Fingerprint-based detection of DNS hijacks using RIPE Atlas

MAPRG Meeting, IETF 99

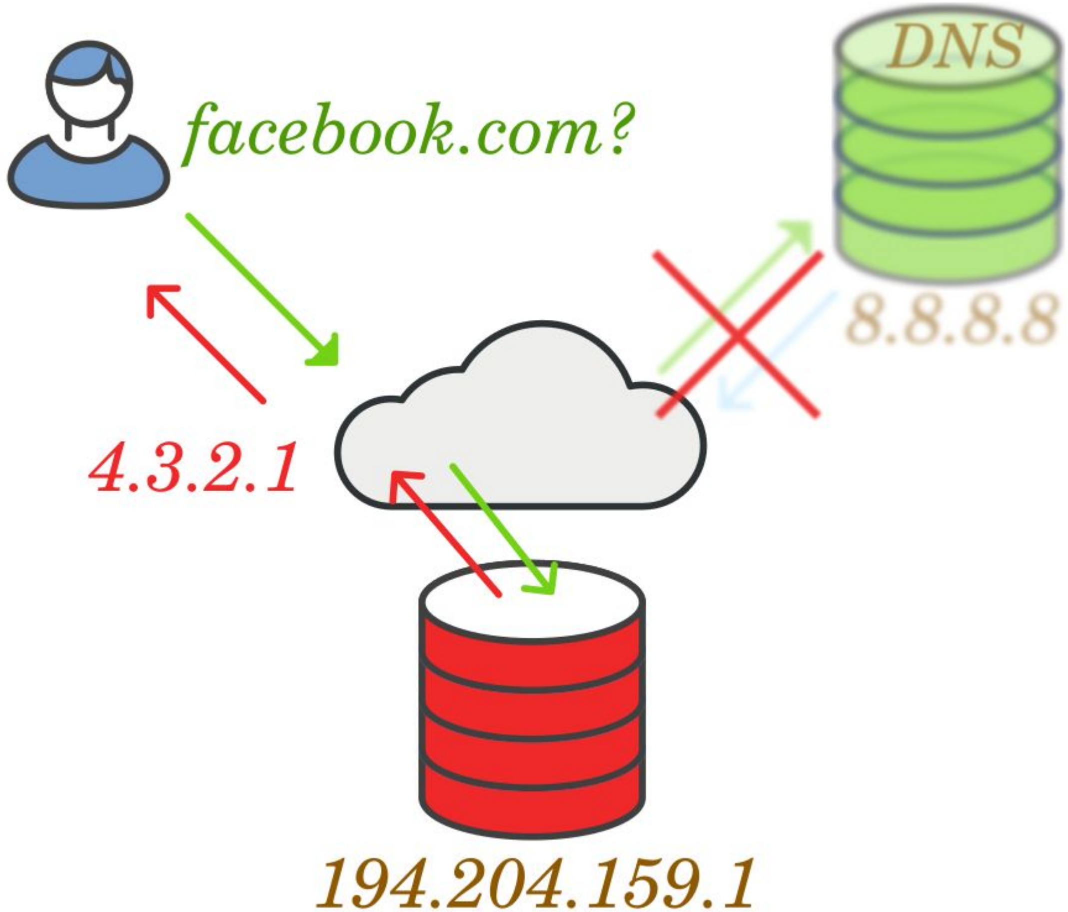
20th July 2017, Prague

Paweł Foremski
Farsight Security, Inc.
IITiS PAN

Maciej Andziński
NASK

FARSIGHT SECURITY IITiS NASK

DNS hijacks?

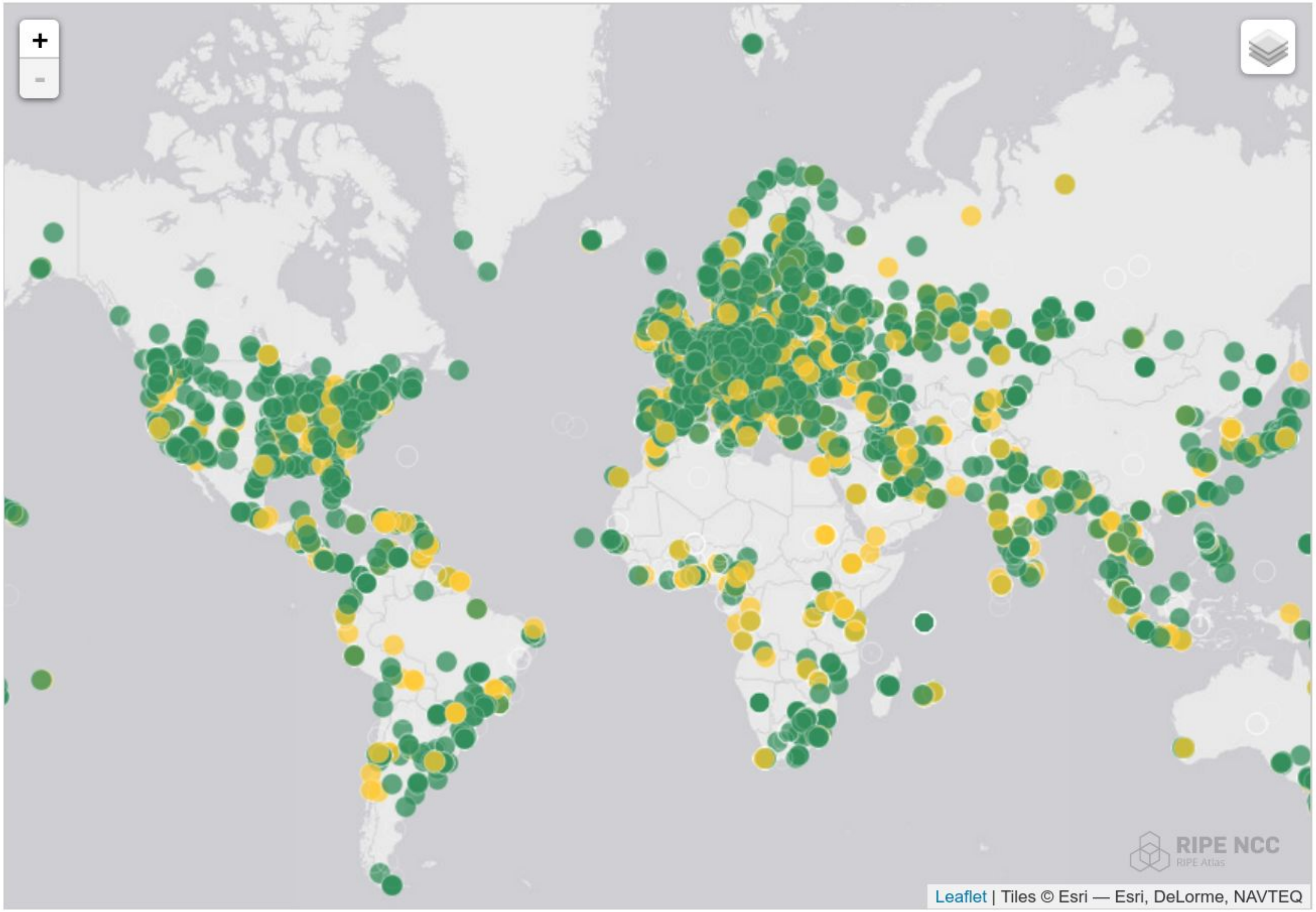


DNS hijack: you *think* Google answers your queries

DNS hijacks? #2

No hijacking: Google really answers your queries

RIPE Atlas?



An Internet measurement platform, ~10,000 probes

Research idea & goals

1. Send several DNS queries to 8.8.8.8 (*) -> *RIPE Atlas*
2. Rewrite DNS replies as feature vectors -> *fingerprinting*
3. Check if the fingerprints match the model -> *detection*

- Determine feasible fingerprint features
- Target Google Public DNS & Cisco OpenDNS
- How prevalent hijacking is - globally, per-country, per-AS?
- Who are the systematic hijackers?
- What does it all mean to the Internet?

Agenda

1. **Features of recursive servers**
2. **Measurements & tools**
3. **Establishing ground-truth**
4. **Classification using ML**
5. **Results**
6. **Conclusions**

Features

1. **RIPE Atlas provides a restricted API for DNS queries**
    a. Allows specifying the target server & some query parameters
    b. Provides low-level access to DNS replies (wire format)
    c. Measures timing

2. **CHAOS TXT queries**
    a. e.g. RFC4892, Requirements for a Mechanism Identifying a Name Server Instance
    b. **CH TXT hostname.bind** -> e.g. “cdns011.ovh.net” or… “who know”
    c. **CH TXT version.bind** -> e.g. “dnsmasq-2.76” or… “[SECURED]”
    d. **CH TXT id.server** -> e.g. “unbound.t72.ru” or… “go away”
    e. For each reply, store:
        i. response time & size
        ii. DNS header flags & rcode
        iii. rdata of first answer

Features #2

3. **DNSSEC support**
    a. see e.g. RFC4033 - RFC4035
    b. **IN A dnssec-failed.org** -> should fail
    c. **IN DNSKEY pl.** -> must not fail

4. **IPv6 support**
    a. Query for a zone hosted on an IPv6-only auth NS
    b. **IN AAAA ds.v6ns.test-ipv6.ams.vr.org** -> should not fail
        i. v6ns.test-ipv6.ams.vr.org. 1800 IN NS v6ns1.test-ipv6.ams.vr.org.
        ii. v6ns1.test-ipv6.ams.vr.org. 1670 IN AAAA2607:f740:d::f77

5. **TCP support**
    a. Force RIPE Atlas to do query over TCP
    b. **IN A facebook.com / TCP** -> should not fail

Features #3

6. **Replies to non-existent domains**
    a. **IN A <timestamp>.<probe-id>.surely1does2not3exist4.com**
    b. Each RIPE Atlas probe prepends its own unique label
    c. Should return rcode 3, NXDOMAIN
    d. If the query is successful (rcode 0), store:
        i. The IP address returned
        ii. AS number & network name

7. **Qname letter case (in-)sensitivity**
    a. **IN A FaCeBoOk.cOm**
    b. Should return the same letter case
    c. If the rname in the answer doesn't match, mark as failure

Features #4

8. **Round-trip time**
    a. Measure the minimum ICMP ping RTT to the resolver

9. **Traceroute**
    a. **Send an ICMP traceroute to the resolver**
    b. Filter out private IP addr space
    c. Store: hop count, ASPATH length, parameters of the exit AS (RTT, ASN, network)

10. **Two independent “who am I?” services:**
    a. **IN A whoami.akamai.com**
    b. **IN TXT test.ipv4.google-pdns-info.andzinski.pl**
    c. An auth server that replies with the resolver IP address
    d. Store: returned IP address, it’s ASN and network name

Measurements & tools

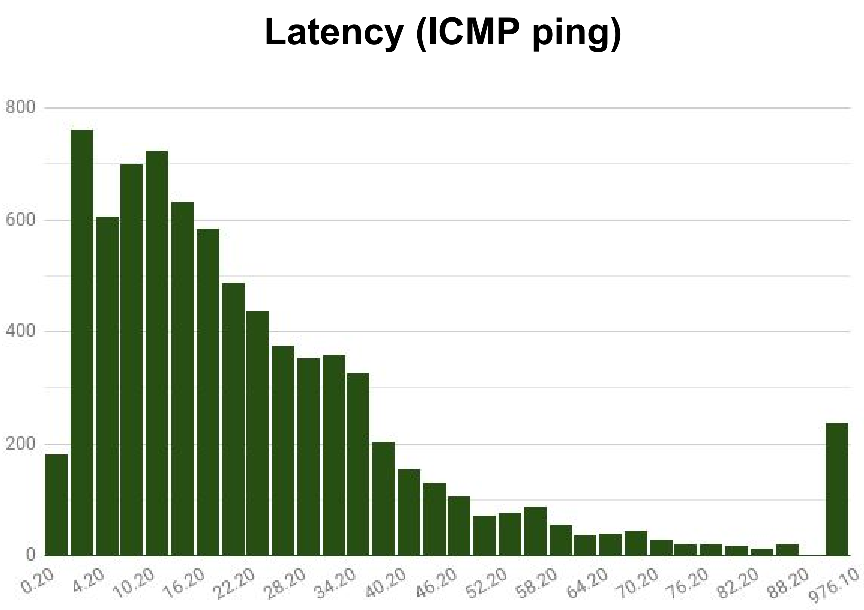
- **Run in June 2017 using 9,790 RIPE Atlas probes (3K ASes)**
  - ...burned a few million RIPE Atlas credits - thanks Vesna & Stephen! ;-)
  - tools published at https://github.com/recdnsfp/measurements
  - parsers at https://github.com/recdnsfp/parsejson

- **Google (8.8.8.8)**
  - Raw: https://github.com/recdnsfp/measurements/tree/master/datasets/google
  - Spreadsheet: https://goo.gl/LSXSjW

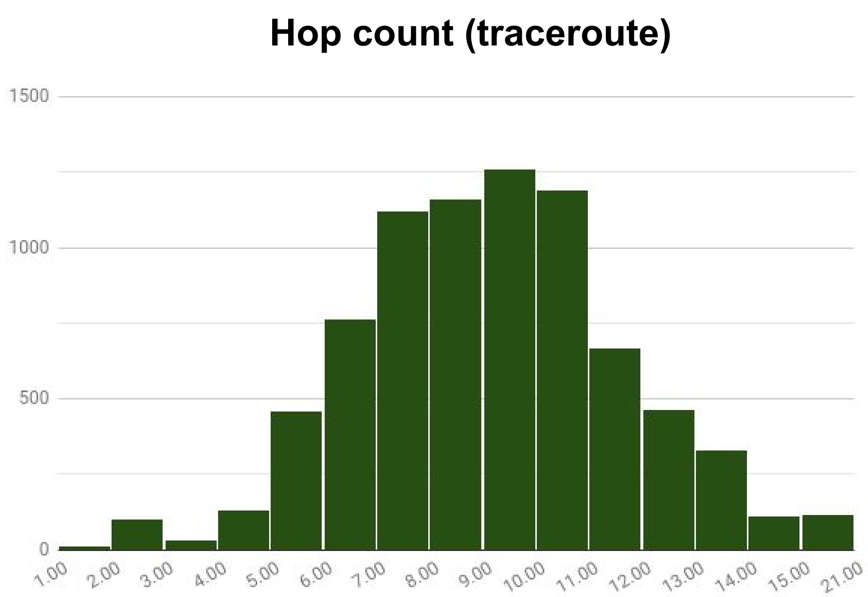
- **OpenDNS (208.67.222.222)**
  - https://github.com/recdnsfp/measurements/tree/master/datasets/opendns
  - Spreadsheet: https://goo.gl/9MEhnx

- **Bonus**: default probe resolvers
  - https://github.com/recdnsfp/measurements/tree/master/datasets/probes
  - Spreadsheet: https://goo.gl/GCZ4Xu

Measurements: Google Public DNS

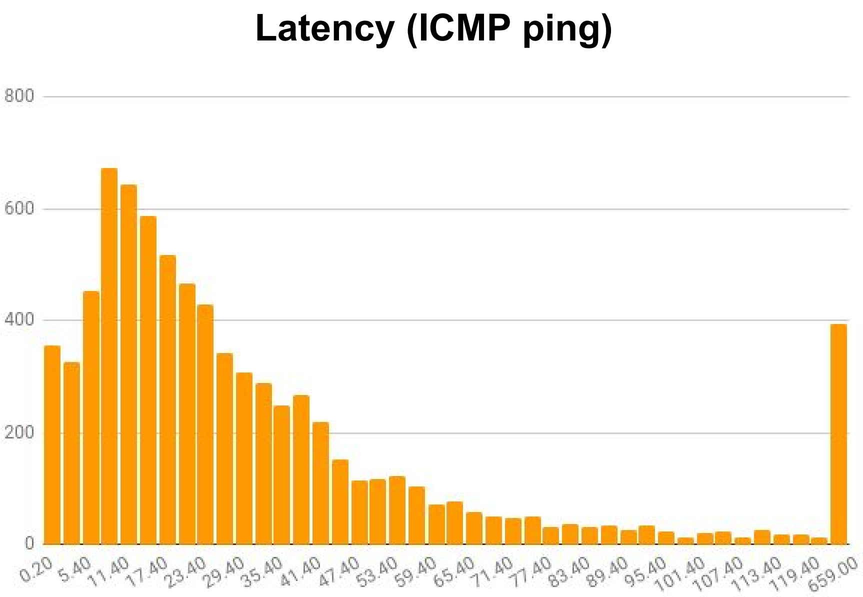


Median: 17.8 msec

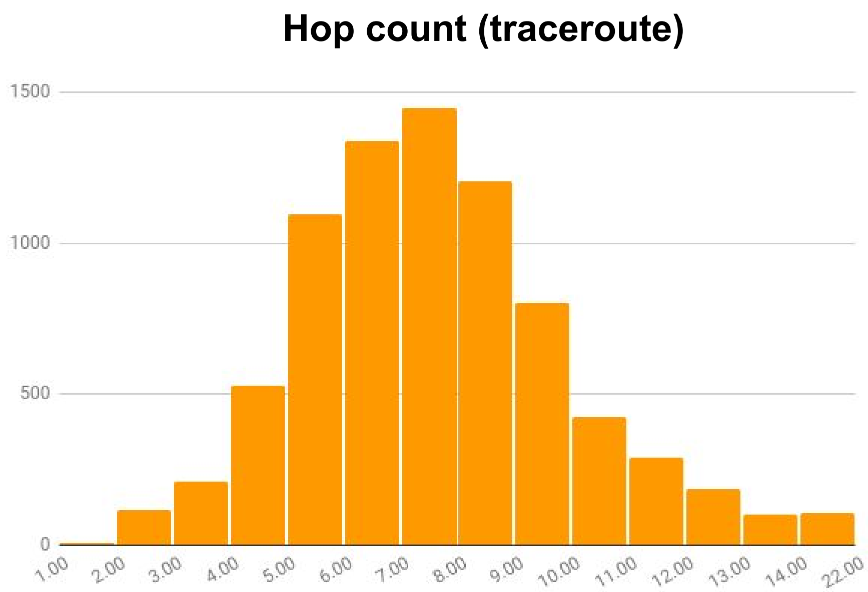


Median: 9 hops

Measurements: Cisco OpenDNS



Median: 22.6 msec



Median: 7 hops

Measurements: probe resolvers

Resolver network,
as seen by whoami.akamai.com

#	Network	Count	Percentage
1	GOOGLE	1,857	21.63%
2	OPENDNS	351	4.09%
	+ DIRECT_MEDIA	*31*	*0.36%*
3	LIBERTY_GLOBAL_OPERATIONS	234	2.73%
4	DEUTSCHE_TELEKOM	222	2.59%
5	COMCAST_CABLE_COMMUNICATIONS	212	2.47%
6	ORANGE	147	1.71%
7	FREE_SAS	115	1.34%
8	XS4ALL_INTERNET_BV	65	0.76%
9	BRITISH_TELECOMMUNICATIONS_PLC	65	0.76%
10	MCI_COMMUNICATIONS	61	0.71%
	Other / N/A:	5,224	60.86%

Ground-truth

- **No way to obtain from network operators**
- **Assume the most common fingerprint as "legitimate"**
- **Assume any deviations in the following as "hijacked":**
  - **whoami & whoami2 network** == expected network
  - **ipv6-only rcode** == SUCCESS
  - **qname letter case** == OK
  - **TCP rcode** == SUCCESS
  - **CH hostname & version & serverid rcode** != SUCCESS
  - **dnssec fail rcode** != SUCCESS
  - **nxdomain rcode** != SUCCESS
- **Hope that ML will pick up all features (40+) & learn how to use them**
- **Working on an improved, statistics-based ground-truth method**

Classification using ML

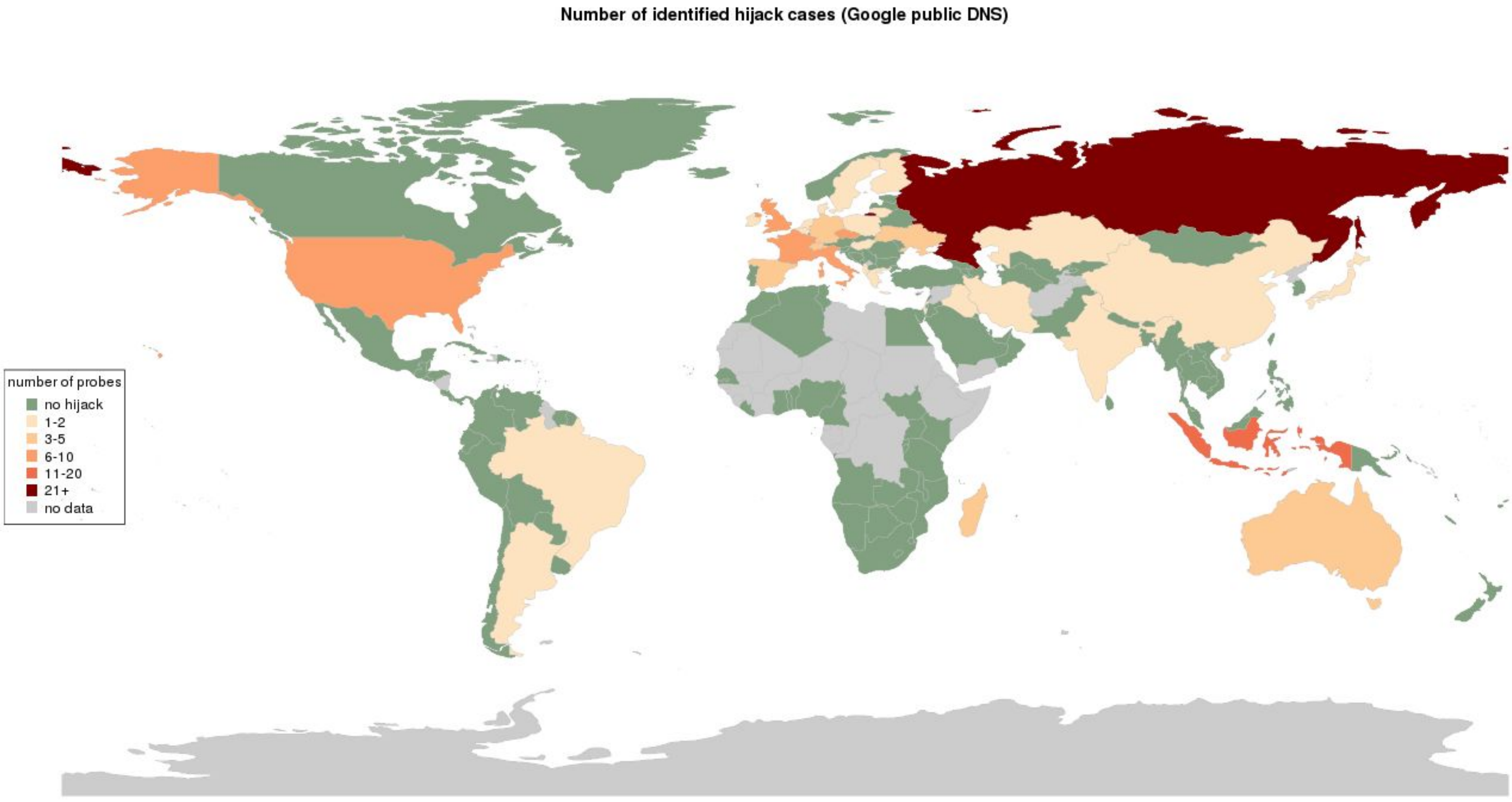
1. **Randomly sample 50 "legitimate" vs. 50 "hijacked" probes**
   a. Evenly split into training and testing subsets -> evaluate
   b. Build the target classifier using full training + testing

2. **Evaluate the classification performance:**

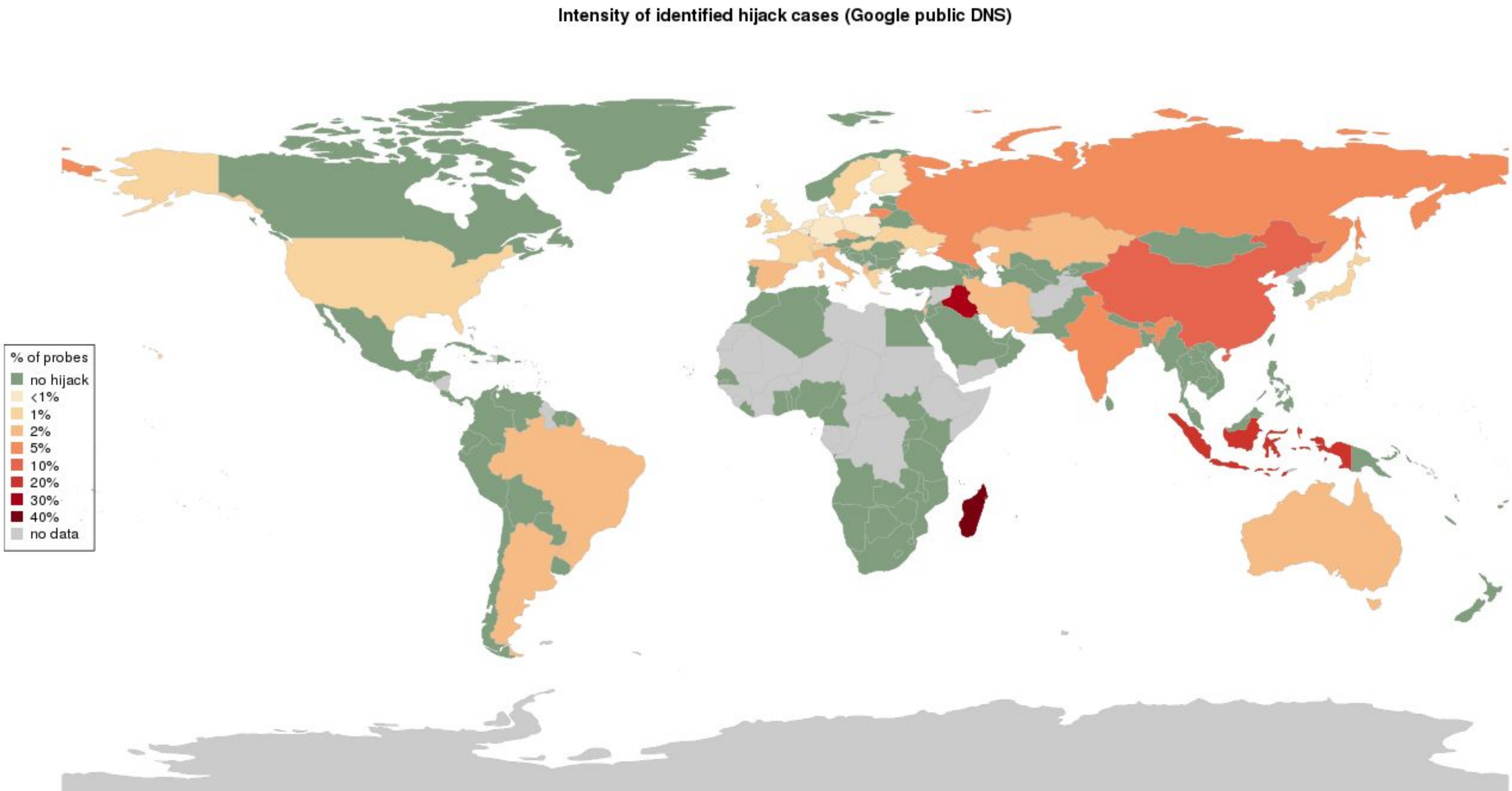
	Google (x 30)			OpenDNS (x 30)		
	Accuracy	**%FP**	**%FN**	**Accuracy**	**%FP**	**%FN**
k-NN (n = 3)	78.11%	6.29%	15.60%	81.44%	0.60%	17.97%
Decision Tree (CART)	92.82%	0.97%	6.22%	93.56%	1.14%	5.30%
Random Forest (n = 10)	93.84%	0.00%	6.16%	93.50%	0.25%	6.25%

3. **Classify the rest of data using Random Forest classifier**
   a. Implementation at https://github.com/recdnsfp/classify

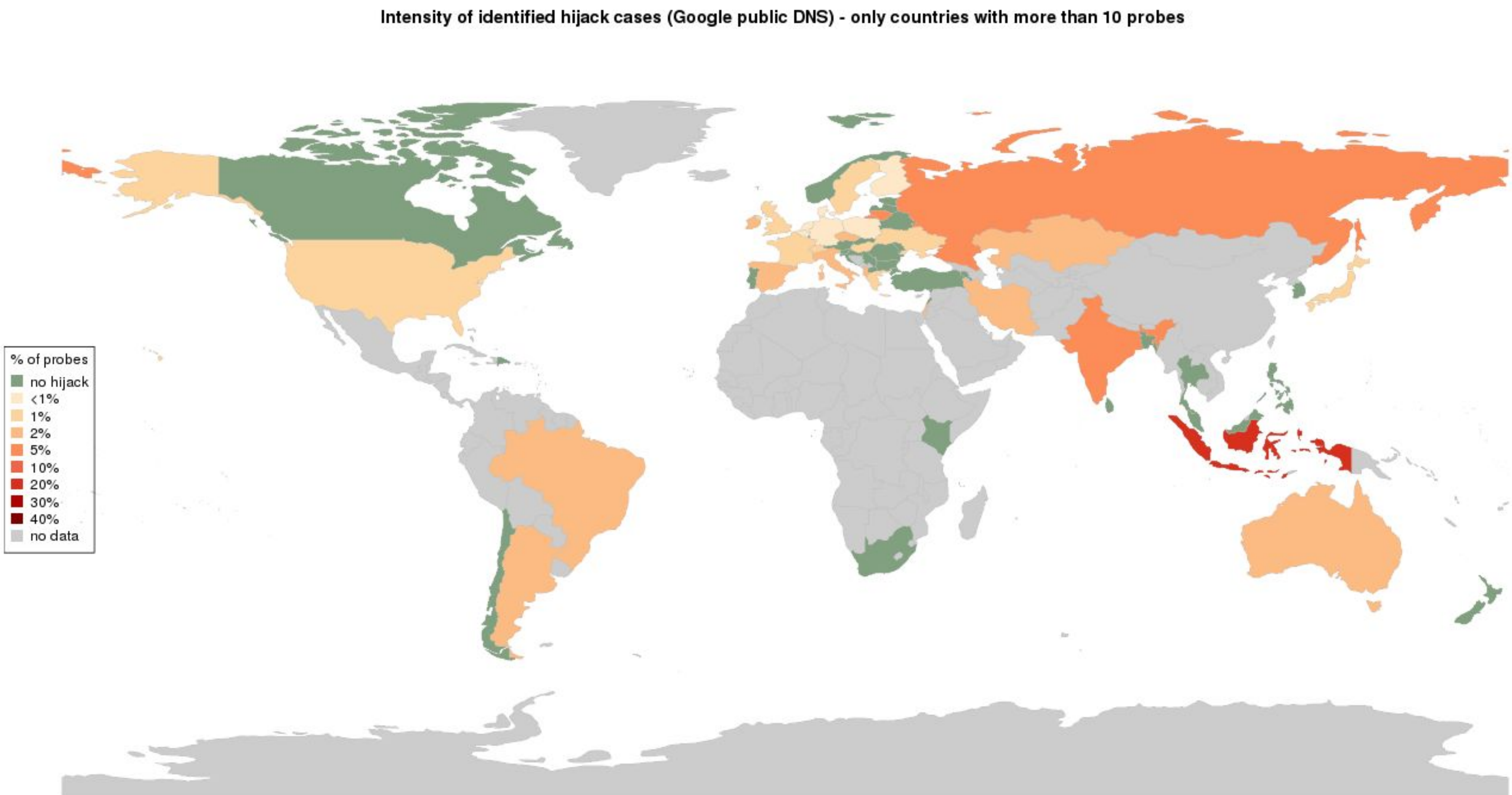
Results: Google DNS hijacks (120 = 1.54% globally)



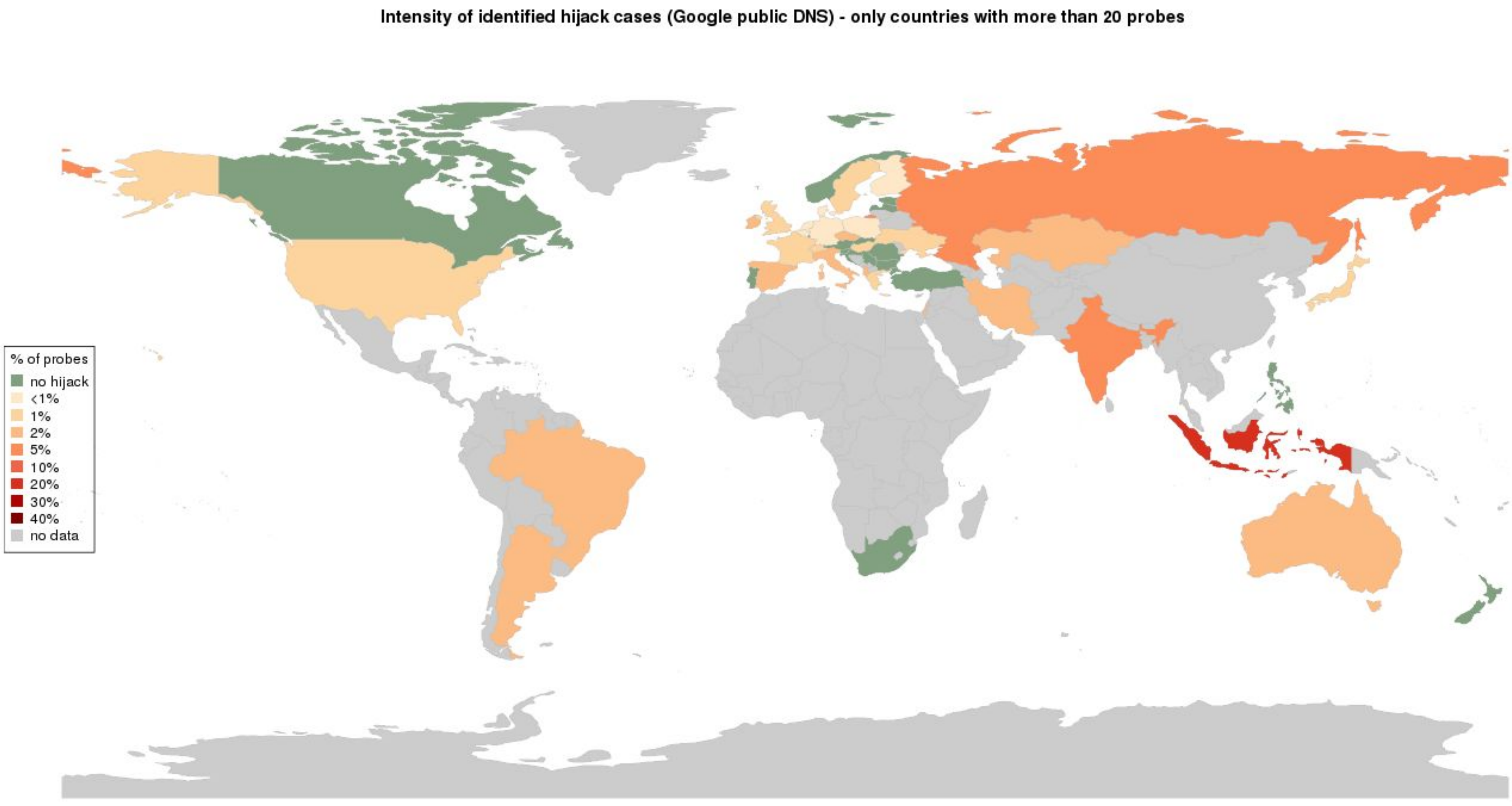
Results: Google DNS hijacks (%)



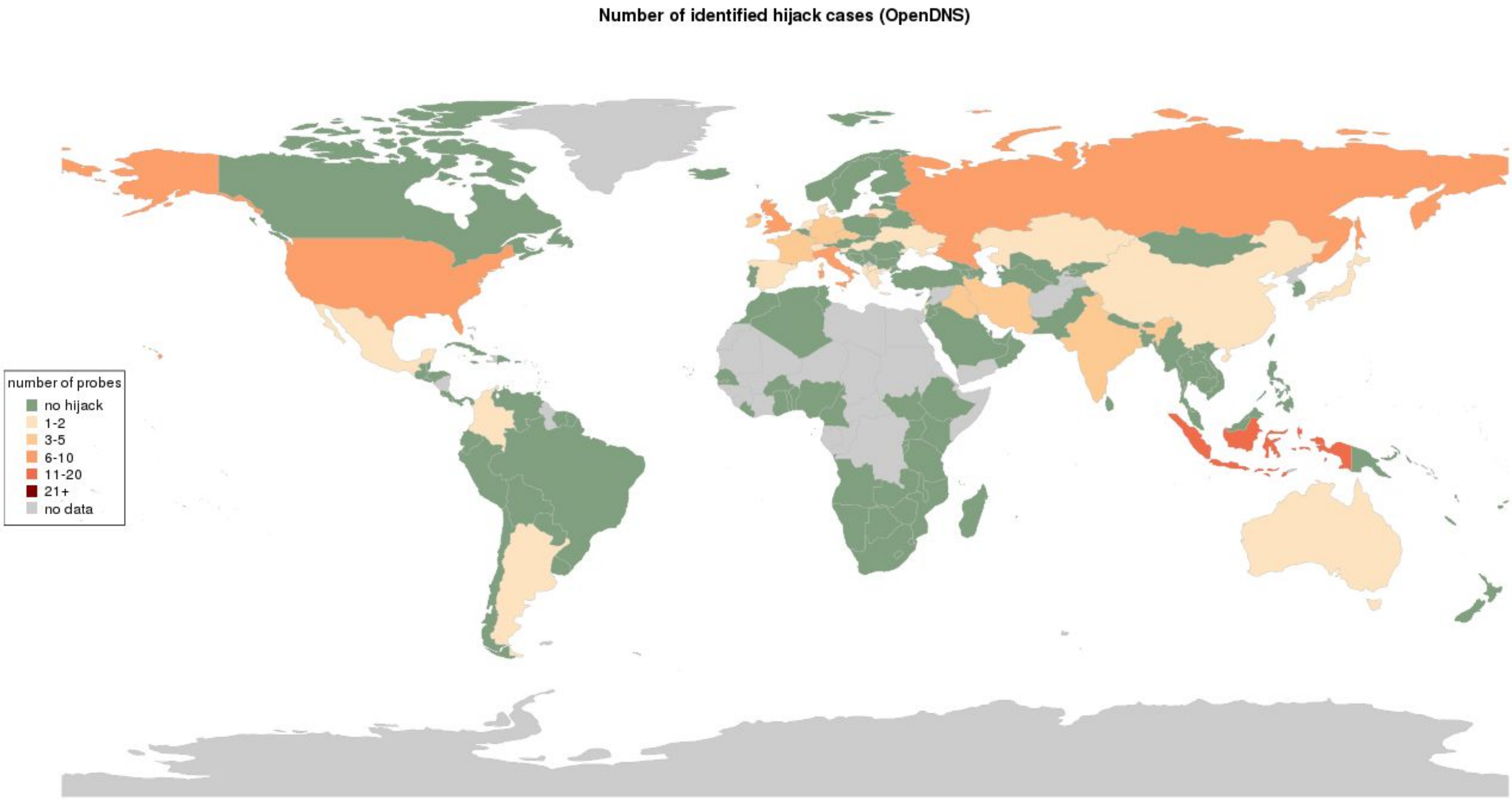
Results: Google DNS hijacks (% for >10 probes)



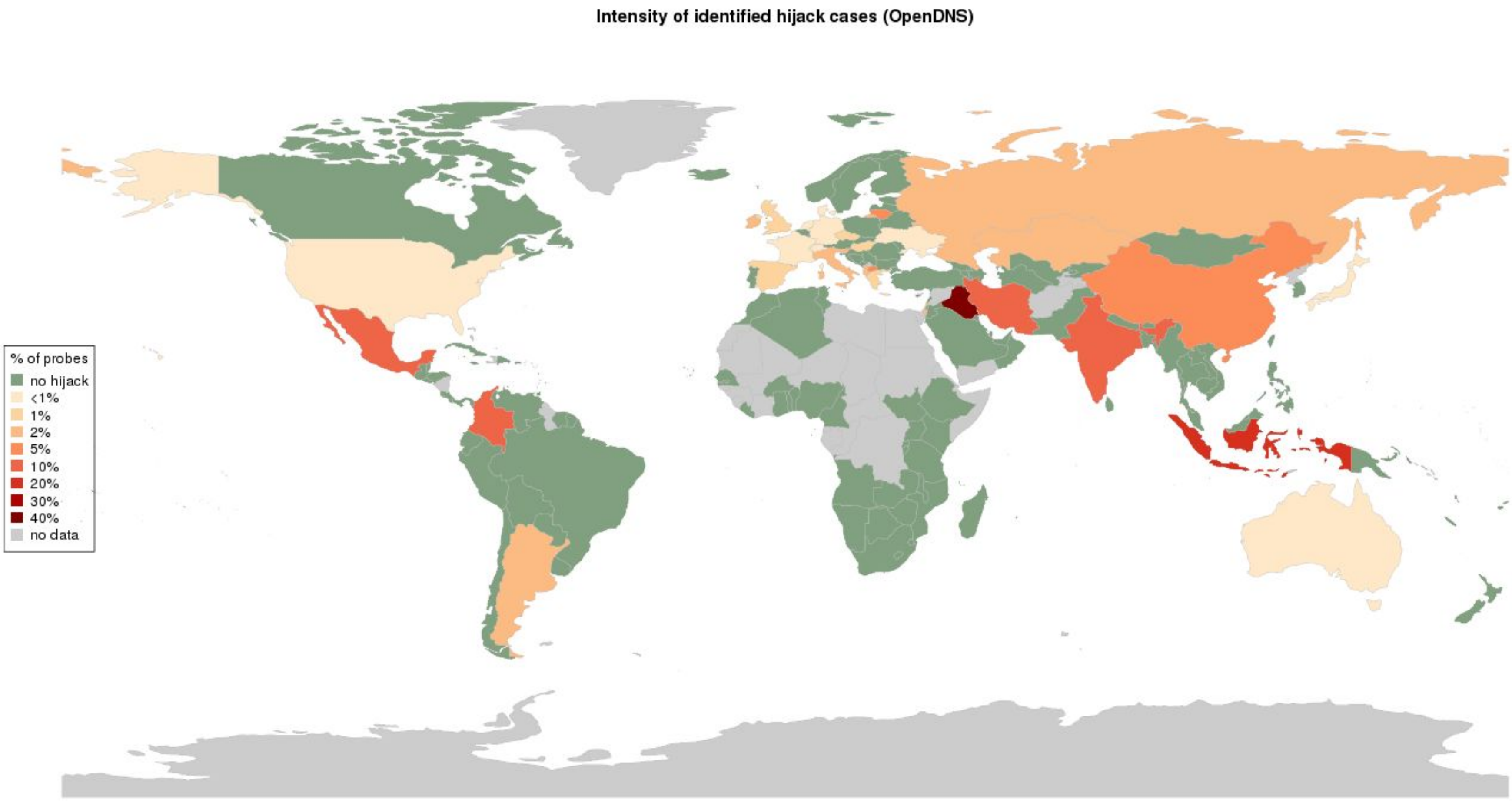
Results: Google DNS hijacks (% for >20 probes)



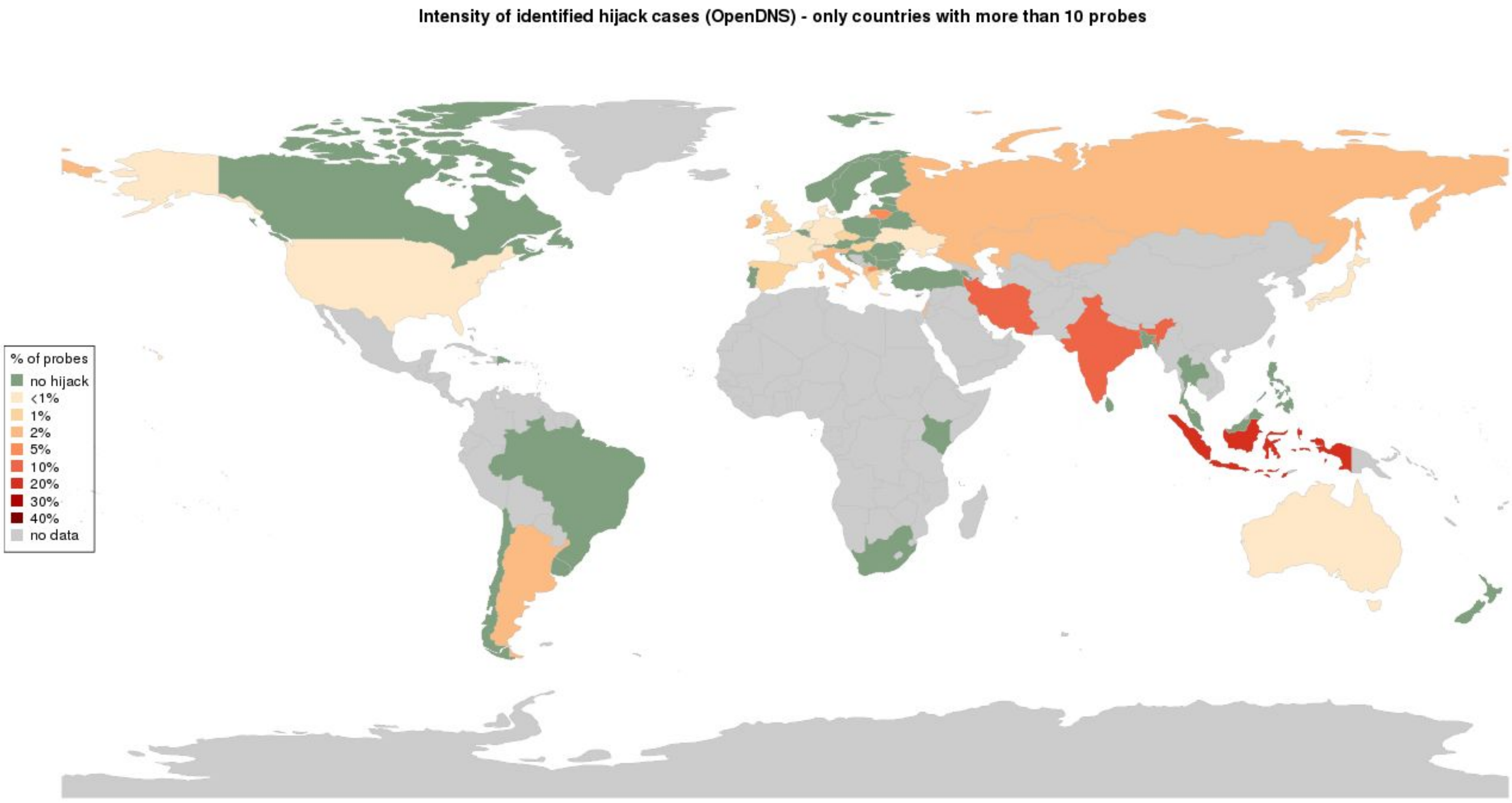
Results: OpenDNS hijacks (94 = 1.22% globally)



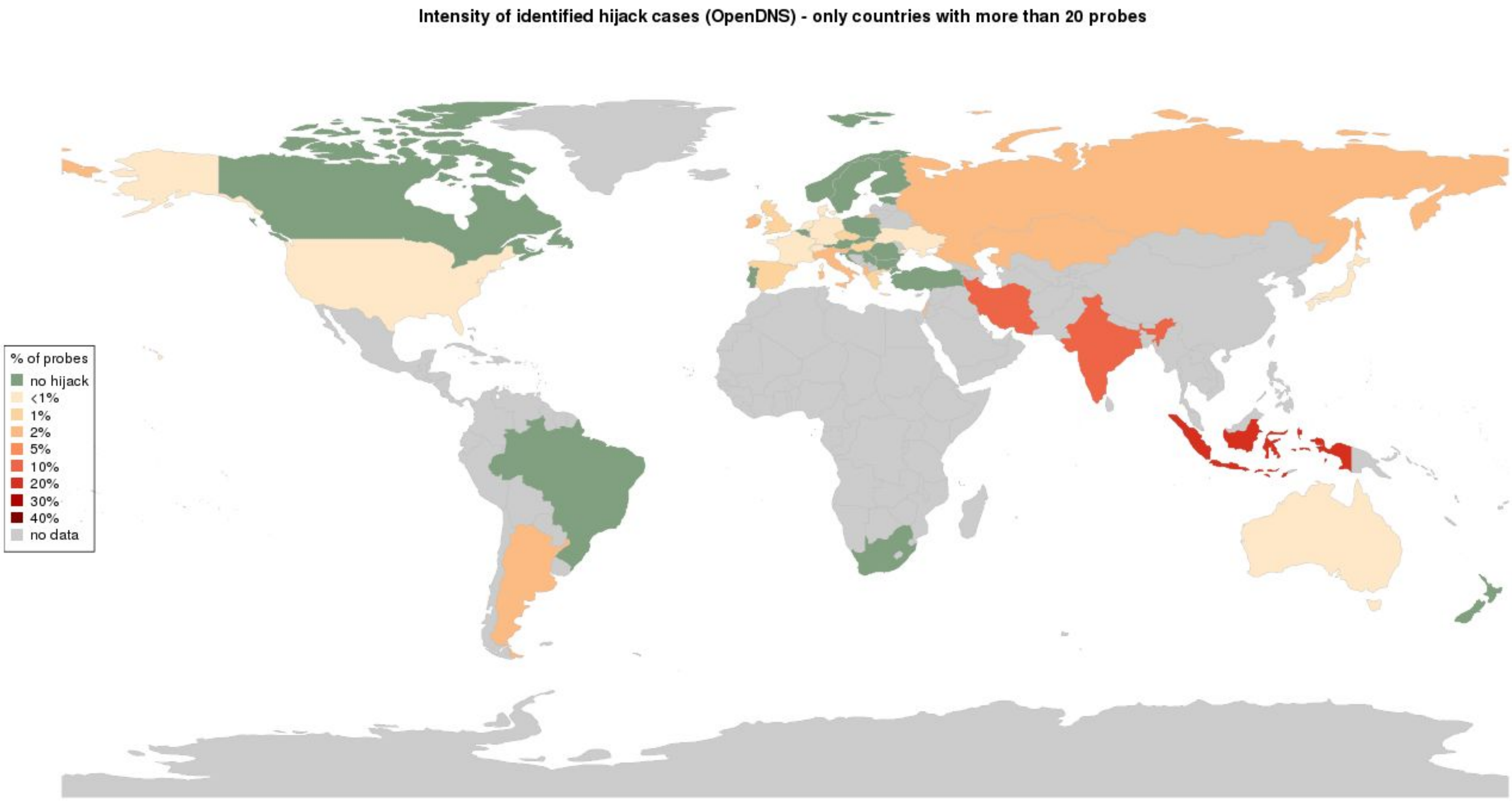
Results: OpenDNS hijacks (%)



Results: OpenDNS hijacks (% for >10 probes)



Results: OpenDNS hijacks (% for >20 probes)



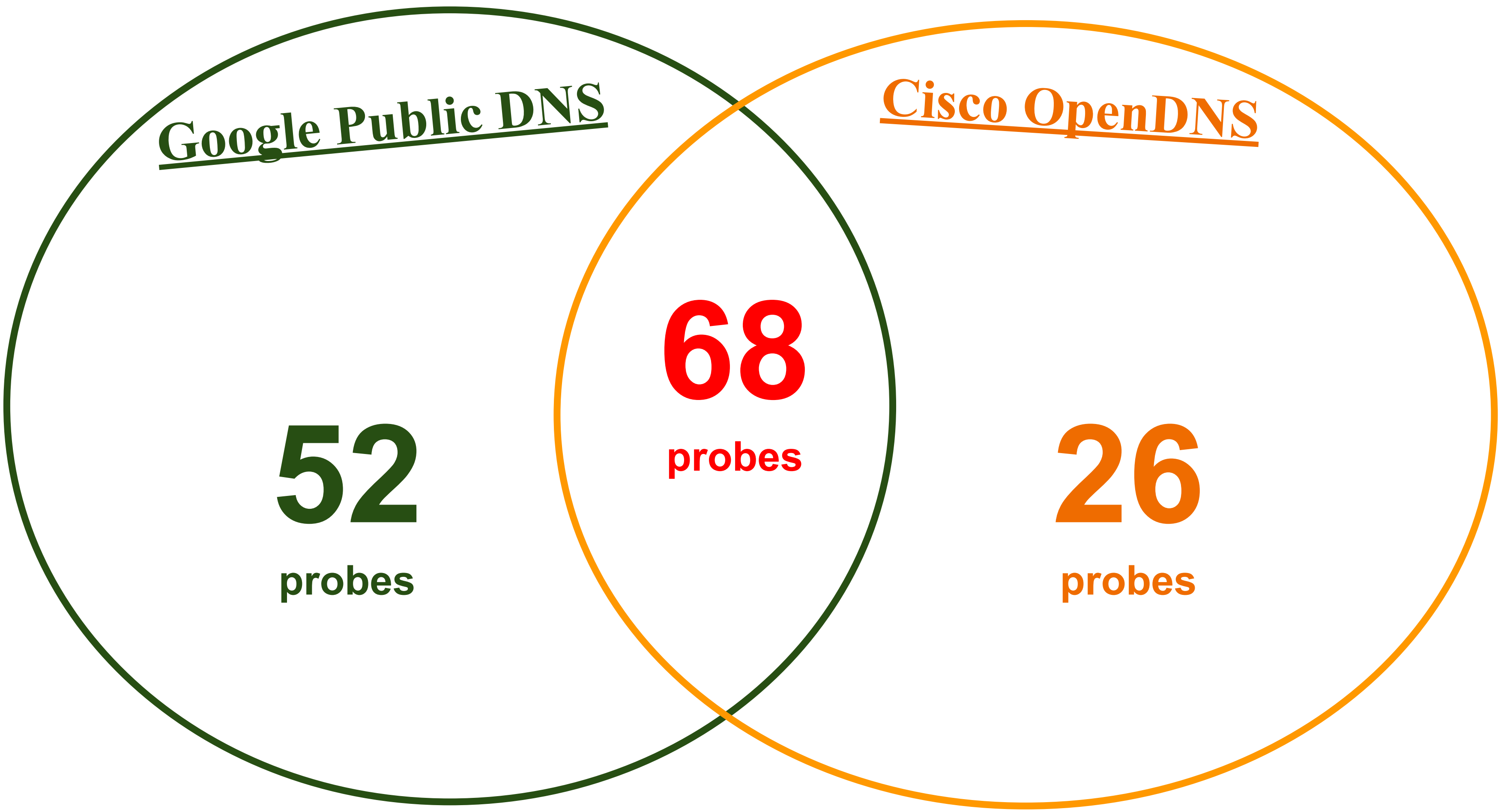
Results: Google hijacks per ASN

#	Network	ASN	Count	% Total	% for ASN
1	BRITISH_TELECOMMUNICATIONS_PLC	AS2856	6	5.00%	8.96%
2	VODAFONE_ITALIA_SPA	AS30722	5	4.17%	62.50%
3	COMCAST_CABLE_COMMUNICATIONS_LLC	AS7922	4	3.33%	1.35%
4	LIBERTY_GLOBAL_OPERATIONS_BV	AS6830	4	3.33%	1.63%
5	UNARTEL_SRO	AS198977	4	3.33%	80.00%
6	PT_TELEKOMUNIKASI_INDONESIA	AS17974	4	3.33%	80.00%
7	CLOSED_JOINT_STOCK_COMPANY_TRANSTELECOM	AS47313	2	1.67%	100.00%
8	IRENALA	AS37608	2	1.67%	100.00%
9	ABSOLIGHT	AS29608	2	1.67%	100.00%
10	BREDBAND2_AB	AS29518	2	1.67%	40.00%
	Other		85	70.83%	

Results: OpenDNS hijacks per ASN

#	Network	ASN	Count	% Total	% for ASN
1	BRITISH_TELECOMMUNICATIONS_PLC	AS2856	6	6.38%	9.52%
2	VODAFONE_ITALIA_SPA	AS30722	5	5.32%	62.50%
3	PT_TELEKOMUNIKASI_INDONESIA	AS17974	4	4.26%	80.00%
4	COMCAST_CABLE_COMMUNICATIONS_LLC	AS7922	3	3.19%	1.02%
5	LIBERTY_GLOBAL_OPERATIONS_BV	AS6830	2	2.13%	0.82%
6	TELECOMMUNICATION_INFRASTRUCTURE_COMPANY	AS48159	2	2.13%	100.00%
7	SKYLOGIC_SPA	AS29286	2	2.13%	100.00%
8	FREE_SAS	AS12322	2	2.13%	1.36%
9	JASA_TERPADU_TELEMATIKA_JASATEL	AS9785	1	1.06%	100.00%
10	TOKYO_INSTITUTE_OF_TECHNOLOGY	AS9367	1	1.06%	100.00%
	Other		66	70.21%	

Results: who are the systematic hijackers?



Results: who are the systematic hijackers?

1. **Analyze the probes with both Google & OpenDNS hijacked**

2. **Drop incidental observations with less than 3 probes hijacked**

Results:

1. AS 17974, Telkom Indonesia: 4 out of 6 (66.7%)
2. AS 30722, Vodafone Italy: 5 out of 9 (55.6%)
3. AS 2856, British Telecommunications: 5 out of 88 (7.4%)

Conclusions

- **DNS hijacking is a real thing happening on the Internet**
  - We found ~100 RIPE Atlas probes with hijacked DNS for the two biggest providers
  - Some regions have >25% chances of DNS being hijacked
  - Globally, there's >1% probability on average
- **The risk does not necessarily come from a state actor**
  - We found Autonomous Systems that seem to have a policy of DNS hijacking
  - Many hijacks in developed countries (e.g. US, UK, Italy)
  - Probably many motivations...
- **No big difference for Google DNS vs. OpenDNS**
  - Just switching the resolver IP will not help
- **The Internet absolutely needs a more secure DNS**
  - Hijacking opens endless possibilities for manipulation & surveillance
  - We need to secure the stub vs. recursive resolver path

Thank You!

Paweł Foremski

pjf@fsi.io

@pforemski

Maciej Andziński

maciej.andzinski@nask.pl

@MaciejAndzinski

https://github.com/recdnsfp

Acknowledgements & Thanks:

- Mateusz Kaczanowski (Facebook)
- Vesna Manojlovic (RIPE NCC)
- Stephen D. Strowes (RIPE NCC)