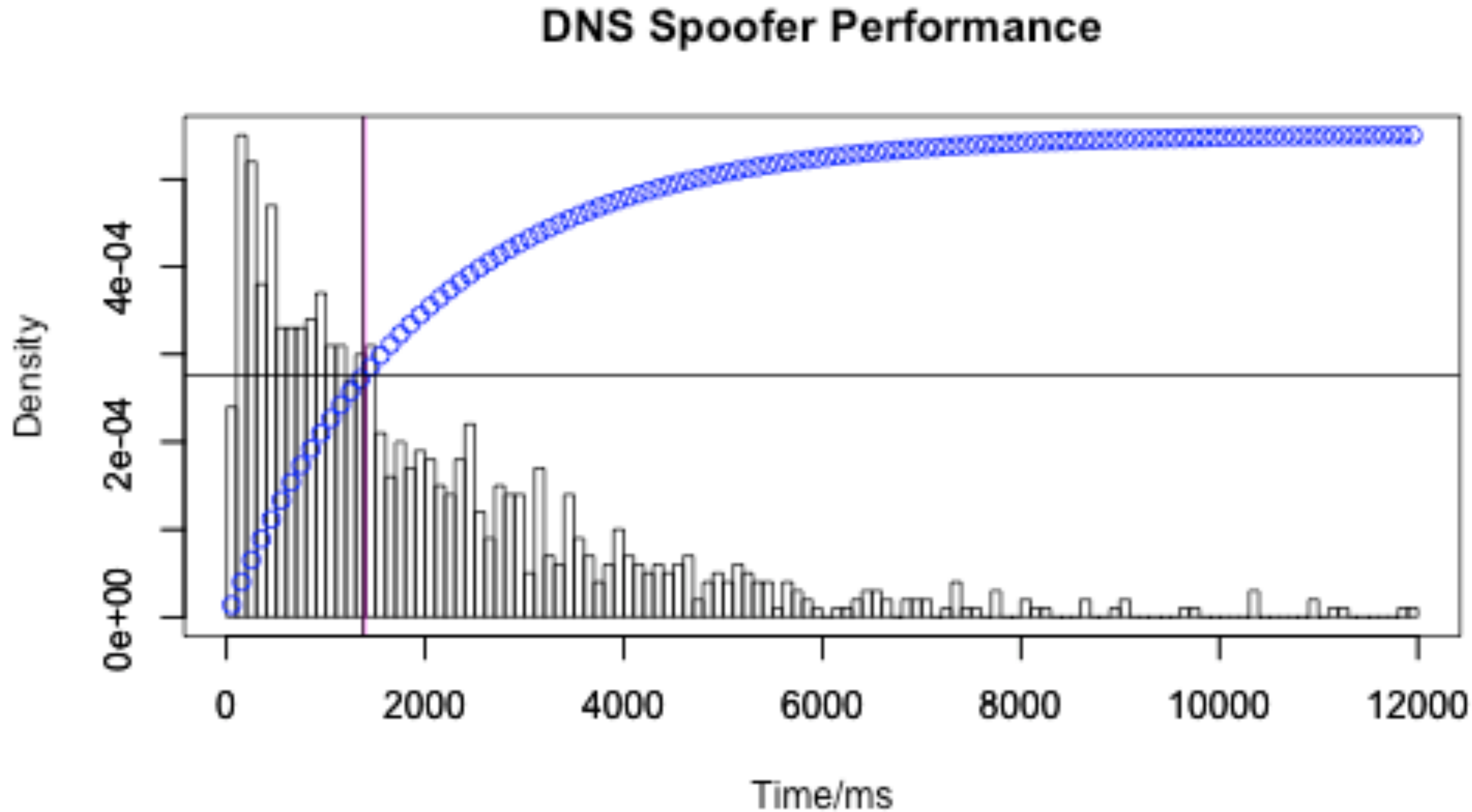
The key variability is in the reference number. Other values are mostly deterministic.

* Number of authoritative name servers for the domain (average is 2.5)

What has been discovered recently?

This attack is highly effective

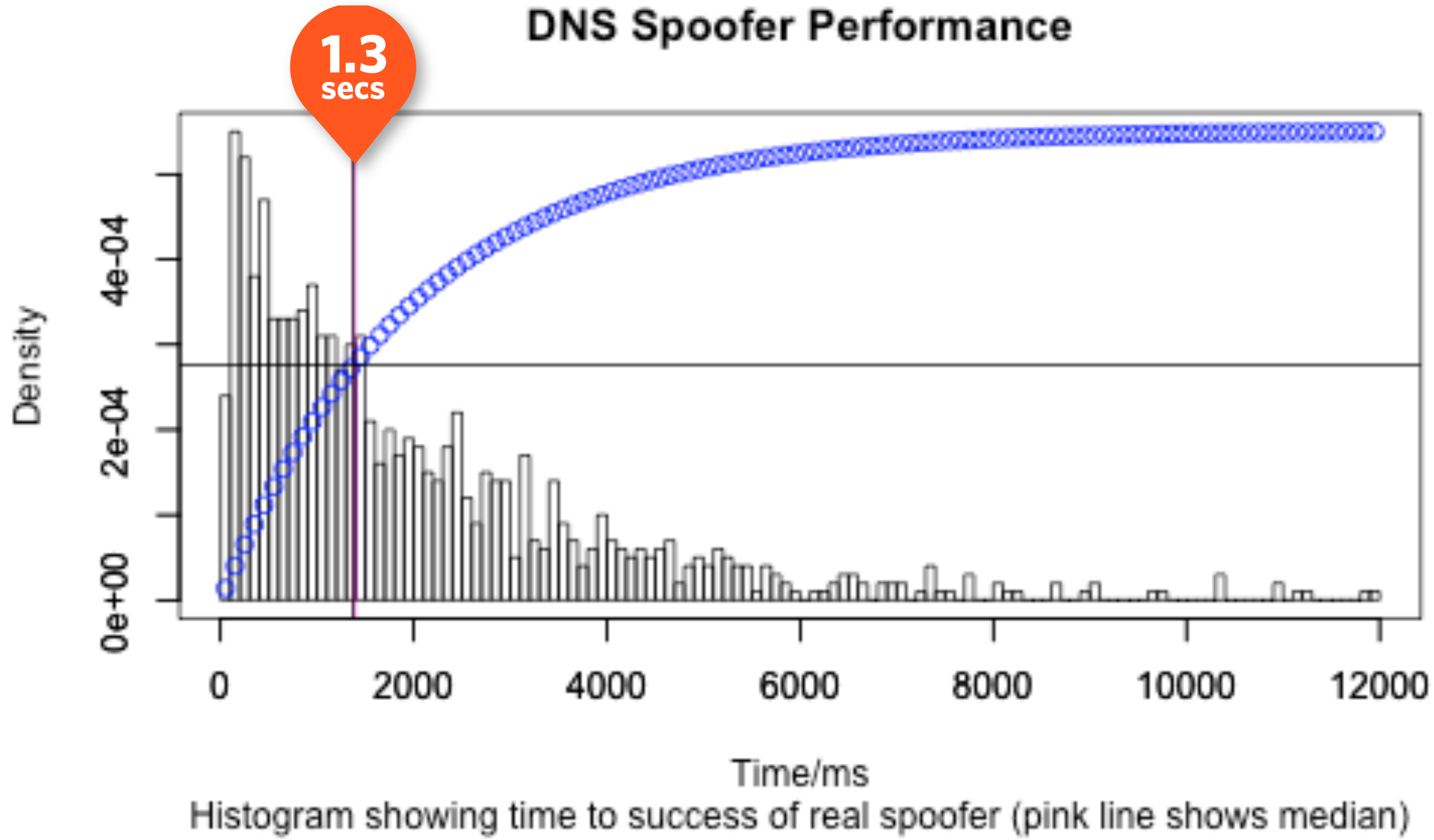
- Dan Kaminsky identified there is a straightforward way to flood an attack target with lots of answers, so that the right combination would be found very quickly (a few seconds)
- By querying for random hosts within a domain (`0001.targetdomain.com`, `0002.targetdomain.com`, etc.), you can take over the target domain by filling the cache with bad referral information.



Histogram showing time to success of real spoofer (pink line shows median)

How effective?

Courtesy John Dickinson (jadickinson.co.uk)



How effective?

Courtesy John Dickinson (jadickinson.co.uk)

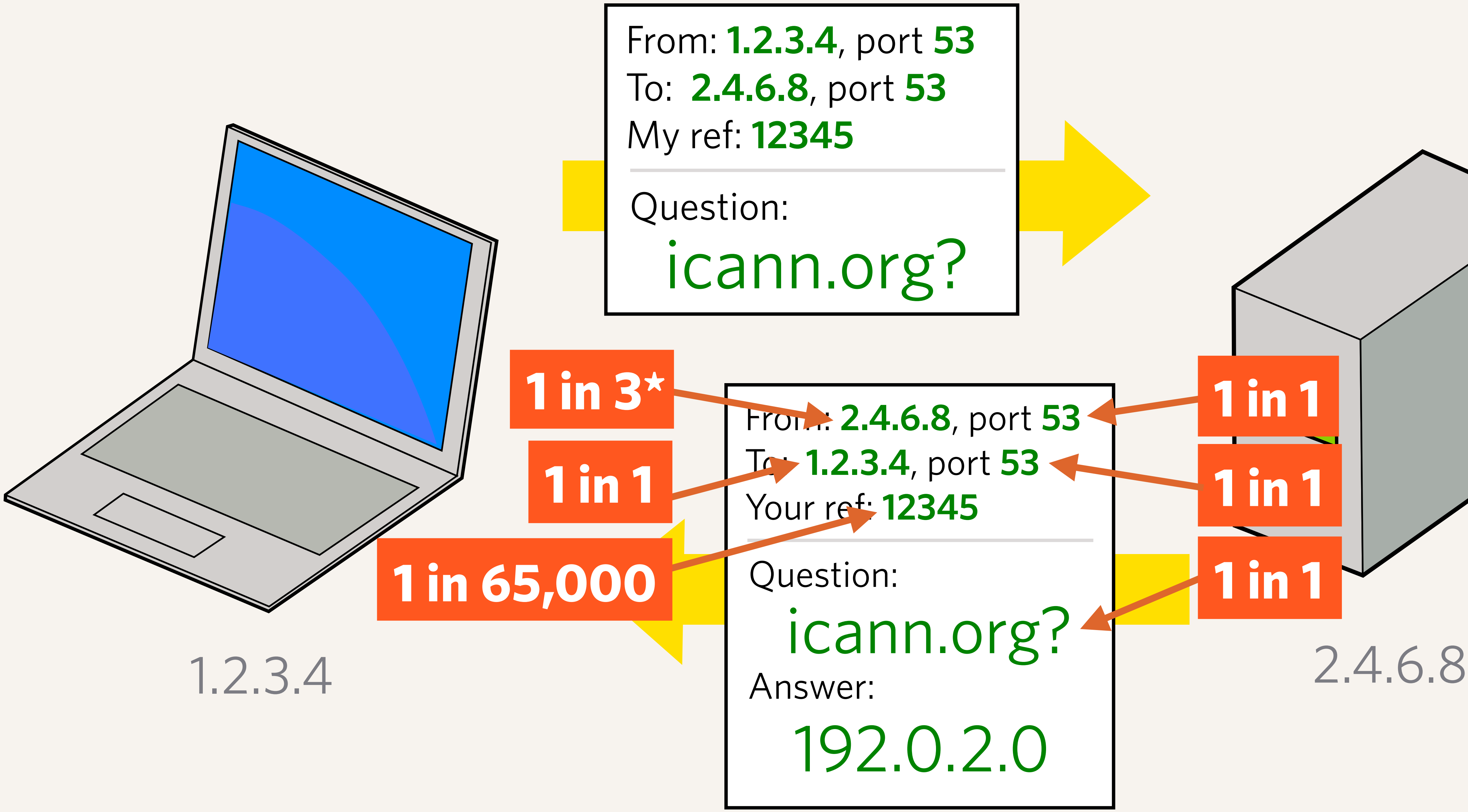
An impact on authoritative name servers

- This attack affects caching or recursive name servers that speed up DNS lookups at ISPs and corporate networks.
- Domain name zones are hosted on a different type of name server called an authoritative name server.
- If a name server provides both caching 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 domain on its authorities.

Short term solutions

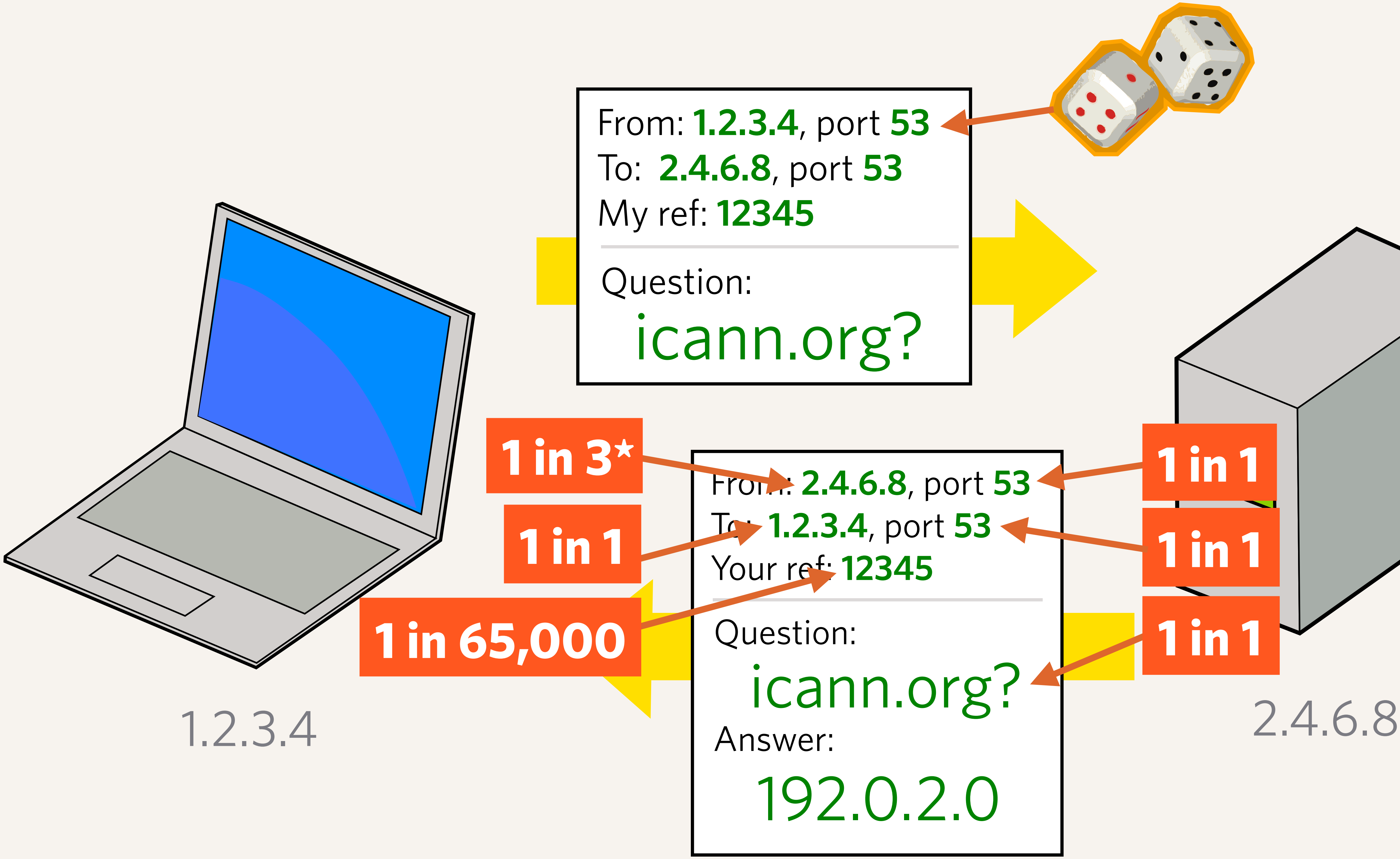
1. Maximise the amount of randomness

- Most implementations use randomised transaction numbers already. (The risk with that was discovered years ago, and fixed in most software)
- The port number 53 is assigned by IANA for DNS.
- However it is only required to be 53 as the *destination* port, not the *source* port.
- The patches that have been released in the last few months work by randomising the source port for the recursive server.



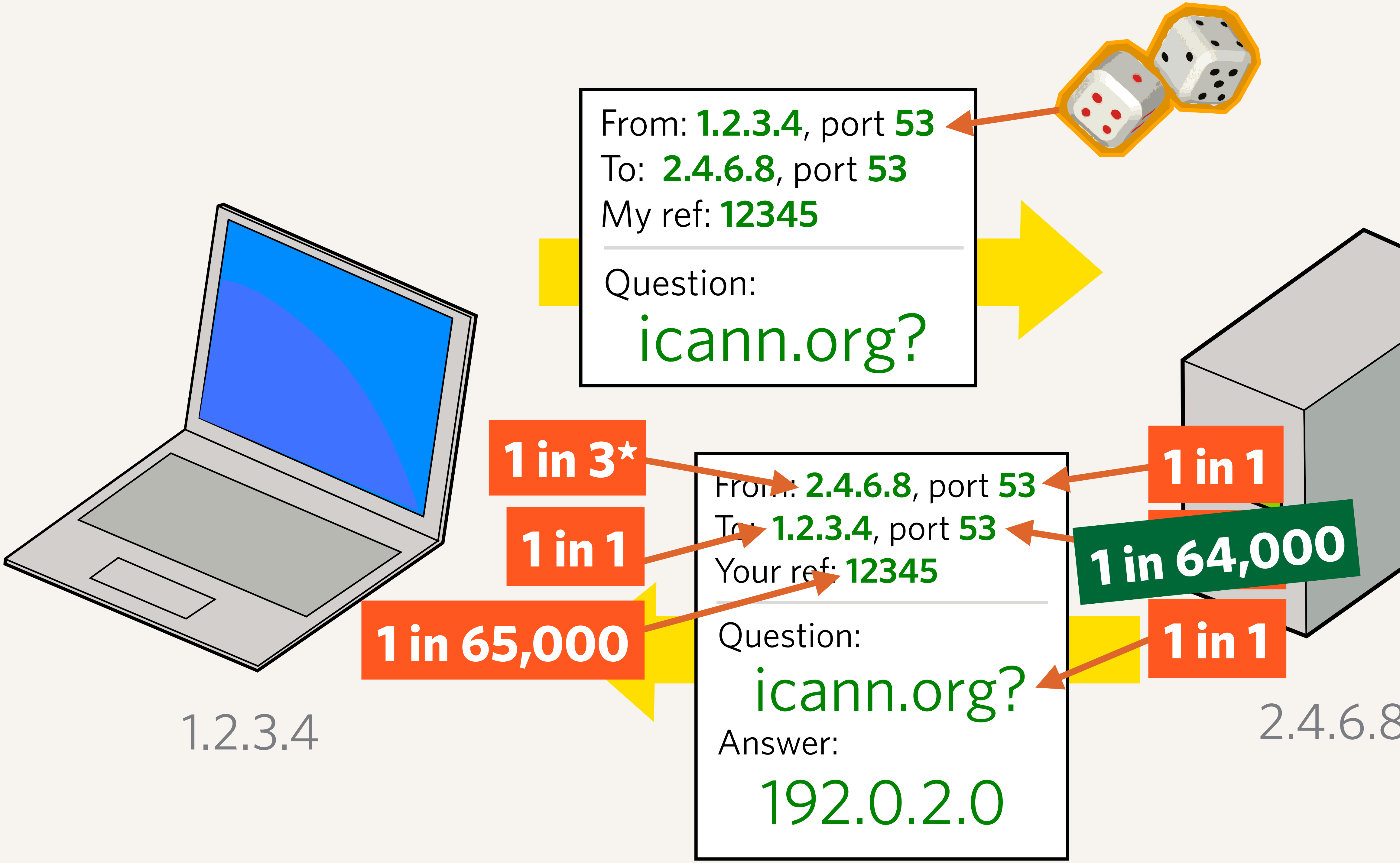
Possible combinations

Varying the source port increases the number of combinations



Possible combinations

Varying the source port increases the number of combinations



Possible combinations

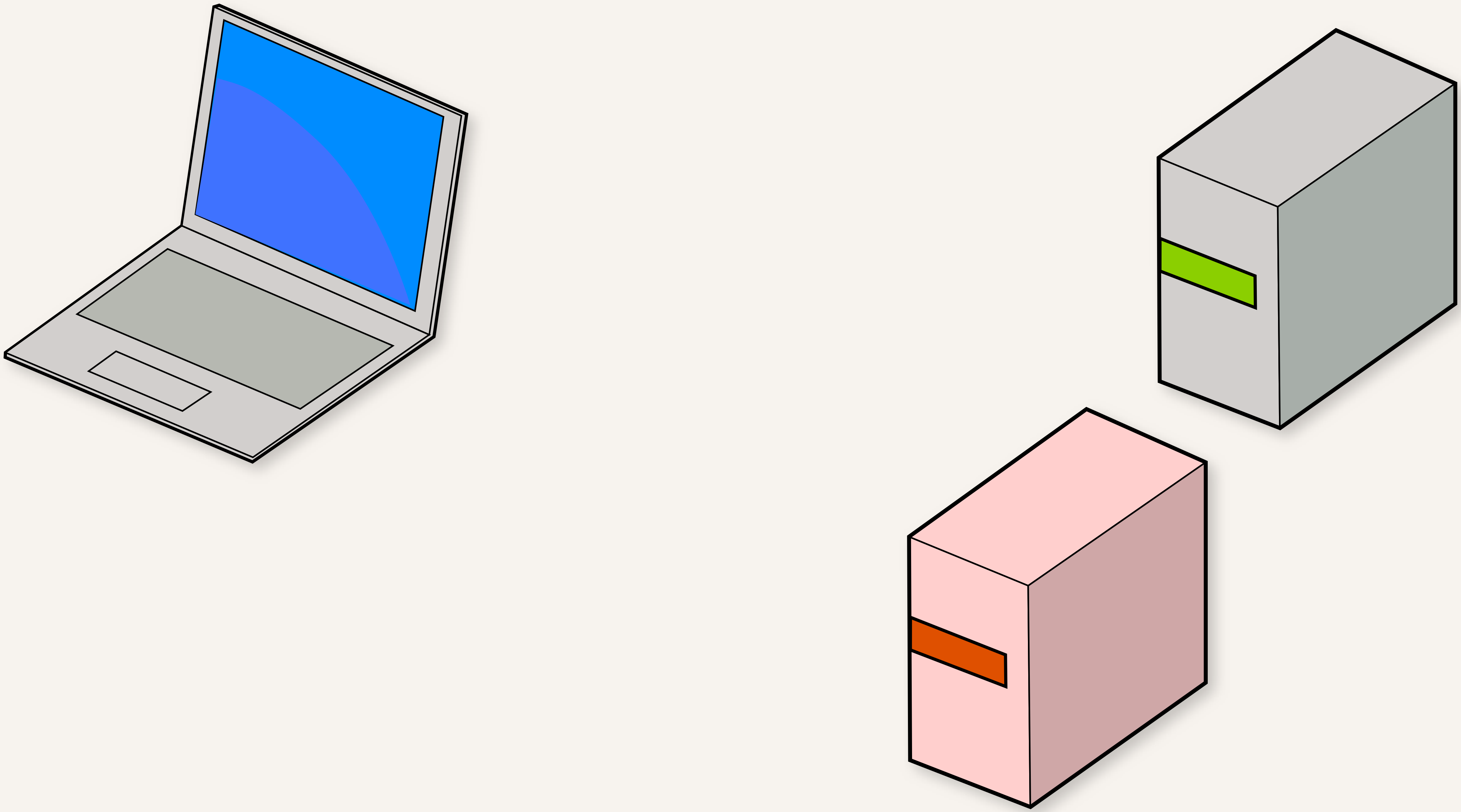
Varying the source port increases the number of combinations

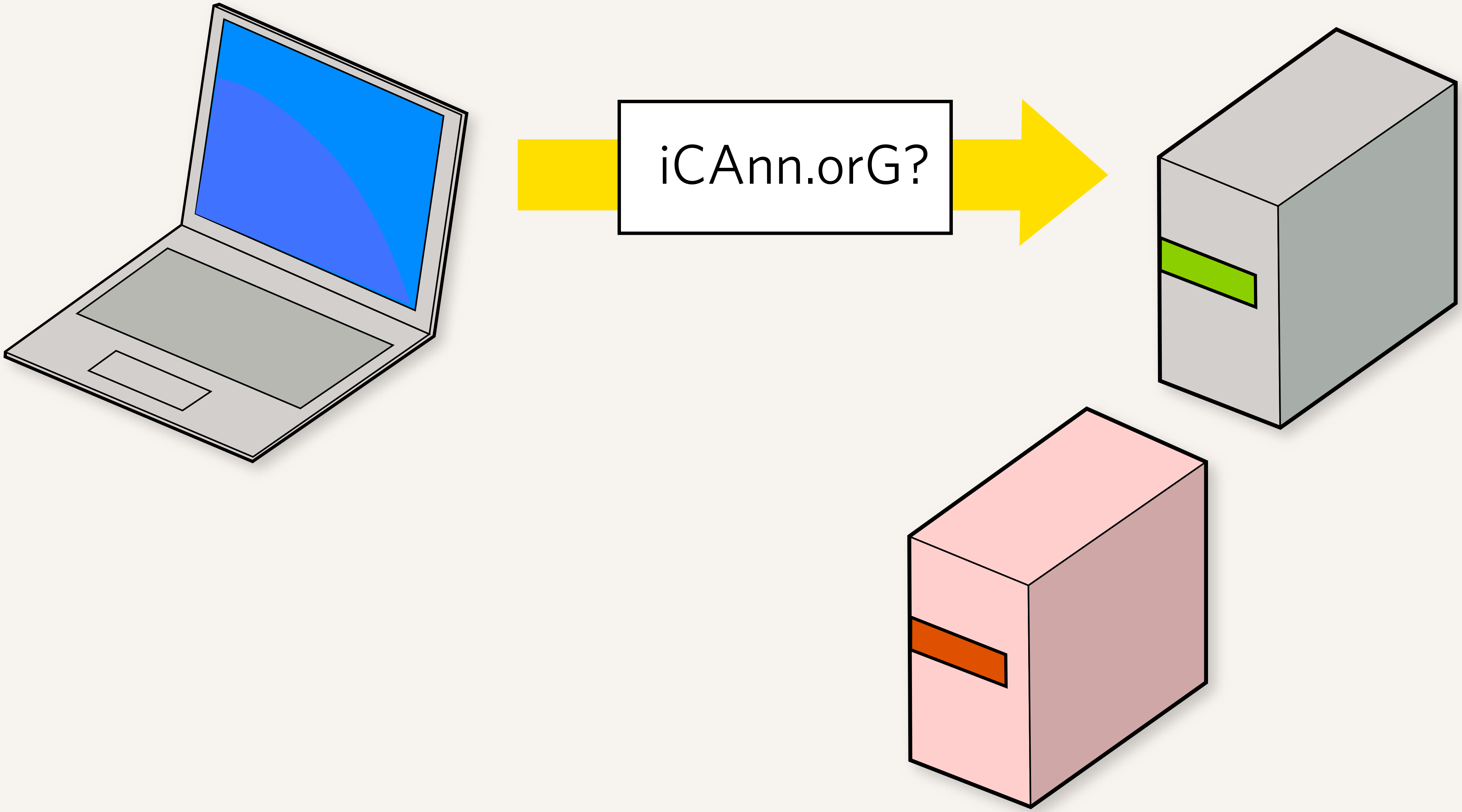
2. Disable open recursive name servers

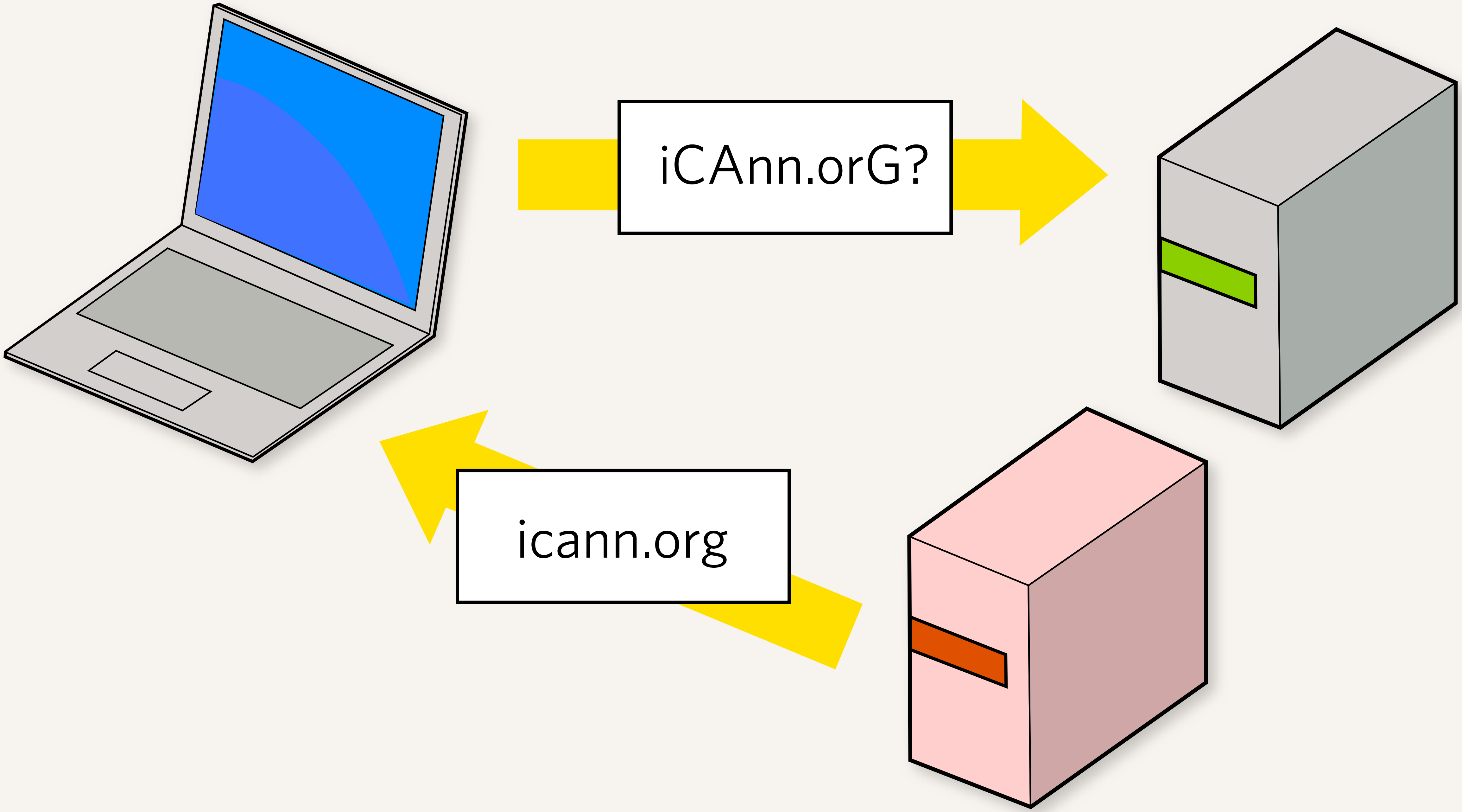
- The attack is not effective if the attacker can not send question packets to the name server.
- If you must run a recursive name server, limit access to only those computers that need it. (e.g. your customers). They will still be able to execute the attack, but the exposure is reduced.
- Turning off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

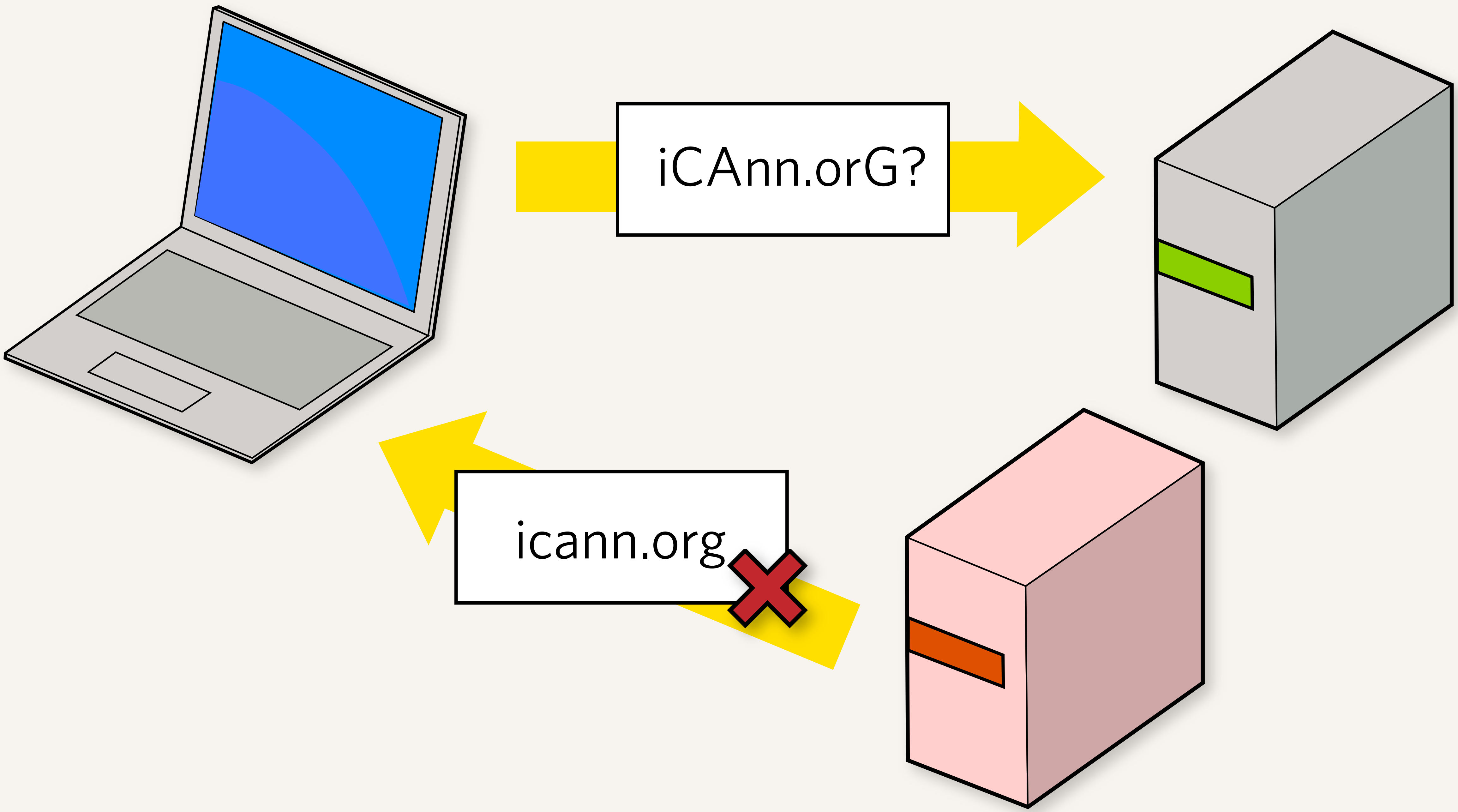
3. Use upper/lower case to add randomness

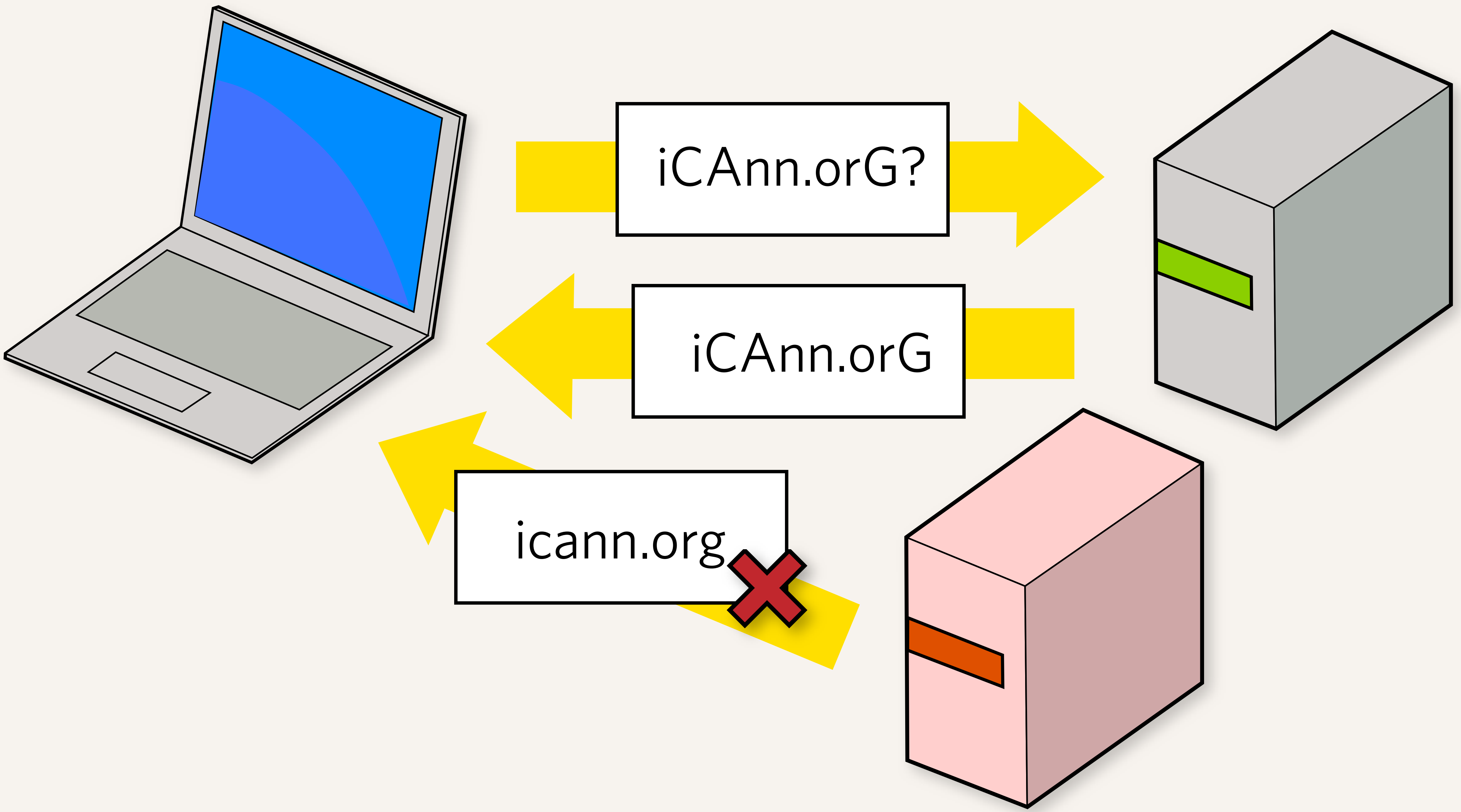
- The answer should preserve the same capitalisation as the question. By mixing upper and lower case, it provides more combinations that an attacker has to guess.
- This is a way of adding extra entropy to the DNS without modifying the protocol.
- Still under discussion (not implemented)

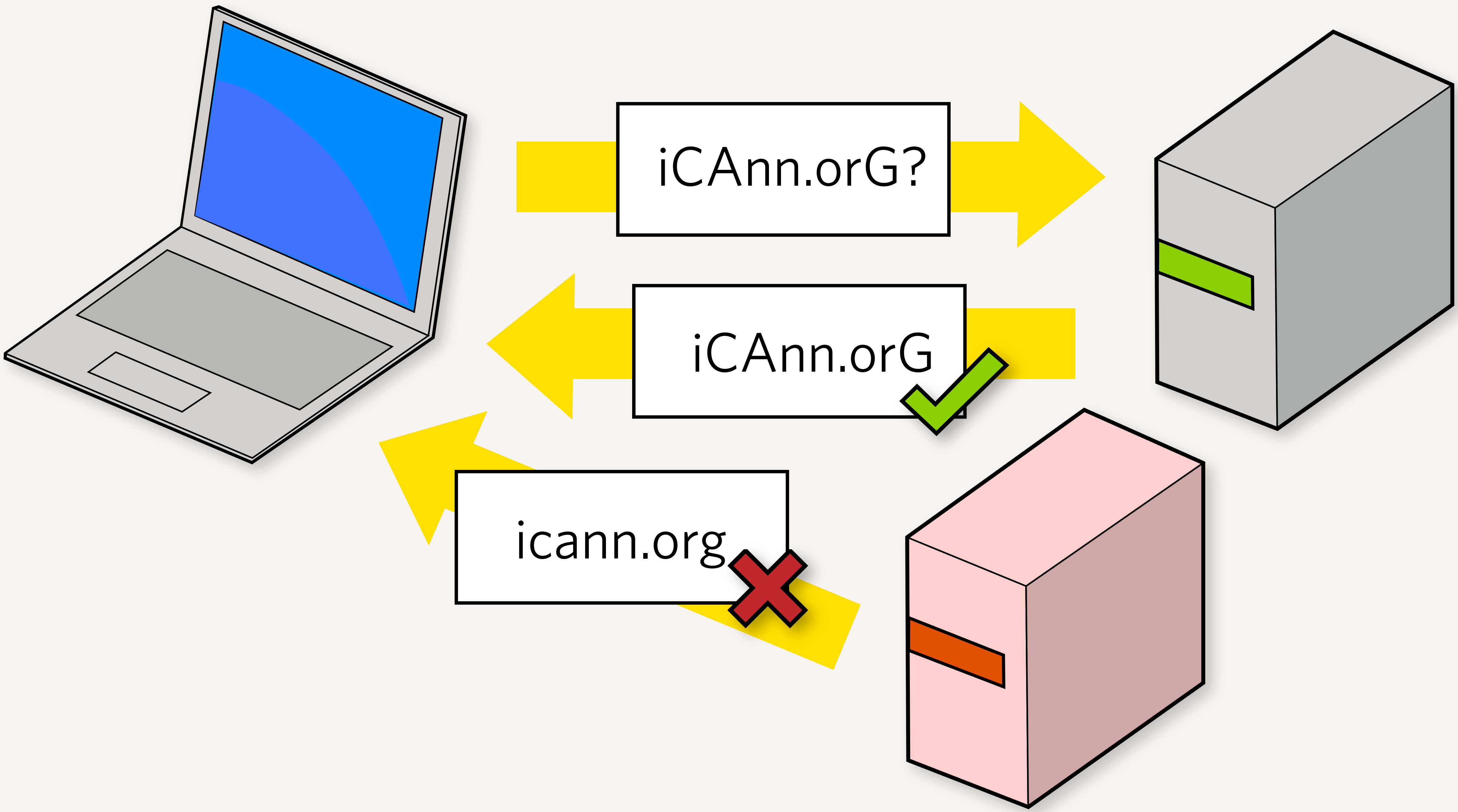


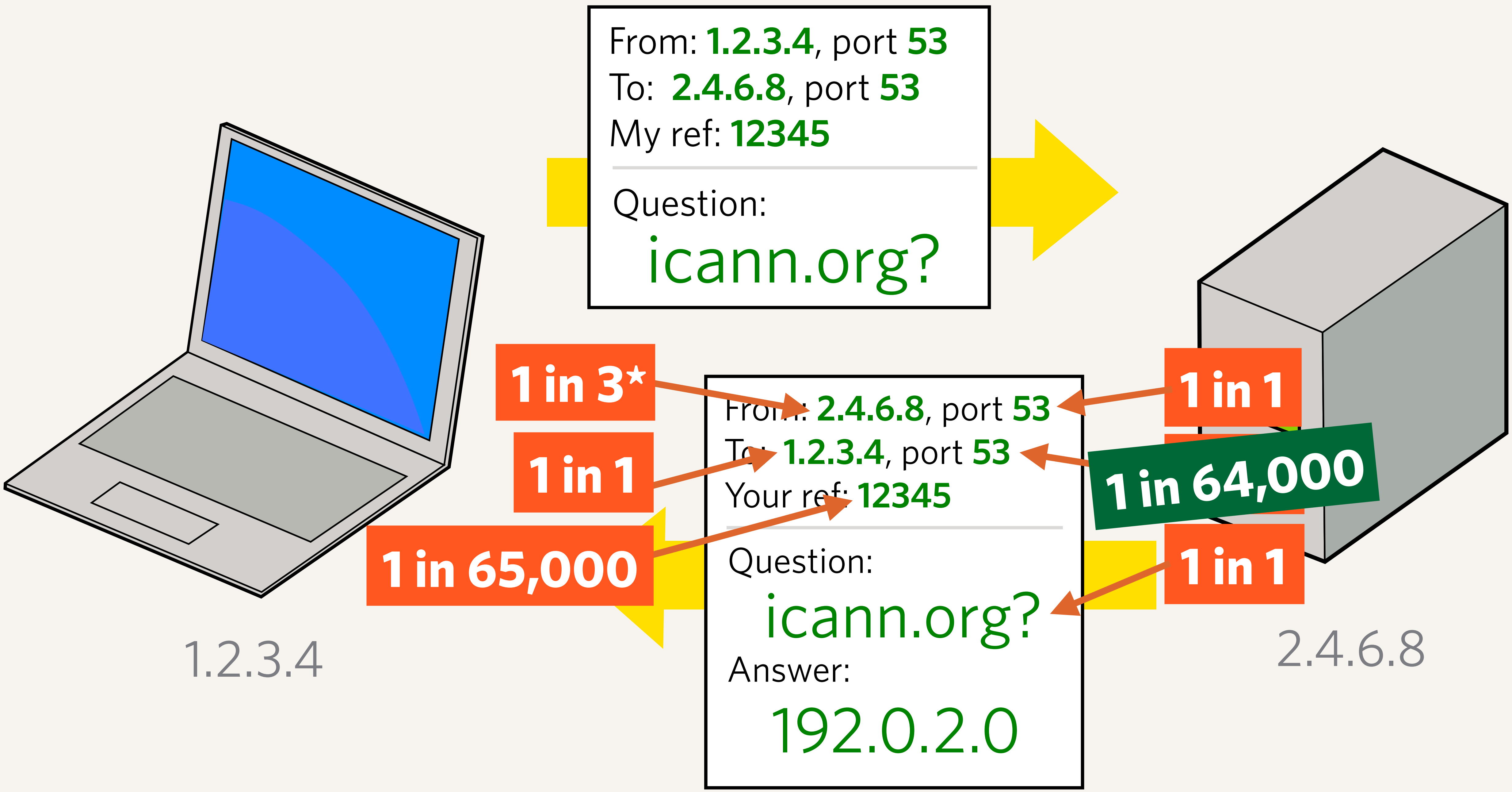

iCAnn.orG?


iCAnn.orG?
icann.org


iCAnn.orG?
icann.org

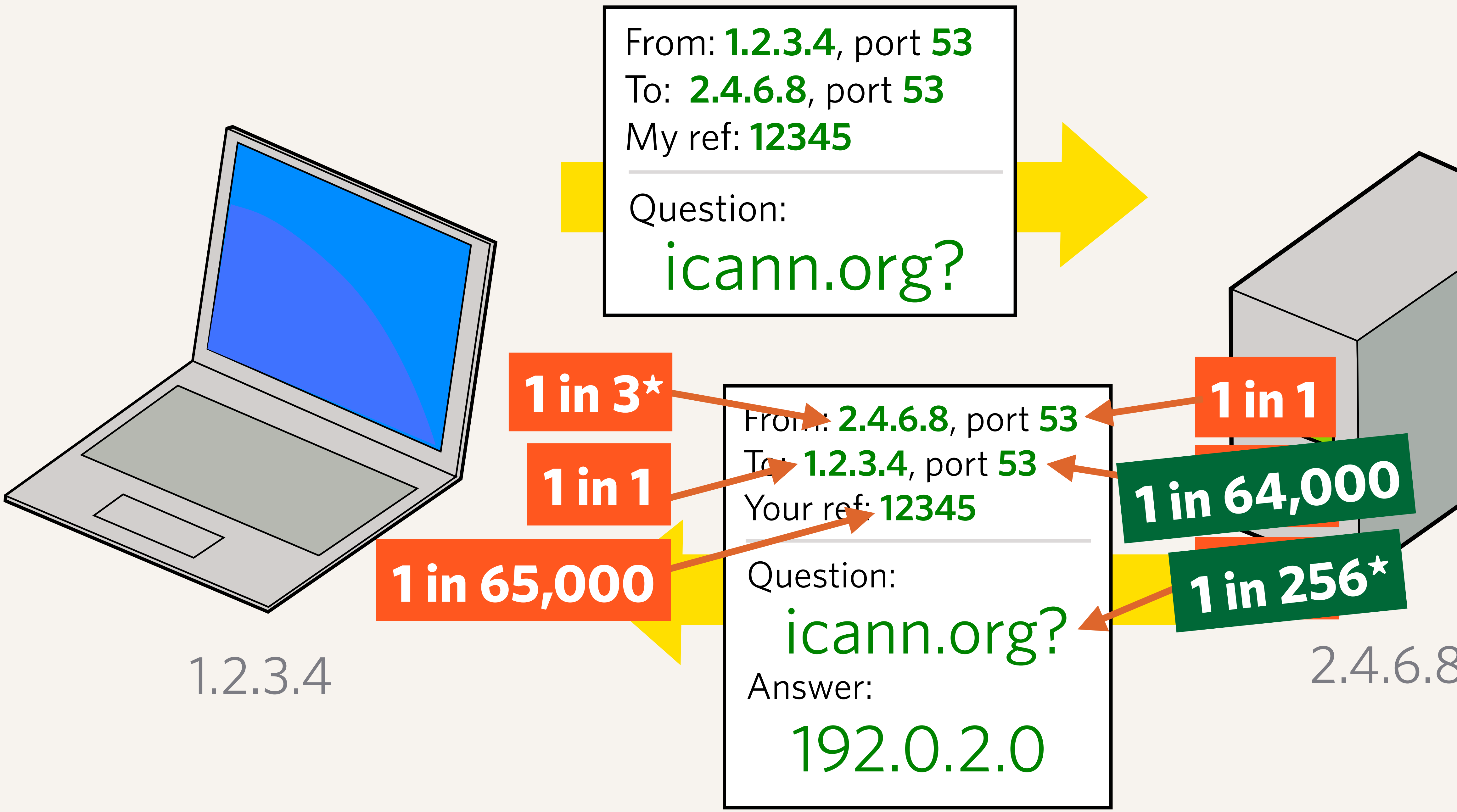

iCAnn.orG?
iCAnn.orG
icann.org


iCAnn.orG?
iCAnn.orG
icann.org



Possible combinations (3)

Varying the case increases the number of combinations to 2^L where L is the number of letters in the domain. (e.g. ICANN.ORG = 8 letters = 2^8 = 256)



Possible combinations (3)

Varying the case increases the number of combinations to 2^L where L is the number of letters in the domain. (e.g. ICANN.ORG = 8 letters = 2^8 = 256)

Net effect of short term solutions

- Old (unpatched) entropy ≈ 2^{16} to 2^{18} possibilities
  New (patched) entropy ≈ 2^{32} to $2^{(34+\text{length})}$ possibilities
- More entropy makes these types of attacks harder, but does not prevent them
- Computer processing power and network speeds will only increase in the future, improving the viability of these attacks

Long term solution

Introduce security to the DNS

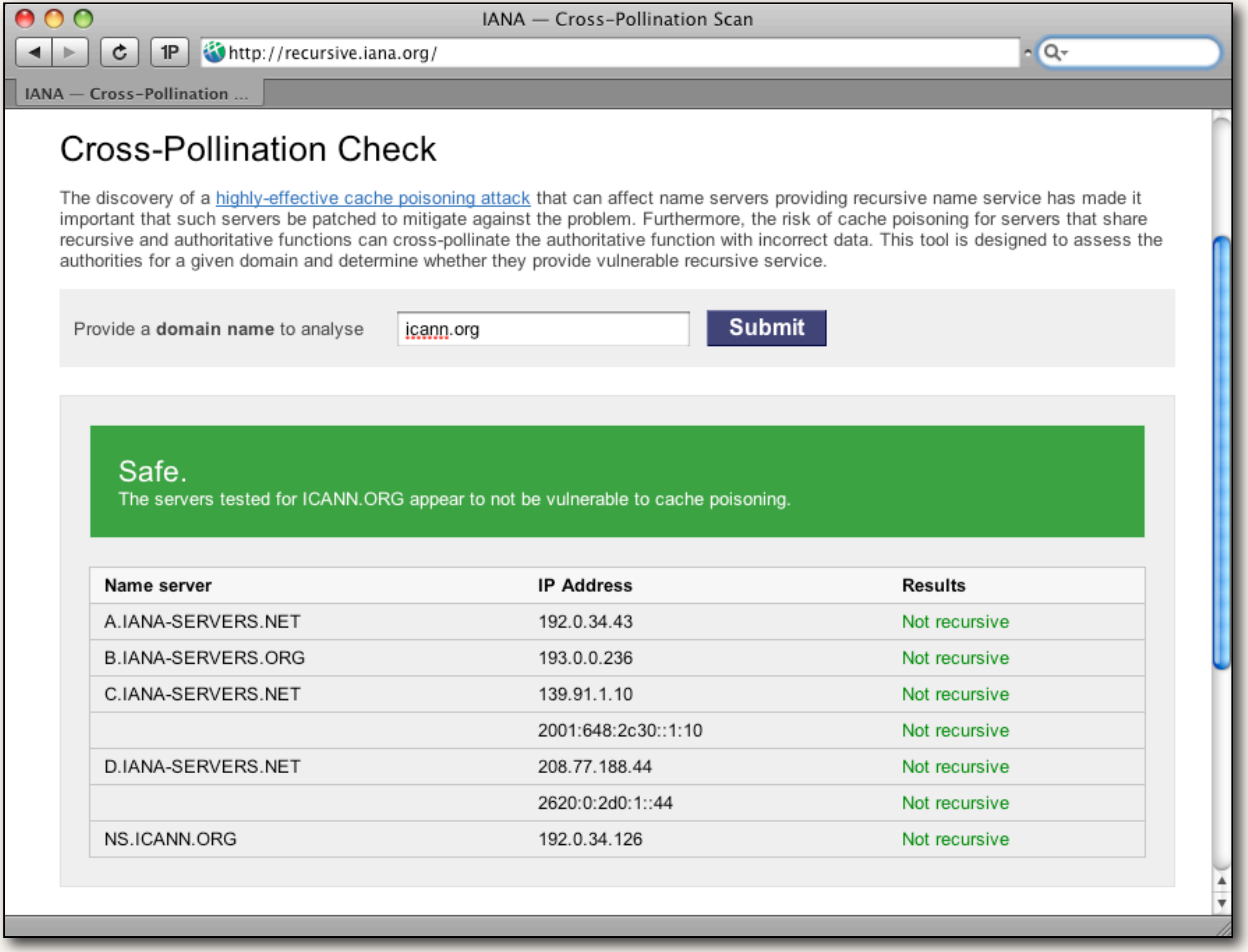
- The DNS is insecure. Upgrade the DNS for security.
- DNSSEC is the current answer to this problem.
- This attack provides clear incentive to deploy a solution like DNSSEC, because without security the DNS will continue to be vulnerable to cache poisoning attacks.

Impact on TLDs

- At the time the vulnerability became 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

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 re-implemented this way.
- The tool is not TLD specific, and works with any domain name.

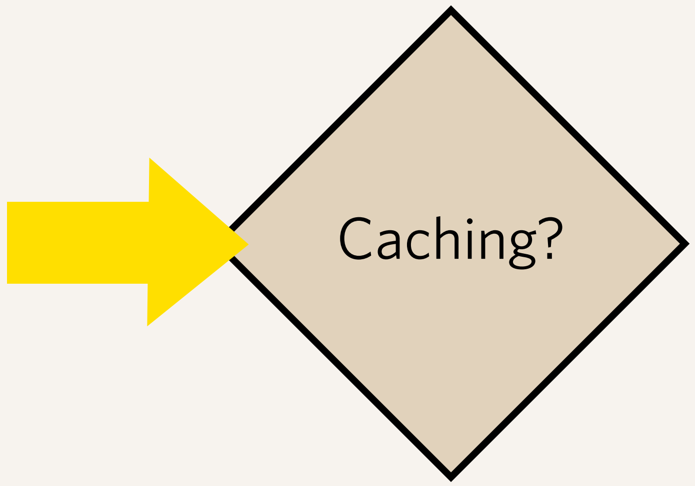


Vulnerability checking tool

http://recursive.iana.org/

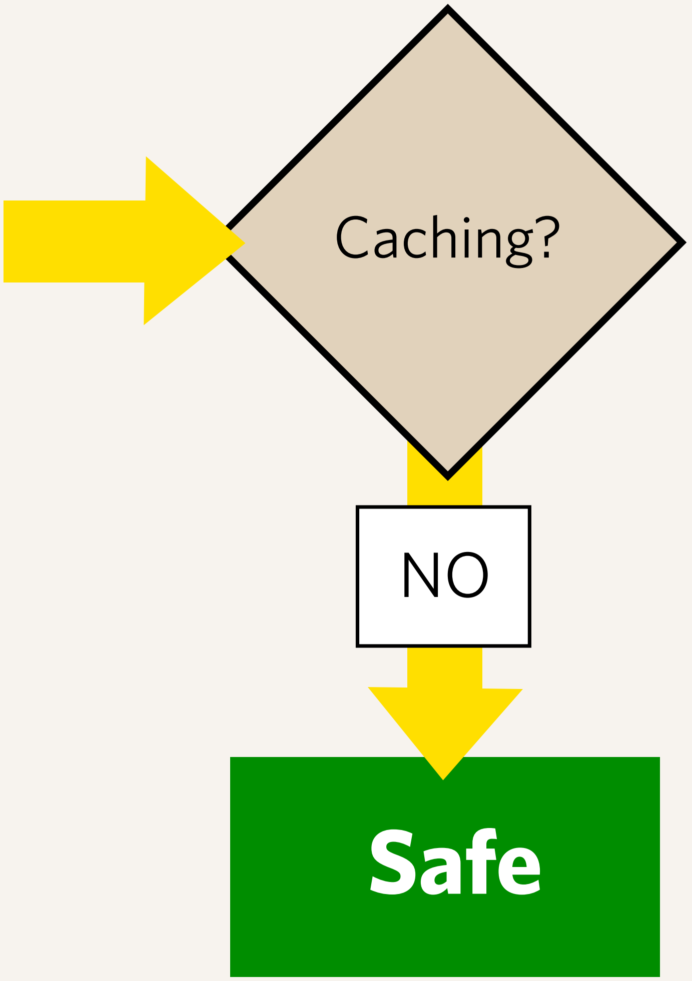
How the tool works

The tool checks for the two aspects that enable the attack



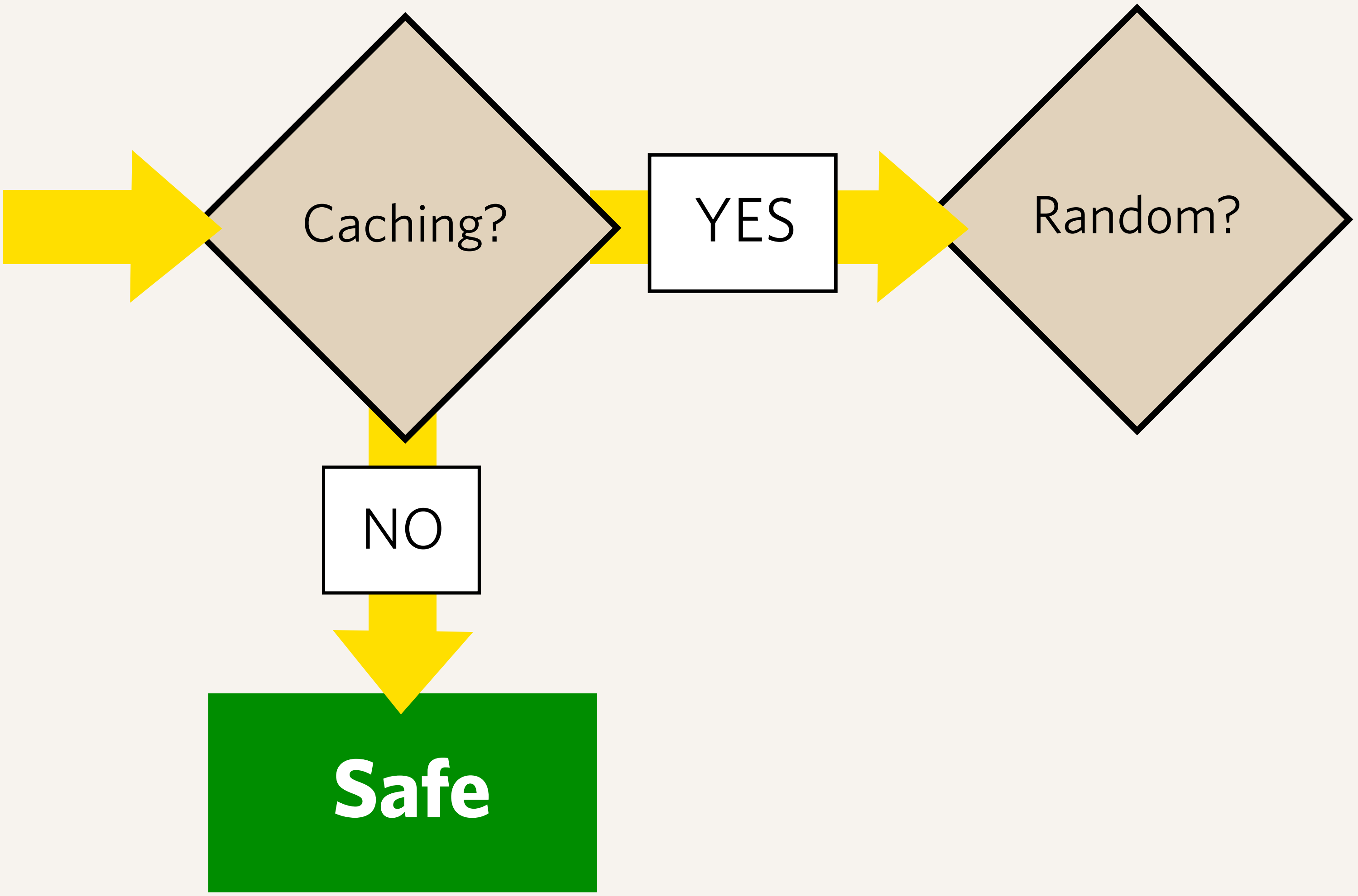
How the tool works

The tool checks for the two aspects that enable the attack



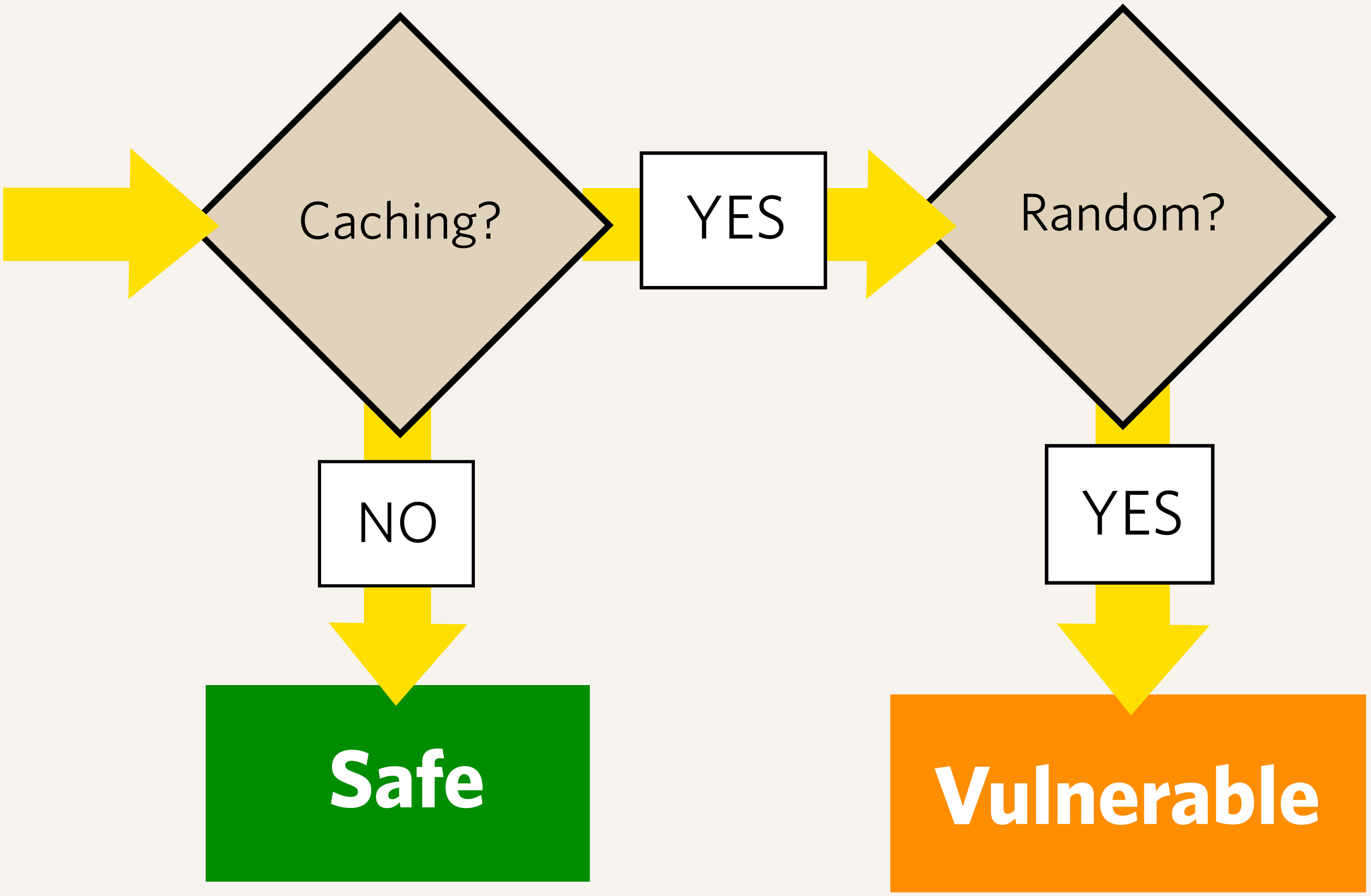
How the tool works

The tool checks for the two aspects that enable the attack



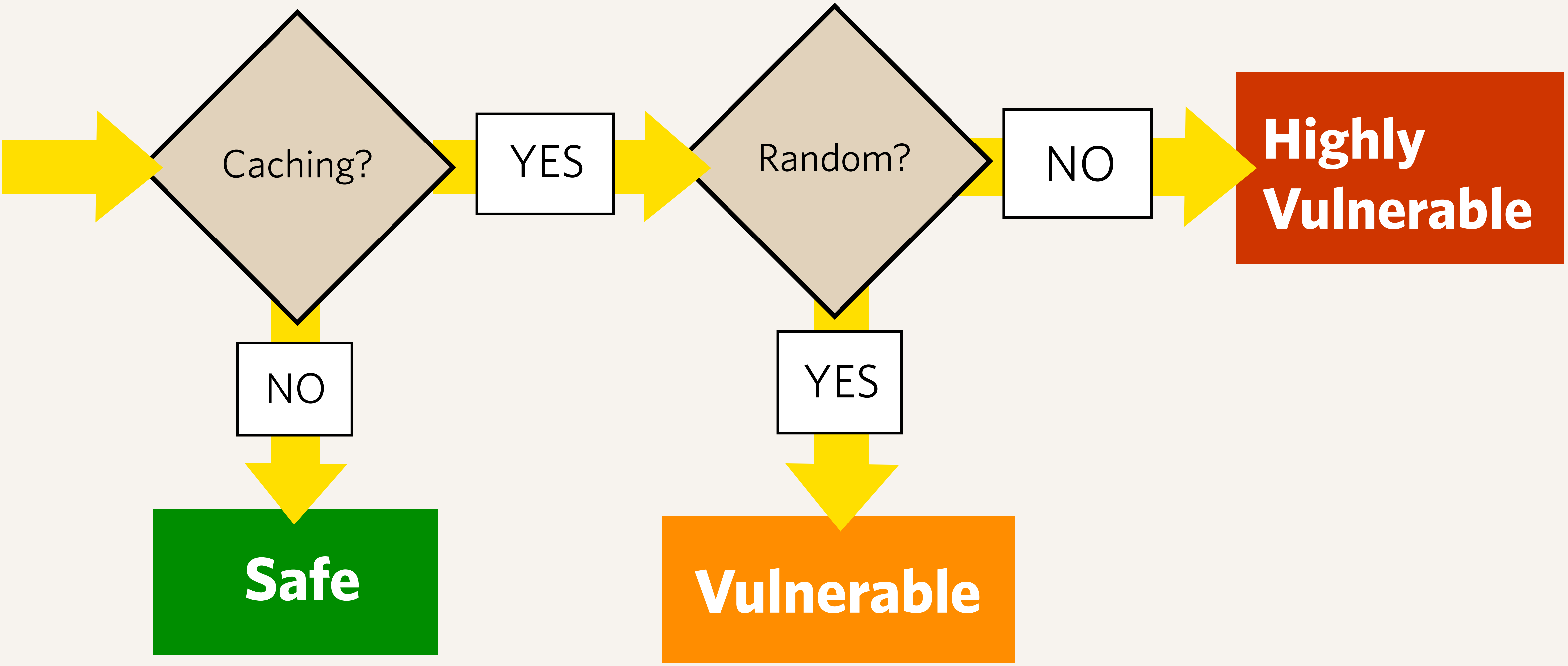
How the tool works

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


CHANGE
CHANGE
WE NEED
WWW.BARACKOBAMA.COM

Thanks!

kim.davies@icann.org