

Transparent Forwarders: An Unnoticed Component of the Open DNS Infrastructure

Marcin Nawrocki, Maynard Koch, Thomas C. Schmidt, Matthias Wählisch

`{marcin.nawrocki, maynard.k, m.waehlich}@fu-berlin.de`
`t.schmidt@haw-hamburg.de`

In a nutshell

Do common DNS scanning methods detect all relevant parts of the open DNS ecosystem?

No.

They ignore transparent forwarders, which compose 26% of the open DNS ecosystem.

Why should you care? Open DNS enables amplification attacks!

Attacker



Spoofed DNS Request

Open DNS



Unwanted DNS Response

Victim



Our contributions

1. Comparison of DNS measurement methods
2. Measurements that reveal transparent forwarders
3. Comprehensive analysis of transparent forwarders
4. DNSRoute++ to explore DNS infrastructure
5. Impact assessment on attack surface

Our contributions

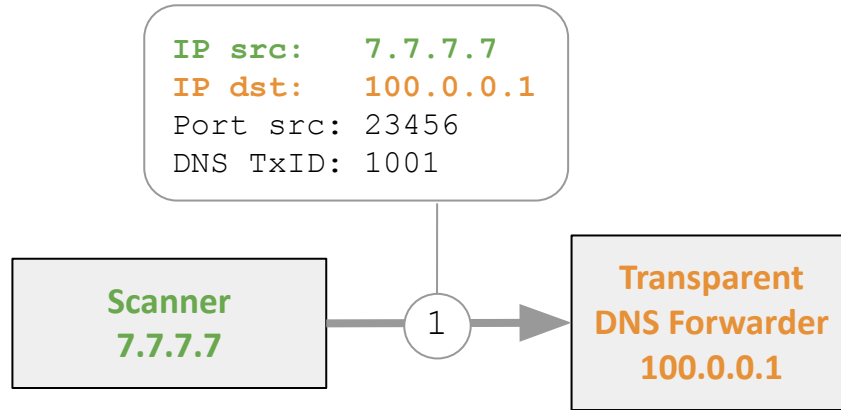
Details, see paper.

1. Comparison of DNS measurement methods
2. Measurements that reveal transparent forwarders
3. Comprehensive analysis of transparent forwarders
4. DNSRoute++ to explore DNS infrastructure
5. Impact assessment on attack surface

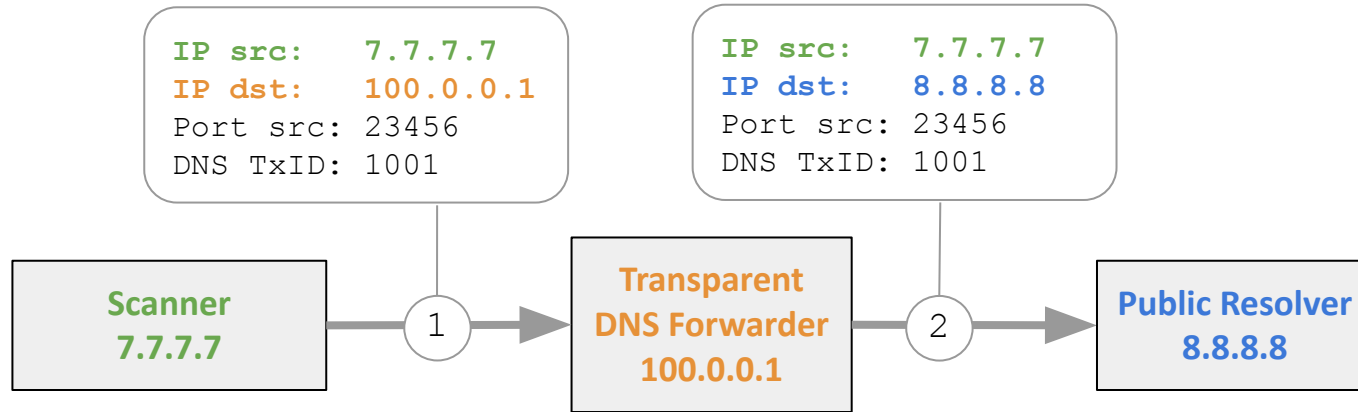
How do transparent forwarders work?

**Transparent
DNS Forwarder
100.0.0.1**

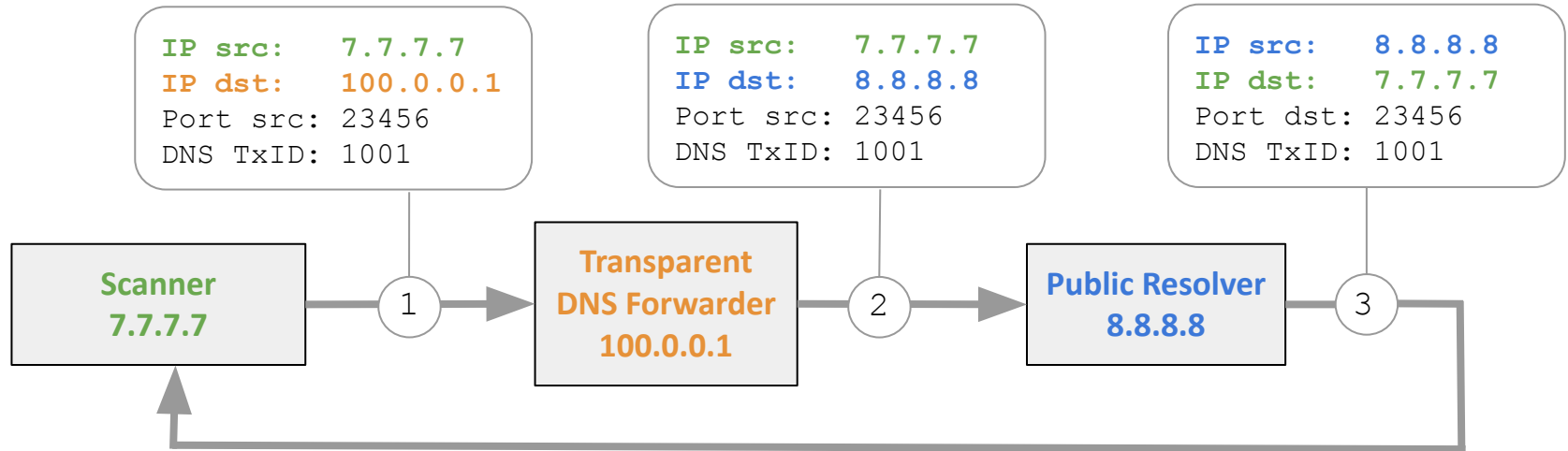
How do transparent forwarders work?



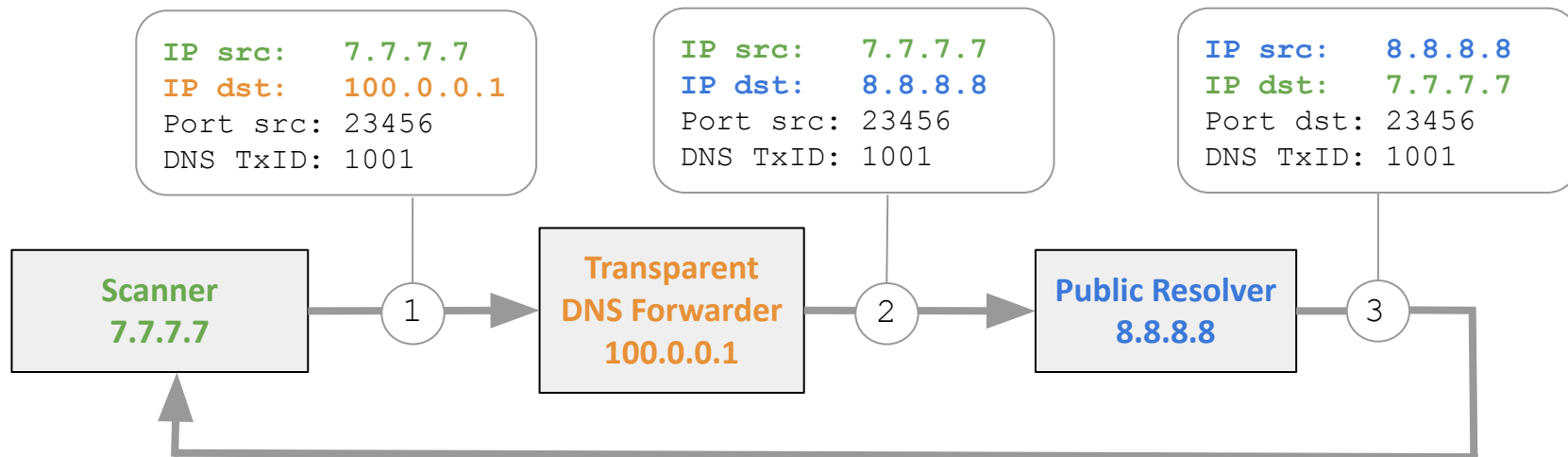
How do transparent forwarders work?



How do transparent forwarders work?

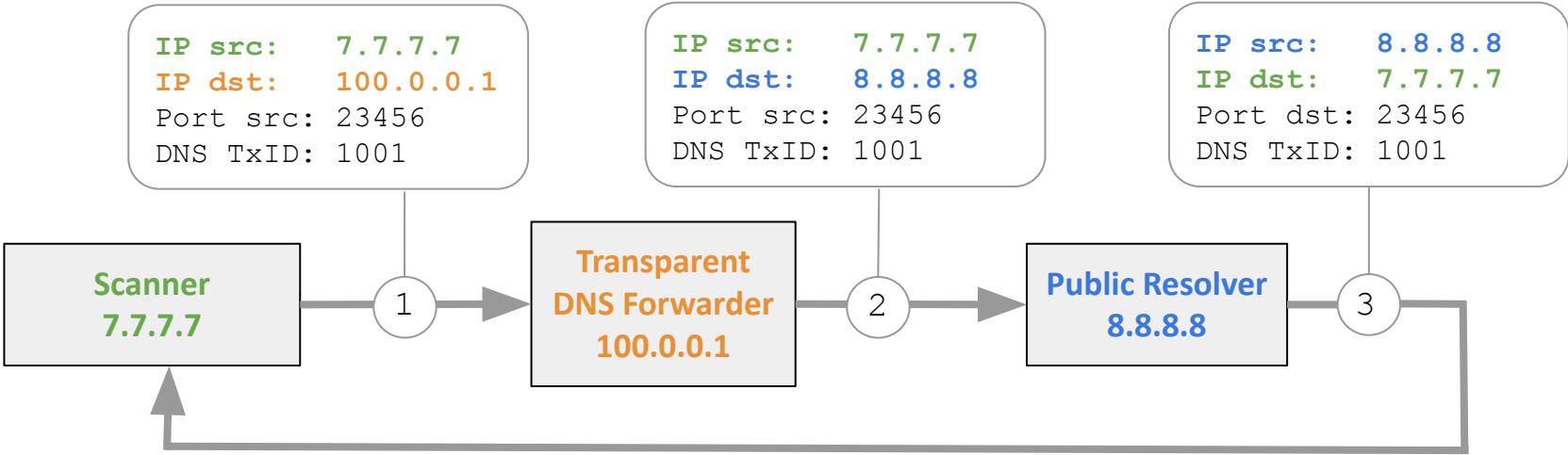


How do transparent forwarders work?



Transparent forwarders
send spoofed packets.

How do transparent forwarders work?

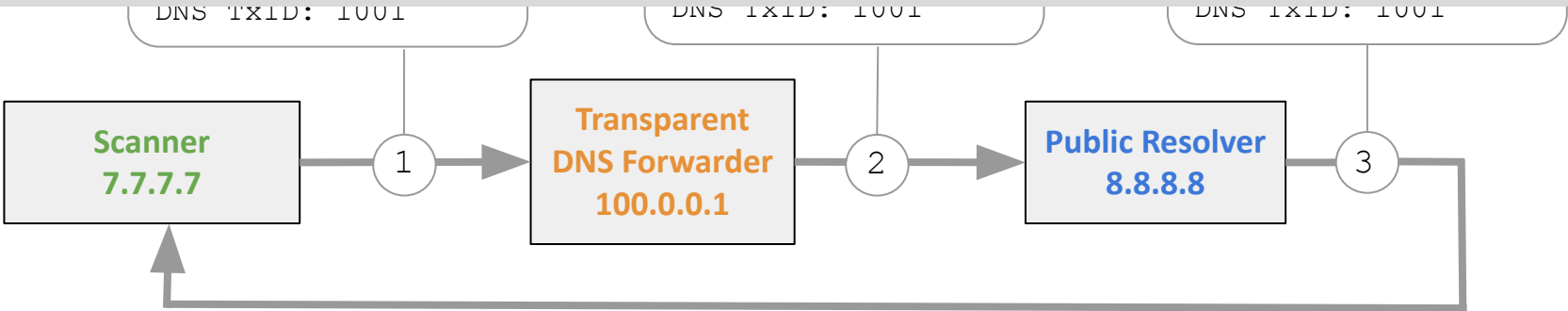


Transparent forwarders send spoofed packets.

Scanners that consider only the replying source IP address, make mistakes.

How do transparent forwarders work?

This behavior does not comply with DNS interception or redirection methods.



Transparent forwarders send spoofed packets.

Scanners that consider only the replying source IP address, make mistakes.

How do transparent forwarders work?

This behavior does not comply with DNS interception or redirection methods.

DNS TXID: 1001

DNS TXID: 1001

DNS TXID: 1001





Transparent DNS forwarders have been identified first in 2013 by Jared Mauch but common DNS campaigns do not cover them.

100.0.0.1

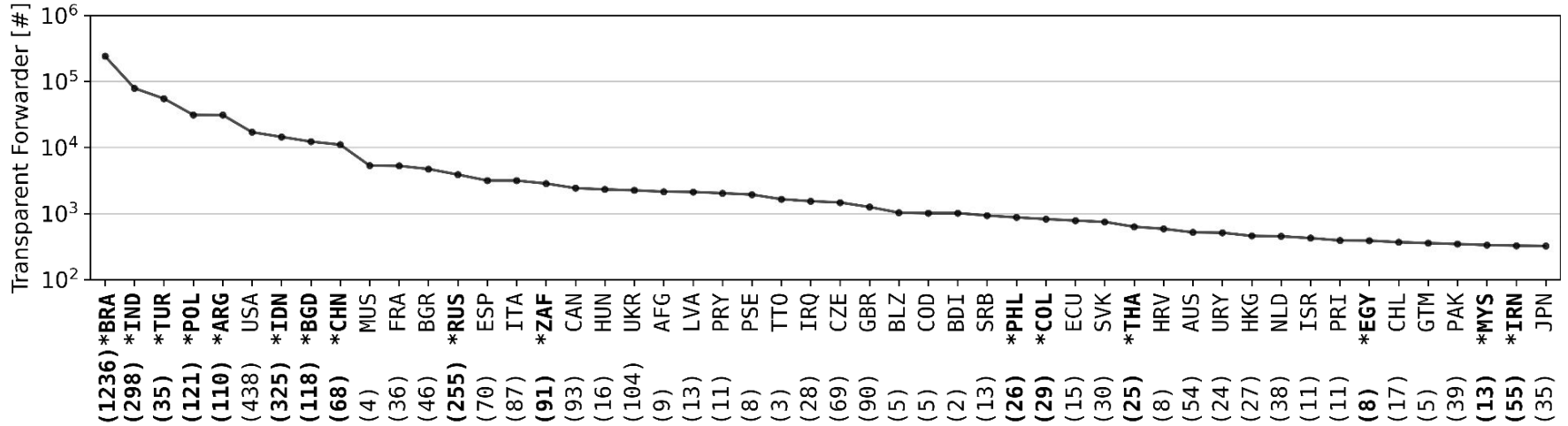
Transparent forwarders
send spoofed packets.

Scanners that consider only
the replying source IP
address, make mistakes.

Our controlled experiment confirms that transparent DNS forwarders fell of the radar.

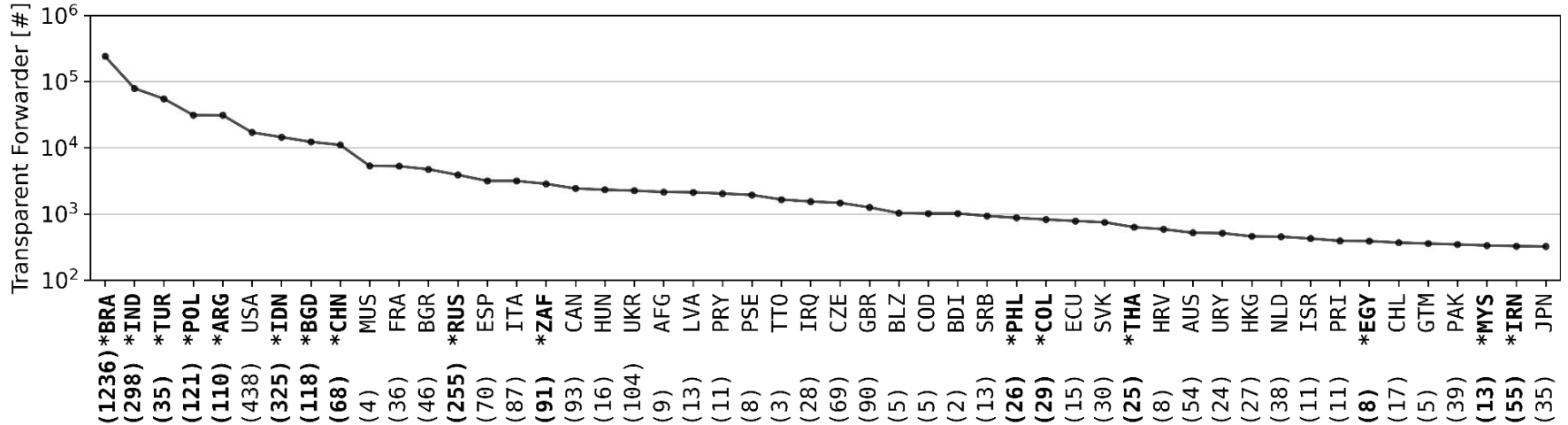
	Censys	Shadowserver	Shodan	Our Scans
# ODNS	1.75M	1.8M	1.6M	2.15M
Transparent forwarders detected				 (26% forwarders)

Where is transparent forwarder deployment most popular?



Top 50 Countries Descending by Transparent Forwarders; * Emerging Markets and (#ASes) with a Transparent Forwarder

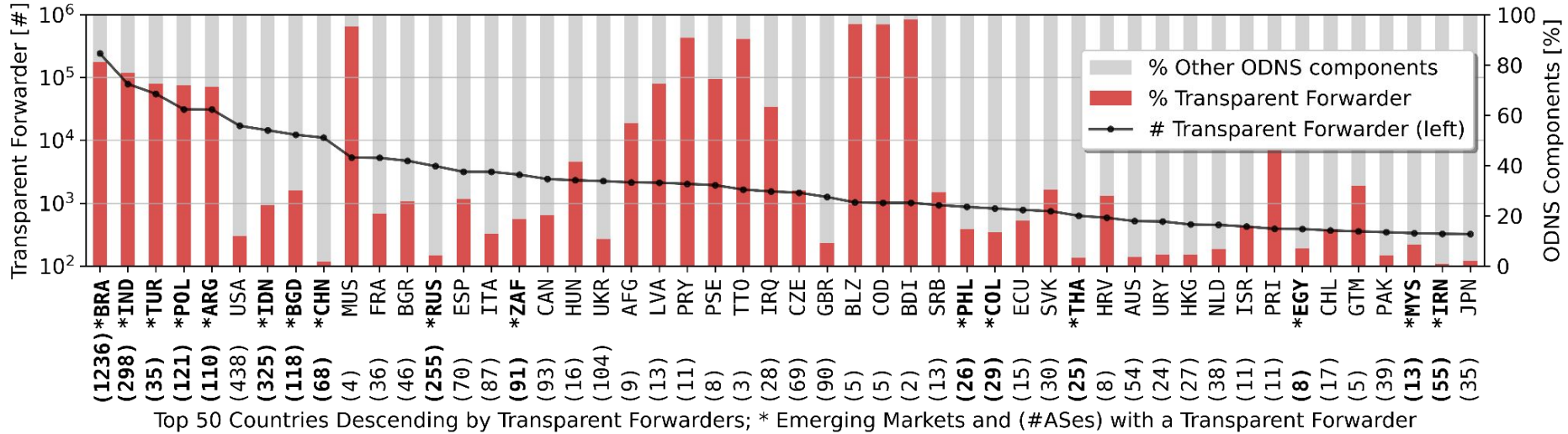
Where is transparent forwarder deployment most popular?



Top 50 Countries Descending by Transparent Forwarders; * Emerging Markets and (#ASes) with a Transparent Forwarder

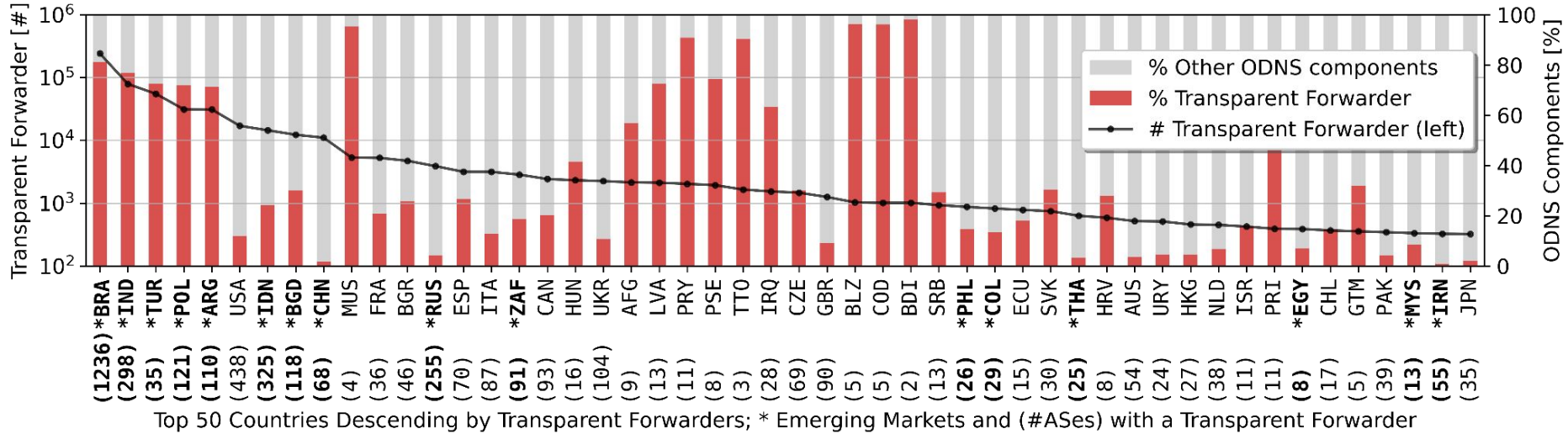
1. Countries classified as emerging markets are more likely to host transparent forwarders
2. In each country, multiple ASes host forwarders.

Where is transparent forwarder deployment most popular?



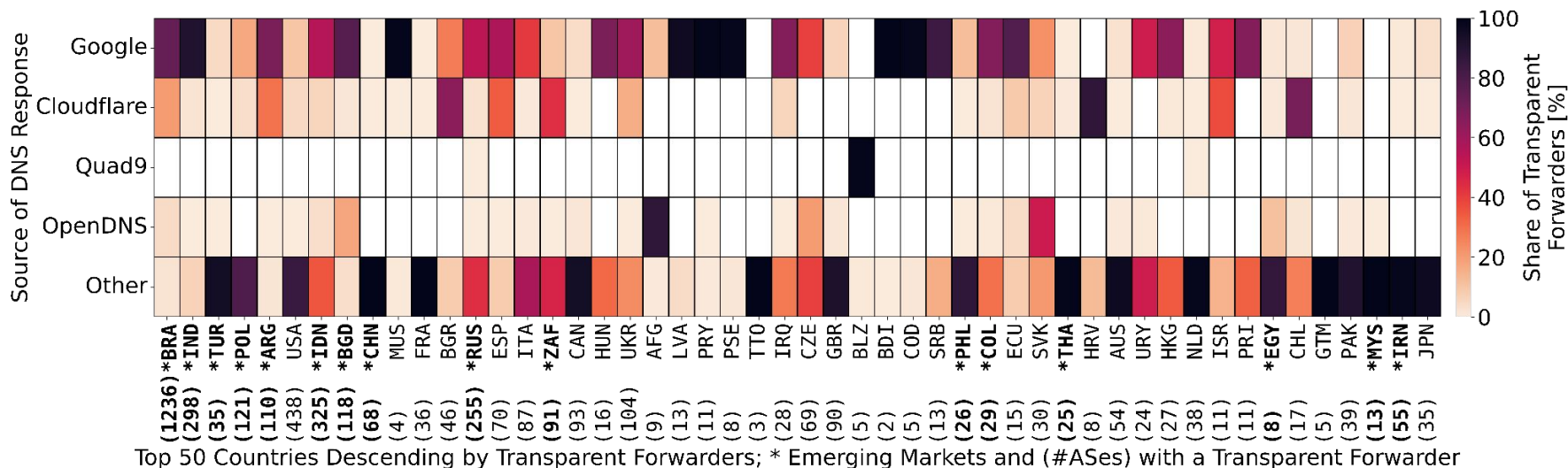
1. Countries classified as emerging markets are more likely to host transparent forwarders
2. In each country, multiple ASes host forwarders.

Where is transparent forwarder deployment most popular?



1. Countries classified as emerging markets are more likely to host transparent forwarders
2. In each country, multiple ASes host forwarders.
3. In some countries, the ODNS consists almost exclusively of transparent forwarders.

Which recursive resolvers are used by transparent forwarders?

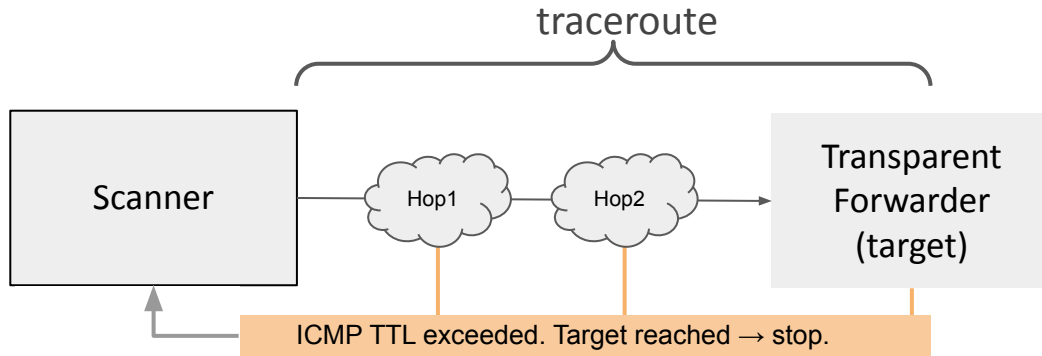


In some countries, significant consolidation is visible.

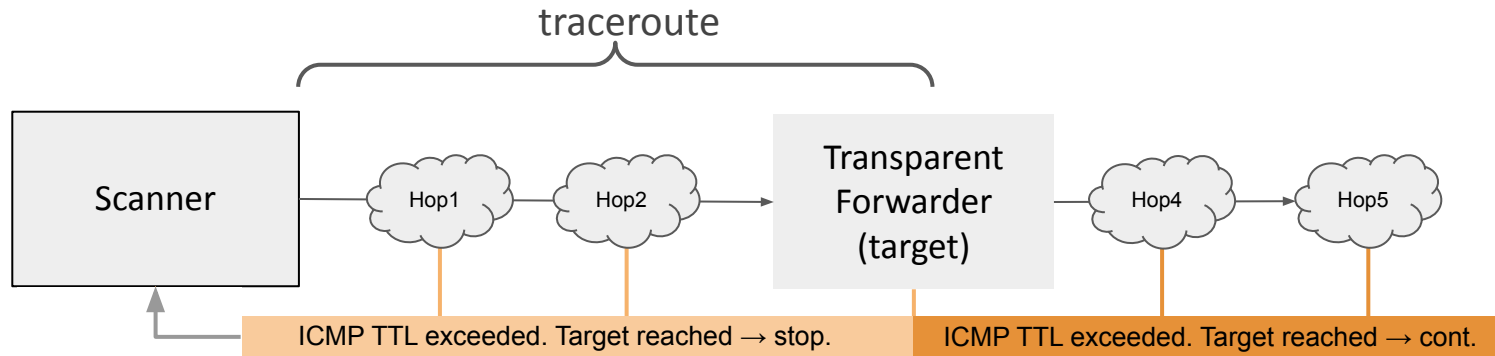
Potential for measurements: DNSRoute++

Transparent forwarders manipulate **only** the IP destination address.
They keep all the other header values as-is.

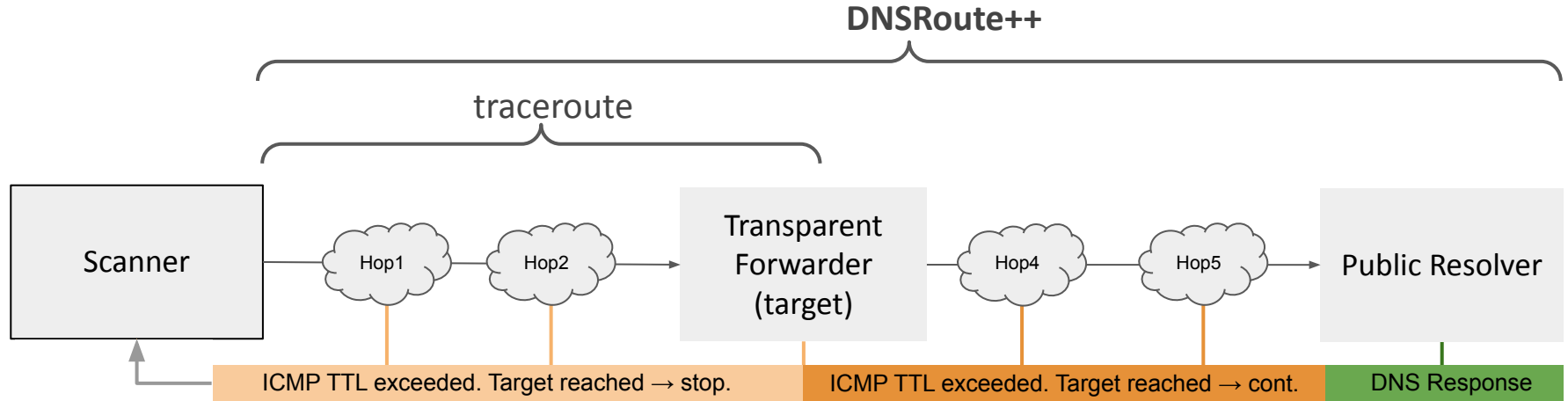
1. DNSRoute++ increments the IP TTL like common traceroute



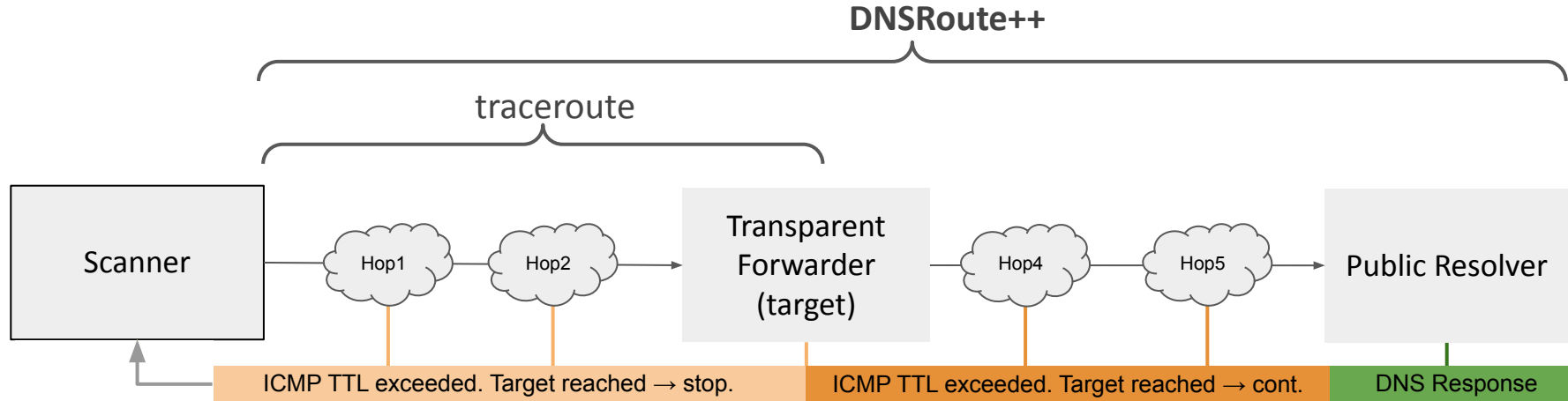
2. ... but sends DNS queries + keeps incrementing TTL after target



3. ... it stops when DNS response received cos resolver reached.

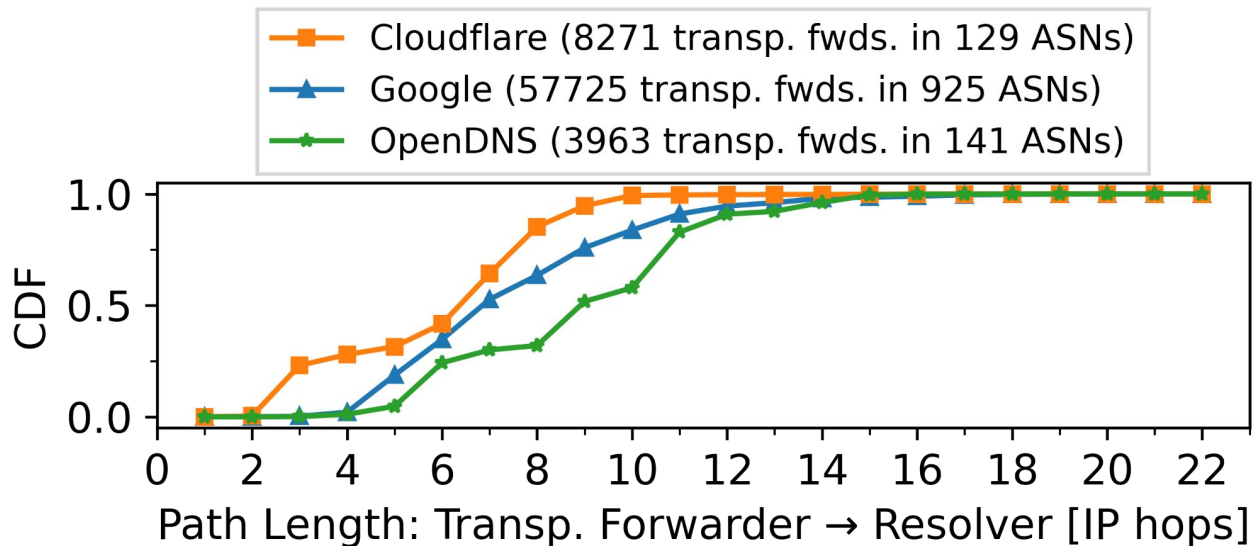


DNSRoute++ discovers paths between forwarders and resolvers.



1. This confirms that transparent forwarders spoof source and only rewrite destination address.
2. This also allows us to use transparent forwarders to measure path lengths to anycast DNS resolvers.

Which anycast DNS deployment leads to shorter paths?



Conclusion

All artifacts and
new measurements



The open DNS system includes transparent forwarders (26%).

Relative amount increased over the last years.

Popular DNS measurement campaigns miss transparent forwarders.

They can be captured with low-overhead transactional scans.

Transparent forwarders can be leveraged as vantage point.

They reveal further DNS consolidation. DNSRoute++ explores infrastructure.

Transparent forwarders often deployed in emerging-markets and ease attacks.

We should start reducing the attack surface.

++ QnA Preps ++

What is the major reason for missing transparent forwarders?

Transparent forwarders are basically missed because of *too* efficient scanning methods.

For example, think of zmap which can scan the complete IPv4 space under an hour. A common zmap configuration only records the incoming messages, so only the DNS responses.

However, all this happens based on the unfortunately wrong assumption that you do not need to correlate DNS requests and responses. That's why they are missed.

What kind of devices are transparent forwarders?

In most cases, transparent forwarders are misbehaving CPE devices (or home-gateways) in ISP networks. We performed 3 measurements to better understand the deployment:

1. We used Shodan data and checked all the other ports, OS fingerprints and so on. And the majority of devices that could be fingerprinted were MikroTik devices, which is a CPE producer. However, many devices also remained unclassified.
2. We used PeeringDB and manual inspection to classify the autonomous systems which host transparent forwarders. Again, most of them were ISPs.
3. Also, to understand whether transparent forwarders are individual devices or middleboxes that operate prefix-wide, we checked the density of transparent forwarders in each prefix which hosts transparent forwarders. There are some cases of prefix-wide operation, however, we usually observed only a couple of devices per prefix. So it all looks like individual devices.

What is the misuse potential?

They can be misused for amplification attacks. I think we should consider two points:

1. Transparent forwarders forward the source IP address unchanged. So an already spoofed IP address will be simply forwarded also spoofed. This means that attackers can ingest spoofed traffic anywhere in the world (actually in over 170 countries), which makes attack attribution way more difficult.
2. Also, similar to our DNSroute++ measurement, they can be utilized as “jump host” to reach different PoPs of the same provider, for example different Google PoPs. This makes PoP-based DDoS mitigation way more difficult, because the attack traffic is distributed across multiple PoPs.

Which DNS scanning method do you recommend?

I would recommend our methodology which we call transactional scans. For DNS, you have two ways to correlate requests and responses:

1. What we did is: We recorded the outgoing scan traffic and used random client ports as well as random DNS transaction IDs for each request. The nice thing here is that this allows us to always issue the same DNS query (so we still can scan fastly) but also to utilize DNS caches. Then, **in the case of a response, we simply correlated the client port and transaction ID. This behaviour is actually suggested by RFC1035.**
2. The second way, which was used for example by Jared Mauch, is to always query a unique name with each request. However, we argue that this might have adverse effects because if many forwarders relay to the same resolver, this resolver tries to cache all these unique names which might lead to unwanted cache evictions of legitimate names.

How does your controlled experiment with honeypots look like?

We implemented honeypots which actually mimic the behaviour of transparent forwarders. This was obviously quite challenging, because we had to deploy them at a network where we can ingest spoofed traffic to public resolvers.

We also had a second honeypot type which was simply a recursive resolver – this was our control sensor. And what we saw after some days is that the **control honeypot was detected by the major DNS scanning campaigns, however, the transparent forwarder was not detected.**

Why should we observe transparent forwarders?

Transparent forwarders are a major part of the ODNS ecosystem, they account for 26% of all the open components which trigger complete and correct DNS responses.

Transparent forwarders likely belong to domestic setups but interact with unsolicited, external requests, which might lead to impaired performance, security risks and even liability implications for its users.

Also, for us researchers, transparent forwarders can bias you measurement if you are completely unaware of them.

What is the influence on DNS scan results?

The large number of transparent forwarders has a direct influence on the scan results.

We identified countries with most ODNS components and compared our ranking with shadowserver. And the ranking of some countries even changed by 12 ranks.

But the detailed numbers don't really matter here, the main takeaway is simply that transparent forwarders can have a significant impact on your DNS measurements if you're unaware of them.

Why so many responses from the same source?

314k responses
from 8.8.8.8?

Although we perform single-packet scans, we receive multiple, valid responses from the same source address.

Retransmissions? No.

Ports and transaction IDs are different.

Spoofed responses? No.

Source and RR addresses (mostly) belong to same provider.

What are common DNS manipulations?

	Source of DNS Response	DNS Response Records
Transparent Interception	Expected (but spoofed)	<i>n/a</i>
Redirection	<i>n/a</i>	Incorrect (e.g. Ads)
Transparent Forwarding	Unexpected (real)	Correct