Mirrors in the Sky: On the Potential of Clouds in DNS Reflection-based Denial-of-Service Attacks

Ramin Yazdani
University of Twente

Alden Hilton
Brigham Young University

Jeroen van der Ham
University of Twente

Roland van Rijswijk-Deij
University of Twente

Casey Deccio
Brigham Young University

Anna Sperotto
University of Twente

Mattijs Jonker
University of Twente

ABSTRACT

Clouds are likely to be well-provisioned in terms of network capacity by design. The rapid growth of cloud-based services means an increased availability of network infrastructure for all types of customers. However, it could also provide attackers opportunity to misuse cloud infrastructure to bring about attacks, or to target the cloud infrastructure itself.

In this paper we study, focusing on DNS-based reflection DDoS attacks, how cloud networks can be misused to carry out attacks, with possible consequences for the internal cloud infrastructure itself. A straightforward way to misuse cloud infrastructure would be to host open DNS resolvers in the cloud – a phenomenon that we quantify in the paper. More importantly, we structurally analyze how the internal DNS infrastructure of a cloud can be misused. The novelty of this paper lies in identifying and formalizing six attack models for how DNS cloud infrastructure can be abused to bring about reflection attacks, and testing these increasingly complex and progressively specific models against real cloud providers.

Our findings reveal that a steady average of 12% of open DNS resolvers are hosted in cloud or datacenter networks, which gives them well-provisioned network access. Much more worryingly, our results reveal that a number of providers, several of which among market leaders, expose parts of their DNS infrastructure to outsiders, allowing abuse against a provider's infrastructure, its customers, as well as hosts in external networks. In the course of our study, we responsibly disclosed our findings to these providers.

CCS CONCEPTS

• **Security and privacy → Denial-of-service attacks**.

KEYWORDS

DDoS, DNS-based reflection, cloud networks, spoofing

ACM Reference Format:
Ramin Yazdani, Alden Hilton, Jeroen van der Ham, Roland van Rijswijk-Deij, Casey Deccio, Anna Sperotto, and Mattijs Jonker. 2022. Mirrors in the Sky: On the Potential of Clouds in DNS Reflection-based Denial-of-Service Attacks. In *25th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2022), October 26–28, 2022, Limassol, Cyprus.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3545948.3545959

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike International 4.0 License.

RAID '22, October 26–28, 2022, Limassol, Cyprus
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9704-9/22/10.
https://doi.org/10.1145/3545948.3545959

1 INTRODUCTION

Cloud networks have become the de facto go-to place for outsourcing services and infrastructure, often motivated by easier management and better security. For sure, one of the many intrinsic benefits of moving to the cloud is that a service will have, by design, a well-provisioned infrastructure in terms of computational power and network capacity. Such resource richness, however, means that the cloud is attractive not only to regular customers, but to attackers as well.

In this paper we study how cloud networks can be misused to carry out attacks, with possible consequences for the internal cloud infrastructure itself. We focus in specific on Domain Name System (DNS)-based Reflection & Amplification (R&A) Distributed Denial of Service (DDoS) attacks, as these are continually used to significantly disrupt Internet services and networks.

The interplay between clouds and DNS-based reflection attacks brings forward the following observations. A straightforward way to misuse cloud infrastructure would be to host an open DNS resolver in the cloud itself and we postulate that the link capacity of such a reflector decisively affects its ability to deliver attack traffic at a high rate. For this reason, as a first step in our research, we quantify, at scale, how prominently open DNS resolvers are hosted in clouds. Clouds, however, are not only a group of hosts, but they have complex internal infrastructure that may also include dedicated DNS servers. A more pressing question then becomes: could the internal DNS infrastructure of a cloud also be misused in reflection-based DDoS attacks? As we will reveal in this paper, the list of R&A DDoS reflectors (typically open resolvers and authoritative nameserver) can be extended with *improperly shielded* (and likely highly-provisioned) DNS infrastructure specific to cloud providers, which by design should serve exclusively the clients of their own network. Such DNS servers may be misused to bring about DDoS attacks and could themselves experience disruption as a consequence of such misuse.

The novelty of our paper is that we structurally analyze how the internal DNS cloud infrastructure can be misused in R&A DDoS attacks. We identify and formalize six attack models – of increasing complexity and progressively specific to cloud infrastructure. We assess the feasibility of attacks by conducting a proof-of-concept

study on 19 leading, public cloud providers. Furthermore, by fusing longitudinal open DNS resolver scans with IP intelligence data, we investigate the contribution of different types of networks in hosting open resolvers.

The main contributions of our paper are that we:

- Conduct an Internet-wide study to classify open DNS resolvers based on their hosting network. By fusing longitudinal open DNS resolver scans with IP intelligence data, we extend and strengthen observations made in related work, and show that hundreds of thousands of resolvers exist in well-provisioned networks.
- Identify and formalize different attack models to misuse cloud infrastructure to bring about R&A DDoS attacks – both from external networks and by customers of the cloud provider.
- Demonstrate through a proof-of-concept study that some of the attack models are currently feasible for a number of leading cloud providers, whom we have notified of this threat and engaged with during a coordinated disclosure process.

The remainder of this paper is structured as follows. In Section 2 we provide background information. We identify and formalize cloud-based R&A attack models in Section 3. We detail our methodology to investigate the attack models in Section 4. In Section 5 we present our results. Defense mechanisms are given in Section 6. We provide a discussion about our study as well as ethical considerations in Sections 7 and 8. We discuss related work in Section 9. Finally, we conclude our paper in Section 10.

2 BACKGROUND

In this section, we provide background information on reflection and amplification attacks, common mitigation practices, and the typical organization of cloud DNS infrastructures.

2.1 Reflection and Amplification DDoS

Due to the connectionless nature of UDP, higher-layer protocols that run on top of UDP may be unaware of source IP address spoofing. That is, attackers can send spoofed queries to so-called reflectors to trigger them to send responses to the intended victim instead. This is referred to as *reflection*. If the responses are also larger than the requests, the attack traffic is *amplified*. In concert, these concepts can be used to bring about powerful reflection and amplification attacks (R&A). A number of protocols are vulnerable to this type of misuse, for example DNS, NTP and SNMP [16, 30, 32].

In this paper we investigate how various DNS implementations in highly-provisioned networks, i.e., clouds, can be misused to instrument reflection-based DDoS attacks. Our study extends the traditional DNS-based R&A scenario in which only open resolvers are misused. We show that improperly shielded closed DNS servers can also contribute to reflection-based attacks or may even themselves become the target of attacks.

2.2 Existing Mitigation Practices

Various mechanisms exist on the Internet that were developed to address problems arising from source IP address spoofing and to reduce the impact of R&A DDoS attacks. In the following we elaborate on three of such methods.

2.2.1 Origin Side Ingress Filtering. Network ingress filtering, which is documented in RFC 2827 [10], also known as Best Current Practice (BCP) 38 and Source Address Validation (SAV), is a filtering mechanism that can be deployed on the edge router of a network to check the source IP address of packets that aim to leave the network. This is possible as the legitimate source addresses of clients in the network are typically known to the operator and any illegitimate traffic can be dropped closest to its origin, thus prohibiting a reflection attack from being conducted. In the rest of this paper we refer to this mechanism as Origin-side Source Address Validation (OSAV). For more complicated networks, proposals to deploy such a filtering are given in RFC 3704 also known as BCP 84 [4]. In an ideal situation where all networks implement such filtering, there would be no reflection-based DDoS potential (except for attacks with both an origin and a destination inside a single network). However, previous studies [5, 23] reveal that many networks exist that do not implement BCP 38. One of the main reasons for this partial adoption is a lack of incentives, as spoofed queries originating from a network do not harm the origin network but rather other networks.

2.2.2 Destination Side Ingress Filtering. Filtering largely similar to that discussed in Section 2.2.1 can also be implemented on the destination side. In this way, an edge router can control the traffic that is to enter the network and drop packets that supposedly originated from within the network itself. We further refer to this as Destination-side Source Address Validation (DSAV). As we will explain and formalize in Section 3, a lack of destination-side ingress filtering can enable reflection-based DDoS attacks. Another security threat of such queries successfully entering a network is DNS cache poisoning [17]. Unlike is the case with OSAV deployment, DSAV deployment directly benefits the deploying network operator. Surprisingly, measurements of the Spoofer project [5] show that DSAV sees less deployment than OSAV, whilst one would intuitively expect it to have a wider deployment than OSAV.

2.2.3 Response Rate Limiting. Response Rate Limiting (RRL) [33, 34] is a technique that can be used, mainly on authoritative nameservers, to reduce the impact of DDoS attacks. Authoritative nameservers that deploy RRL limit sending responses for queries generated from the same IP address block that result in the same answer. This method is based on the concept of a token bucket. A DNS server takes out a token each time a response is sent to a client for a DNS record. On the other hand, while time passes, additional tokens are added for that client/record. Once the token bucket is empty (due to frequent queries), the server will send truncated responses to the client. A legitimate client may fall back to TCP, while malicious requests will be stopped short. Although this method can substantially reduce the impact of a DDoS attack, it has its own limitations. If query names are distributed such that they get different answers, this rate limiting becomes less effective. Moreover, distributing query origins among several subnets can further counteract RRL.

Technically it is possible to configure recursive resolvers to rate limit responses in a similar way to authoritative nameservers. However, due to the lack of caching by most applications, RRL is typically not meant to be deployed on recursive nameservers [34].

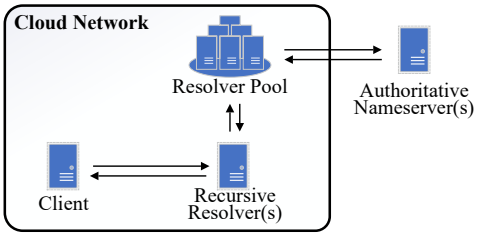

Figure 1: Elements in a DNS resolution process in a cloud

2.3 DNS Infrastructure of Clouds

Similar to most hosts connected to the Internet, hosts running in cloud networks need to resolve domain names for various purposes. In a DNS resolution process, the stub resolver running on the client operating system queries a recursive resolver/forwarder, which then either directly – or via a pool of upstream resolvers – contacts the authoritative nameservers and finally returns an answer to the client. In the rest of this paper we refer to the recursive resolver/forwarder as a "recursive resolver". By default, a DNS service is provided to clients in any network, and cloud networks are no exception. Cloud providers rely on different implementations to deliver this service, but in general, this involves two main elements: one or multiple recursive resolvers; and a pool of upstream resolvers Figure 1 depicts such a deployment. We have based this argument on DNS deployment documentations of a couple of large cloud providers [2, 13]. Note that the recursive resolver(s) and the resolver pool can be outside a cloud network when the DNS resolution process is outsourced to other providers. We are aware that some networks use a different DNS resolution process than our simplified model; however, our experiments further validate that this model represents the DNS resolution process in majority of cloud networks. The recursive resolvers are directly contacted by the client and then they forward the incoming queries to an upstream resolver that typically resides in a resolver pool for query load balancing purposes.

Two different scenarios can be considered for recursive resolvers: servers with private IP addresses and servers with public addresses. Recursive resolvers with a public IP address may be exposed to the public Internet if misconfigured, despite the fact that they are designed to only serve the clients of their own network. Even with a proper access control mechanism, if DSAV is not implemented, packets originated externally can be made to look like they originated from trusted clients and thus get delivered to these resolvers. On the other hand, recursive resolvers with a private IP address [29] would not be reachable by an external host, unless there is a direct peering between two networks.

Contrary to the recursive resolvers, the upstream resolver pool hosts need to use a public IP address as they communicate with authoritative nameservers on the Internet. Inside the cloud, resolver pools are typically meant to only be reachable by the recursive resolver(s) and not directly by clients. We will discuss various ways in which recursive resolver(s) and the resolver pool(s) of a cloud network might be misused to launch DDoS attacks in Section 3.

3 ATTACK MODELS

Considering common DNS implementations in cloud provider networks (see Section 2.3), we formalize six models by which attackers can use cloud infrastructure to launch R&A attacks. We name these *A* through *F*. As we will show, the six models vary in terms of, among others, the type and location of misused infrastructure, and the role of defense mechanisms.

Multiple of the attack models formalized in the next subsections differ from traditional R&A DDoS attacks. In a traditional DNS-based DDoS attack, the IP address(es) of a victim are spoofed as source IP addresses and queries are sent to open DNS resolvers. In a number of our attack models though, spoofing with the IP address of an arbitrary victim would result in these queries being dropped, since we are spoofing towards closed resolvers of cloud networks. Thus, the spoofed source address needs to be in the range dedicated to clients of the cloud. Also, in our attack models, next to the typical victim of a DDoS attack, the recursive resolver(s) or the resolver pool of a cloud as well as the authoritative nameserver(s) of the queried domain name may be the intended victim. This further extends the scope of our attack models to attacks such as NXNSAttack [1]. We summarize the six models in Table 1. Note that by "attack origin" we mean the location of hosts issuing spoofed attack traffic and not the attacker themselves. Also note that in the remainder of this section we differentiate between the words "target" and "victim" in the following manner. The word "target" is used to refer to a host, which ultimately receives the DNS responses because its IP address was misused when issuing spoofed DNS queries. On the other hand, the word "victim" refers to any stakeholder that may get disrupted due to receiving a large amount of traffic that it cannot handle.

3.1 Open Resolvers in Datacenters

Open DNS resolvers exist on the Internet in the order of magnitude of millions. A significant number are hosted in networks that provide user services such as broadband, which suggests that at the very least some are running on consumer devices (e.g., routers and modems) [19]. Open DNS resolvers may also be hosted in a cloud network or datacenter, for example if customers expose a stub resolver to the Internet. We postulate that the latter category, on average, are better provisioned than those found in residential access networks. Our first attack model (model *A*) accounts for cases where such resolvers are leveraged as a reflector in R&A DDoS attacks. Although misusing open resolvers in a DDoS attack is not a new concept, our focus is on the potential provided by datacenter-based open resolvers. A high-level overview of this attack model is shown in Figure 2. Under this model, hosts in a botnet which are located in networks that lack OSAV, issue spoofed DNS queries (dashed arrows in Figure 2) selectively towards datacenter-based open resolvers. Answers for these queries are reflected towards a target (or a set of targets), which could potentially be any host on the Internet (internal or external to the network hosting the open DNS resolvers). Note that a target does not necessarily need to be the victim in this scenario, e.g., with a large set of IP addresses used as the target set, an authoritative nameserver might become the victim of such an attack. In Section 4.1 we present our methodology to investigate datacenter-based open resolvers.

Table 1: Summary of different attack models

Attack model	Attack origin	Abused hosts	Potential Victims	Impact
A	Outside cloud	Cloud-based open resolvers	Any host on the Internet	As in typical R&A DDoS
B	Inside cloud	Open resolvers inside or outside the cloud		
C	Outside cloud	Recursive resolver(s) of the cloud	DNS infrastructure of the Cloud, authoritative nameserver(s), cloud clients	1) Taking down the DNS infrastructure of the cloud which would impact all its clients, 2) Causing disruptions for authoritative nameservers of the queried domains, 3) Causing disruptions for cloud clients
D	Outside cloud	Resolver pool of the cloud		
E	Inside cloud	Resolver pool of the cloud	DNS infrastructure of the Cloud, authoritative nameserver(s), cloud clients or external hosts	1) Taking down the DNS infrastructure of the cloud which would impact all its clients, 2) Causing disruptions for authoritative nameservers of the queried domains, 3) Causing disruptions for cloud clients or external hosts
F	Inside cloud	Recursive resolver(s) of the cloud		

3.2 Spoofing Towards Open Resolvers from a Cloud Network

Attack model *B* concerns networks that might be the origin of a reflection-based DDoS attack (see Figure 3) by allowing customers to send spoofed requests towards reflectors. Alike model *A*, model *B* also does not involve DNS infrastructure specific to the network but rather open resolvers, which could be inside the same network or in an external network. A potential target for such an attack might reside either inside the cloud network or in an external network. Cloud providers implementing BCP 38 can immediately block such attack traffic near the source, either at the edge of their network (see Section 2.2.1) or even closer (e.g., at a hypervisor level). However, if open resolvers are inside the cloud network, filtering on the edge router would not be of any help in this attack model. In Section 4.4 we discuss our experiment setup to investigate this attack model.

3.3 Recursive Resolver(s) as a Reflector for Externals

We now shift to two attack models that involve DNS infrastructure specific to the cloud provider. The recursive resolver(s) as discussed in Section 2.3 (see also Figure 1) are meant to exclusively serve the clients internal to the network. Our third attack model, model *C* in Figure 4, concerns cases where such resolvers can be contacted by external hosts. This mainly applies to networks that assign publicly routable IP addresses to their recursive resolvers (unless there is a peering between the attacking and target network). Even while such resolvers restrict access to internal IP addresses, lack of DSAV allows external hosts to still contact the recursive resolver(s), by purporting to be a client inside the network. In such a scenario, the attacker sets the source IP address to an address of the network for which the resolver provides service. Once such spoofed packets arrive at the recursive resolver due to lack of DSAV, the resolver is not able to verify the legitimacy and processes the request as if they originated from an internal client.

Multiple intended victims can be considered for such an attack, depending on several factors such as size of the spoofed source IP address set, diversity of the query names used in the spoofed packets, capacity of the recursive resolver(s), etc. A first potential victim could be a client host of the network. Since recursive resolver(s) are typically deployed to serve a large number of clients, such an attack can cause a disruption for client hosts if spoofed queries use a limited set of the IP addresses. Other potential victims could be the DNS infrastructure of the cloud provider (i.e., the resolver pool or the recursive resolvers) or the authoritative server of the queried domain name. We discuss our methodology for experimenting with this attack model in Section 4.5.

3.4 Resolver Pools as a Reflector for Externals

Our next attack model concerns cases where the resolver pool is leveraged by outsiders to bring about reflection-based DDoS attacks. This model is shown in Figure 5. As we have discussed in Section 2.3, typically a resolver pool is not meant to be directly accessible even to clients internal to the network. If the resolver pool is directly accessible to customers and the network does not use DSAV, external hosts may, analogously to model *C*, be able to contact the resolver pool by purporting to be an internal client. As we will show in Section 5, several cloud providers are susceptible to this type of abuse. As before, attack model *D* can have multiple intended victims, depending on the capacity of the infrastructure and on attacker-controlled parameters (query names and number of spoofed client addresses, among others). Similar to the previous attack model, the methodology for this attack model is presented in Section 4.5.

3.5 Resolver Pools as a Reflector for Insiders

If the resolver pool of a network is directly accessible to internal clients, it may be misused by insiders to reflect spoofed queries towards a target. Two scenarios might be considered for a potential target: hosts inside the cloud network, and external hosts. The resolver pool should normally reject such packets if they are using a source IP address external to the network. However, it might be the case that access controls (if they exist) are only applied at the edge of the network, thus exposing the resolver pool to insider abuse. A layout of such an attack (model *E*) is shown in Figure 6.

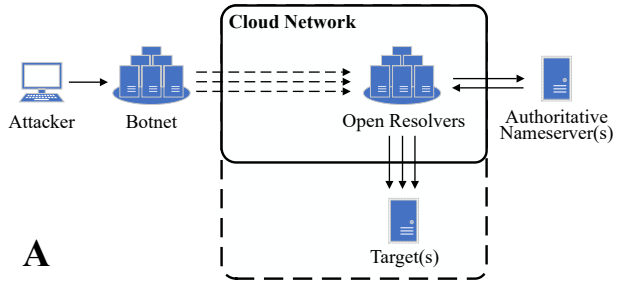

Figure 2: Attack model *A*: Open resolvers in datacenters as reflectors that can be misused by an external source to bring about a reflection-based attack, either to a target internal or external to the network of the reflector

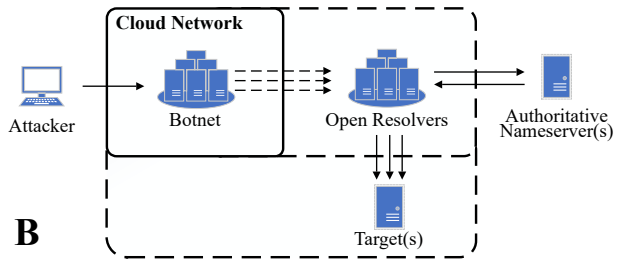

Figure 3: Attack model *B*: Spoofing towards open resolvers from a cloud network, where the open reflectors as well as the target can be internal or external to the network

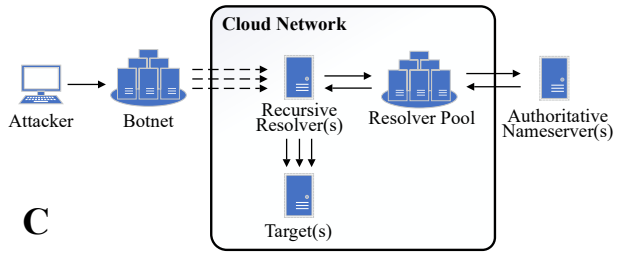

Figure 4: Attack model *C*: Recursive resolver(s) of a cloud provider network are misused as reflector by an external source

Similar to the previous models, attack model *E* can have multiple intended victims, which is again influenced by the capacity of the infrastructure and by attacker-controlled parameters such as query names and number of spoofed client addresses. In Section 4.4 we discuss our experiment setup to investigate this attack model.

3.6 Recursive Resolver(s) as a Reflector for Insiders

Similar to model *E*, recursive resolvers, if not properly restricting access to internal clients, may be susceptible to spoofing. Due to the limited size of the recursive resolver set in a single cloud provider, a logical victim for such an attack might be the recursive resolvers themselves or the authoritative servers as such packets can be easily

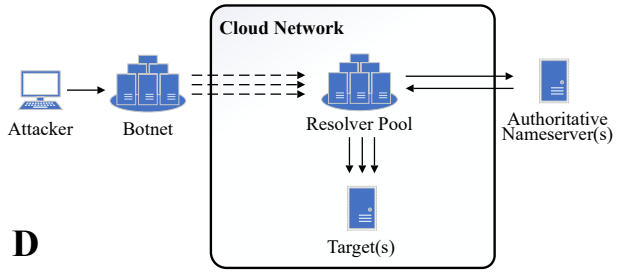

Figure 5: Attack model *D*: The resolver pool of a cloud provider network is misused to generate a reflection-based attack by external hosts

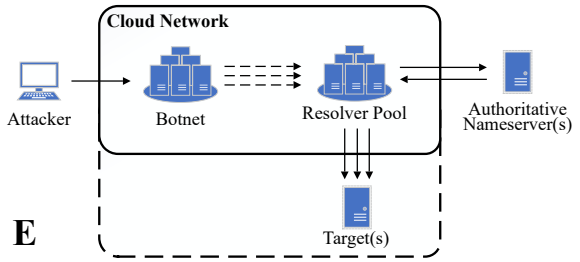

Figure 6: Attack model *E*: The resolver pool of a cloud provider network is misused for reflection by insiders and the target of the reflection can be internal or external to the cloud provider network

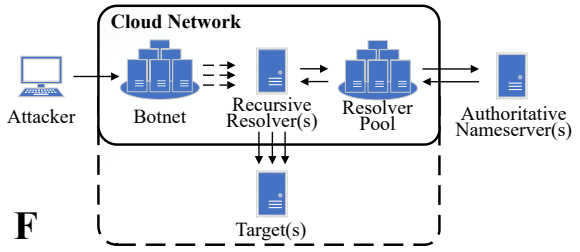

Figure 7: Attack model *F*: The recursive resolver(s) of a cloud provider network are misused as reflectors by insiders and the target of the reflection can be inside as well as outside the cloud provider's network

dropped at the destination network since they will be coming from a limited number of IP addresses. In Figure 7 we depict this sixth and last attack model. Similar to the previous attack model, the methodology for this attack model is presented in Section 4.4.

4 METHODOLOGY

In this section we detail our methodology to: quantify open DNS resolvers; select cloud providers and discover their DNS infrastructure; and assess feasibility of the six attack models.

Table 2: *usage_type* field values in IP2Location database

usage_type	Description
COM	Commercial
ORG	Organization
GOV	Government
MIL	Military
EDU	University/College/School
LIB	Library
CDN	Content Delivery Network
ISP	Fixed Line ISP
MOB	Mobile ISP
DCH	Data Center/Web Hosting/Transit
SES	Search Engine Spider
RSV	Reserved
ISP/MOB	Fixed and Mobile ISP

4.1 Datacenter-based Open Resolvers

With attack model *A* in mind (see Section 3.1), we quantify to what extent open DNS resolvers exist in cloud networks by scanning for them. This requires us to tie IP addresses to clouds, a step for which we considered several approaches. Various cloud providers (e.g., Amazon, Microsoft, Google) regularly publish their range of IP addresses to help network administrators allowlist traffic. Many cloud providers however do not publish this information. In addition, this approach does not scale well as there is no complete list that identifies all of the cloud providers on the Internet.

A second approach involves leveraging reverse DNS records, as pointer records (PTR) may allow inferences to be made about the intended use of IP addresses. For example, a reverse DNS lookup for an IP in the cloud range of Amazon would result in a string that can be used to infer that IP belongs to a cloud instance (e.g., `ec2-ip.compute-1.amazonaws.com`). This method has its own limitations. First, not all providers setup reverse DNS records. Second, such inferences may be non-trivial as the content of pointer records is not always clean-cut.

A third approach is to make use of the information provided by IP intelligence databases. Compared to the previous two approaches the latter is more comprehensive and easy to use. The methodology used for data collection of such databases is typically not published for commercial reasons. However, there have been studies [14] which investigate the geolocation accuracy of these datasets. Nevertheless, to the best of our knowledge, there has been no study that audits the accuracy of network type data in such databases and this is left as a potential future work. We use the IP2Location [15] database to quantify open resolvers in cloud networks. (The results of this quantification will be presented in Section 5.1.) The IP2Location database includes a field called *usage_type* for each IP address. This field can take various values such as GOV, CDN, DCH, EDU, to name a few. A full list of values for this field and their description can be found in Table 2. We use the DCH (Data Center/Web Hosting/Transit) tag to infer IP addresses located in a datacenter. Note that a DCH-tagged IP address does not necessarily equate to an IP address being tied to a cloud platform. This is however the most fine-grained classification of network types that we could acquire. Besides, we argue that Data Center, Web Hosting and Transit networks all fall under the umbrella of well-connected networks which are the network types of concern in our study.

Table 3: Ranking of top DCH-tagged ASes

Rank	ASN	AS name	DCH IP (count)	DCH IP (%)
1	**AS16509**	**Amazon.com, Inc.**	**37.45M**	**10.58%**
2	**AS8075**	**Microsoft Corporation**	**35.31M**	**9.98%**
3	**AS14618**	**Amazon.com, Inc.**	**14.56M**	**4.11%**
4	**AS37963**	**Hangzhou Alibaba Advertising Co.,Ltd.**	**11.14M**	**3.15%**
5	**AS15169**	**Google LLC**	**9.80M**	**2.77%**
6	AS2907	Research Organization of Information and Systems, National Institute of Informatics	7.58M	2.14%
7	AS3356	Level 3 Parent, LLC	7.26M	2.05%
8	AS2914	NTT America, Inc.	6.29M	1.78%
9	**AS36351**	**SoftLayer Technologies Inc. (IBM)**	**5.20M**	**1.47%**
10	AS10310	Oath Holdings Inc.	4.33M	1.22%
11	**AS45090**	**Shenzhen Tencent Computer Systems Company Limited**	**4.01M**	**1.13%**
12	AS71	Hewlett-Packard Company	3.67M	1.04%
13	**AS16276**	**OVH SAS**	**3.60M**	**1.02%**
14	AS701	Verizon Business/UUnet	2.84M	0.80%
15	**AS14061**	**DigitalOcean, LLC**	**2.40M**	**0.68%**
16	AS17506	ARTERIA Networks Corporation	2.25M	0.64%
17	AS3741	Dimension Data	2.25M	0.64%
18	**AS45102**	**Alibaba (US) Technology Co., Ltd.**	**2.19M**	**0.62%**
19	**AS24940**	**Hetzner Online GmbH**	**1.95M**	**0.55%**
20	AS4589	Easynet Global Services	1.63M	0.46%
...	...	...	...	...
82	AS60781	LeaseWeb Netherlands	0.43M	0.12%
83	**AS35908**	**Krypt Technologies**	**0.42M**	**0.12%**
...	...	...	...	...

4.2 Selecting Cloud Providers to Study

A wide range of providers play a role in contributing to the entire ecosystem of cloud platforms. Since it is not feasible to investigate hundreds of providers, we have taken a systematic approach to select cloud providers for our study. We extract any IP address tagged as a DCH (Data Center/Web Hosting/Transit) from the IP2Location dataset. We then group by AS number and sort the aggregate by the number of IP addresses per ASN. The descending-order result gives us a list of DCH-tagged ASes that own the largest number of IP addresses. As we will explain later, our methodology involves renting Virtual Private Server (VPS)es from cloud providers. Not every single DCH AS offers VPS services. As such, we manually check if networks offer VPS service by looking up products on their web pages. Besides, a number of VPS providers only provide their services for enterprises, these we also dropped from our list. Table 3 shows the top ASes tagged as a DCH network in the IP2Location database in descending order of the number of DCH-tagged IP addresses that they own. Due to time and cost limitations, we limit our study to a selection of the top providers. Our selection involves 19 providers (selected from the top 83 ASes tagged as a DCH network).[1] Our study thus considers Amazon, Microsoft, Alibaba, Google, IBM, Tencent, OVH, DigitalOcean, Hetzner, Oracle, Vultr, Rackspace, IONOS, Eonix (Serverhub), Linode, B2 Net (Servermania), Online

[1]Note that we had chosen 20 cloud providers, but dropped one at a later stage because VPS rental requires a local, i.e., non-overseas, credit card.

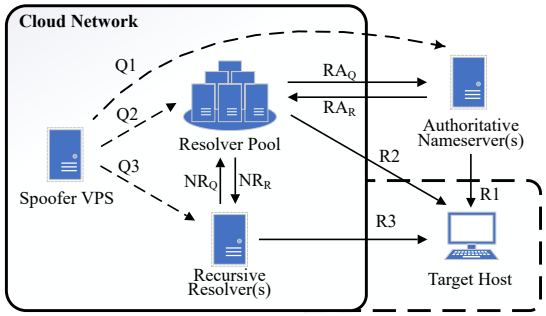

Figure 8: Measurement setup for OSAV deployment with the spoofing source (VPS) inside the cloud provider's network (attack models B, E and F)

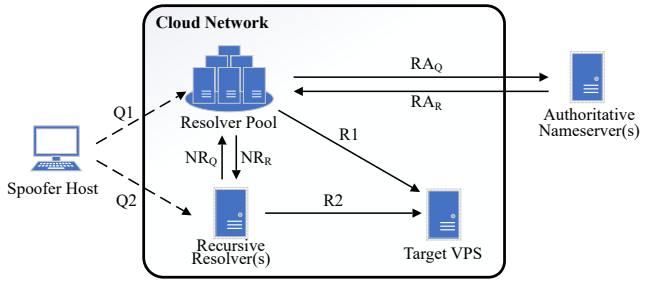

Figure 9: Measurement setup for DSAV deployment with the spoofing source (host) in an external network and a target under our control inside the cloud provider's network (attack models C and D)

S.A.S. (Scaleway), QuadraNet, and Krypt. This accounts for at least 39% of the DCH-based networks in the IPv4 address space.

An alternative approach for cloud provider selection could be to make use of the ratings of top cloud providers, such as provided by Gartner [12]. We decided not to take such an approach to avoid potential bias (e.g., commercial) in our study and to devise a reproducible top ranking ourselves.

4.3 Identifying Cloud DNS Infrastructure

Multiple of our attack models involve an investigation into the DNS infrastructure of the cloud provider, which is meant to serve the provider's customer. A prerequisite for these investigations is collecting the IP addresses of the DNS infrastructure (i.e., recursive resolvers and the resolver pool), which we do as follows.

We start by launching VPSes running a Debian operating system on the public clouds. Once our VPS is up and running we look at the content of `/etc/resolv.conf`, which is the main DNS configuration file for Unix-like operating systems and contains the IP address(es) for the recursive resolver(s) of a network which are configured by the provider using DHCP. While launching multiple VPSes per provider and also in multiple regions (more on this later), we observe that the set of resolvers in `resolv.conf` largely remains consistent. This gives us confidence that we usually learn many to all resolvers for a given cloud provider. In addition, several of the selected providers also publish their recursive resolver(s) in online documentation and these lists are consistent with our inferences.

In order to identify the resolver pool addresses, we issue multiple DNS queries towards the collected (provider-related) recursive resolver(s) from the launched VPSes, using a domain name (and authoritative nameserver) under our control. We listen for DNS queries that arrive on our authoritative nameserver and that were issued by the resolver pool hosts of the cloud. Parsing the captured DNS traffic gives us a subset of resolver pool IP addresses that we use when investigating the relevant attack models.[2] Note that this way we might miss some resolver pool IP addresses as we have no control over which resolver pool host contacts our authoritative nameserver. However, this is not a concern for our study as we do not need to exhaustively quantify the resolver pool and learning the IP addresses of a handful of such servers is sufficient to investigate the feasibility of our attack models.

[2]Note that some providers configure a third-party public DNS resolver as a recursive resolver. These providers are not susceptible to some of our attack models and we exclude them from resolver pool discovery.

4.4 Spoofing Originated Inside Cloud

To investigate to what extent cloud providers are susceptible to being misused as the origin of R&A DDoS attacks (attack models *B*, *E* & *F*), we explore the deployment of network ingress filtering (BCP 38) in cloud provider networks.

From rented VPSes, we issue a handful of spoofed queries towards internal and external hosts under our control. As for the spoofed source addresses, our queries used either the IP address of another VPS in the same region or an external host (see Figure 8). In both cases, the alleged source hosts were under our control. For destination, we use both the IP addresses of the cloud DNS infrastructure (recursive resolvers and directly contactable resolver pool hosts) and the authoritative nameserver of the queried domain. By including our authoritative nameserver as a reflector, our experiment also mimics cases where open resolvers external to the cloud network might be misused as a reflector by issuing spoofed queries from cloud-based hosts (see Figure 3). Spoofed DNS queries Q1, Q2 and Q3 are sent to the recursive resolver(s), resolver pool and the authoritative nameserver, respectively. If there is no OSAV, these queries get processed and the corresponding responses R1, R2 and R3 are reflected towards the target. Note that OSAV might be deployed at different levels, e.g., at the network edge or at the hypervisor level. If OSAV is only deployed at the edge router, spoofed queries towards the DNS resolvers of cloud should still be received at an internal target. By spoofing to different resolvers as possible reflector we make sure to test different levels of OSAV implementation.

In all of the cases in Figure 8, we test using a query name for a domain for which we control the authoritative nameserver. This way we are already able to confirm if our spoofed queries are processed by checking if they end up at our authoritative server. However, by spoofing IP addresses to addresses of hosts under our control, we make sure that even if spoofed packets traverse through the network, they end up being reflected to ourselves (see also our ethical considerations in Section 8). This also allows us to further verify if the DNS responses reach their destination.

4.5 Spoofing Towards Cloud

Attack models *C* and *D* (see Sections 3.3 & 3.4) both have a precondition: spoofed queries from external networks need to be able to successfully reach DNS infrastructure in the cloud provider's network. This is impossible if DSAV is deployed (see Section 2.2.2). To investigate to what extent this is the case for our selection of cloud providers, we conduct a proof of concept study, the results we present in Section 5.2.

Our approach is as follows. We issue spoofed queries from a host in an external network that lacks OSAV (Spoofer Host in Figure 9) to the cloud DNS infrastructure (recursive resolvers and directly contactable resolver pool hosts). We set the spoofed source IP addresses to the addresses of VPSes under our control (Target VPS in Figure 9). Thus, if any spoofed packet is reflected, it would be reflected towards ourselves. For each query we use a unique query name with a domain name under our control. Thus, we can observe the corresponding requests at our authoritative nameserver if the spoofed query is acted upon by the cloud provider's DNS infrastructure. We also embed the destination address (recursive resolver or the resolver pool host) into the query name, so we can infer which resolver acted on the query, even if the resolver forwarded the query. Each query is unique so even in the case of shared caches, the resolver should still query our authoritative server. We ensure that our authoritative returns `NOERROR` status codes for all queries under the experiment domain. Since we spoof queries with the source IP addresses of hosts under our control, we can verify whether DNS responses are actually delivered to those hosts.

Once we identify which cloud providers are vulnerable to our attack models, a next step to further investigate the potential impact of such attacks could be to investigate if RRL is implemented on the DNS infrastructure inside the provider's network (see Section 2.2.3). We have decided not to perform this experiment for multiple reasons. First, the DNS infrastructure that we issue our queries towards are meant to serve the customers of the network under study. RRL tests need to be done using a burst of DNS queries. Since we have no information on the configuration parameters of the RRL on DNS servers under study, we would need multiple experiments to obtain confident results. This might cause disruptions to the normal performance of these servers which we consider unethical. Second, even if an RRL mechanism is deployed, one can reduce its efficiency in several ways, e.g., by issuing random subdomain queries or sending queries from various subnets. Finally, RRL is mainly meant to be implemented on authoritative nameservers, while we explore recursive resolvers and resolver pools of cloud networks [34].

To limit the ethical concerns of the measurements that we do perform as much as we can, we launch VPS instances for each provider (similar to Section 4.4) and use the IP addresses of these instances as source addresses in our spoofed DNS queries (i.e., if queries are reflected, they are delivered to hosts under our control). In order to further verify our findings, we conduct our measurements within multiple regions of the cloud platforms as there may be differences between these.

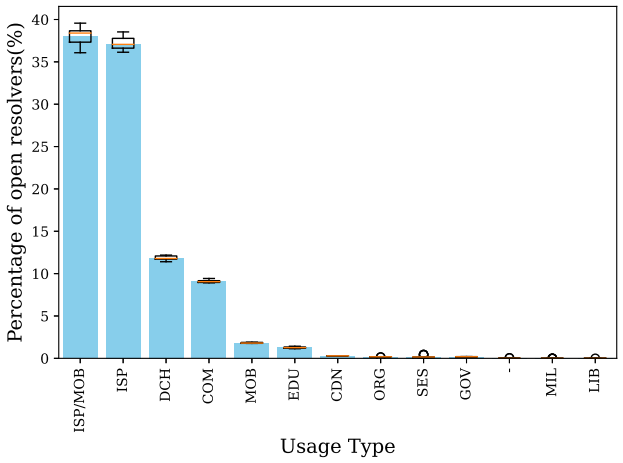

Figure 10: Usage (network) types of open resolvers

5 RESULTS

In this section we report on our feasibility assessment of the six attack models. This includes a quantification of open DNS resolvers (attack model *A*), our observations following insider spoofing tests (models *B*, *E* & *F*), and findings related to provider susceptibility to outsider spoofing and reflection via internal DNS infrastructure (models *C* & *D*).

5.1 Cloud-based Open Resolvers

Considering attack model *A*, we conducted a study to quantify to what extent open resolvers exist on cloud-based platforms. Note that the feasibility of misusing open resolvers in R&A DDoS attacks has already been proven for a long time. Thus, our goal here is not to experiment with the feasibility of attack model *A* but rather to quantify the extent to which such a potential exists within datacenter networks. As part of other, ongoing research efforts, we are already scanning the IPv4 address space for open DNS resolvers on a weekly basis[3]. This measurement results in a list of about 2.7 million open resolvers (each week) that return a correct answer for the queried DNS record. We reuse this data and cross-reference it against IP2Location data to infer which of the open DNS resolvers are hosted in datacenters, as outlined in Section 4.1.

We conduct a longitudinal study, using weekly open DNS resolver scan data from January 4 to June 14, 2021. We show the usage type of these resolvers in Figure 10. We can see that, as far as our measurement goes, mobile and fixed line ISP networks contribute to the vast majority of open DNS resolvers in the IPv4 address space. Note that this involves the ISP, MOB and ISP/MOB network types[4] and comes down to 77% on average. These findings are consistent with the observations of Kührer et al. [19] and also suggest that a non-negligible number of open DNS resolvers may be running on consumer devices such as routers and modems.

[3]Note that we discuss the ethical considerations for scanning the entire IPv4 address space in Section 8.

[4]IP2Location data has different tags to distinguish if a network is, e.g., exclusively for mobile access.

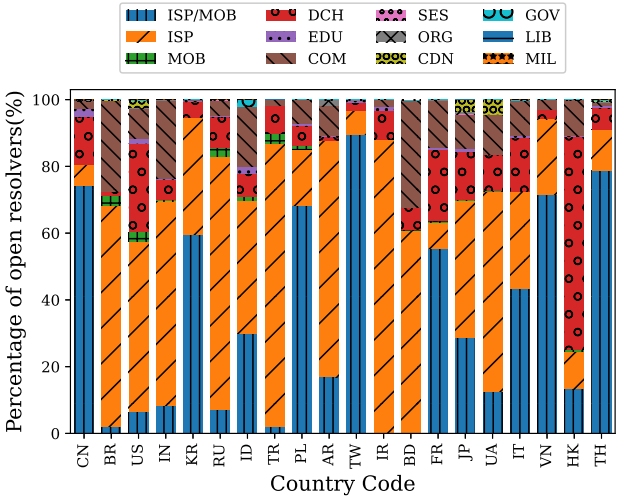

Figure 11: Distribution of open resolvers in various network types among top 20 countries hosting open resolvers

If we look at the network type that is the specific focus of our paper, DCH, we see that open resolvers located in likely well-provisioned datacenters approximately comprise 12% of the entire set on average. In absolute numbers, this comes down to approximately 315k resolvers. This number is multiple orders of magnitude larger than the number of reflectors misused in 90% of DDoS attacks as reported by Nawrocki et al. [25], highlighting that the pool of well-provisioned reflectors is large enough for attackers. To investigate if the distribution of discovered open DNS resolvers among the various network types substantially changes between weekly scans, we also add the percentile boxplots. The datacenter share varies only slightly over our study period, between 11.4% and 12.2%. Considering the other top network categories in Figure 10, we observe that on the whole, the network type distribution of open DNS resolvers does not change substantially over the study period.

Self-evidently, open DNS resolvers in datacenters (and cloud providers housed therein) are not the only potential reflectors that likely are well-provisioned and enjoy high network capacity. However, we find increasingly fewer resolvers in the remaining network types that are likely well-provisioned. For example, the educational networks category, EDU, is only responsible for 1.2% of open resolvers.

As part of our ongoing weekly scans, we also issue `version.bind` queries towards open resolvers to classify the software running on them. In general, different versions of `dnsmasq` run on the majority of the open resolvers. At the same time, our results show no considerable differences in resolver software distribution when comparing various network types. As such, we did not identify particular software characteristics that set apart open DNS resolvers in datacenters.

Using the geolocation information in IP2Location data, we map the discovered open DNS resolvers to countries. A breakdown of our results for the top 20 countries hosting open resolvers on June 14, 2021 is shown in Figure 11. We sort countries by the overall number of open resolvers, meaning for all network type categories. For each of the 20 countries shown, we also plot the distribution of open resolvers specific to that country within various network types. Considering the main focus of this paper, datacenter networks, we observe a considerable diversity in the proportion of datacenter-based open resolvers among countries. While only 1.1% of open DNS resolvers in Brazil are located in a datacenter, for Hong Kong this number goes up to 64%, in both cases deviating considerably from the overall percentage of datacenter-based open resolvers on the Internet (as we show in Figure 10).

Key Takeaway: *Our quantification of open DNS resolvers in various network types shows that, among the likely highly-provisioned categories, the datacenter/cloud type is the most prominent (Figure 10). Hundreds of thousands of open resolvers exist in such networks, which arguably provides attackers with potent reflectors to choose from under attack model A.*

5.2 Spoofing Originated Inside a Cloud

We now move on to attack possibilities that can result from the ability to spoof from within a cloud provider's network. That is, we investigate attack models *B*, *E* & *F*. We apply our methodology from Section 4.4 to the 19 cloud providers selected for our study. Recall, we send a handful of spoofed queries towards hosts inside the cloud provider's network (i.e., recursive resolvers and resolvers in the pool) as well as an external host (our authoritative nameserver). Our experiments reveal that all of the providers that we study, deploy BCP 38 and drop spoofed traffic originated in their network if the spoofed IP address is external to the cloud provider's network. This is a good sign: despite the fact that OSAV primarily benefits other networks, we infer that it is in place. Two providers, however, partially allow spoofed traffic when the spoofed source IP address is also in the same network range as the original source IP address. This limits the feasibility of attack models *B*, *E* & *F* to internal targets. Although the target in this case needs to be internal, the intended victim can still be outside, for example the authoritative nameserver for the domain queried for (see Sections 3.5 & 3.6).

Our view on the susceptibility of cloud platforms to provide spoofing capabilities is centered on the providers that we have examined as discussed in the previous section. To expand this view somewhat, we have made use of the Spoofer project's data [5]. The Spoofer project collects data on deployment of source address validation on the Internet. Similar to Section 5.1 we extracted the list of IP blocks that are tagged as DCH and then looked up the corresponding AS number using the *pyasn* library and loading BGP archives from RouteViews project [26]. An IP2Location lookup on June 7, 2021 resulted in 17,123 unique ASNs tagged as DCH. Cross-referencing these ASNs with the Spoofer project data leaves us with only 546 ASes (3.2%) that are measured for SAV between May 1 and June 21, 2021. Of these networks, 89 allow spoofed IPv4 queries (with a routable IP address) originated in a subset of their IP ranges to exit their network.

Key Takeaway: *Despite the fact that the benefits of OSAV deployment go to other networks, our experiments reveal that top cloud providers almost fully deploy this defense mechanism. This however*

should not be generalized to other cloud providers as we already would expect well-provisioned providers to be more concerned about their network's implementations.

5.3 Spoofing Towards a Cloud

We now present our results relating to the attack models that involve spoofing towards a cloud provider. That is, we investigate an attacker's potential to effectuate models *C* or *D*. Recall, these two attack models involve an attacker setting internal network addresses in spoofed packets to contact the DNS infrastructure of the cloud provider and have it reflect attack traffic. For attack model *C* in particular, the recursive resolver needs to be assigned a publicly routable IP address by the cloud provider (see Section 3.3). Our DNS infrastructure identification steps, as discussed in Section 4.3, have revealed that 11 out of 19 providers use a publicly routable IP address(es) for their recursive resolver(s). We ran our experiments on May 28, 2021. The spoofed queries that we sent from an external host towards the resolvers of the aforementioned 11 providers were successfully resolved and reflected towards VPSes under our control for 3 out of these 11 providers. This means that we were able to observe DNS responses with `NOERROR` response code for spoofed queries that we had sent from the outside to each cloud provider network. There was also a one to one match between the queries for which our authoritative nameserver was contacted and the (reflected) responses that we received on VPSes under our control.

To assess the susceptibility of the 19 cloud providers to attack model *D* we first sent (unspoofed) DNS queries from VPSes directly to the resolver pool hosts of the cloud providers, bypassing the recursive resolver(s). We discovered that 6 out of 19 providers expose resolver pool hosts directly to the hosts of customers. We next sent spoofed queries from an external host towards resolver pool hosts. For 2 out of 6 providers the spoofed queries were successfully resolved and reflected, like before. The set of providers susceptible to attack model *D* is a subset of the ones susceptible to attack model *C*. We infer that the involved providers do not implement DSAV. Note that the focus of our experiment here is on the exposure of cloud provider DNS infrastructure to external hosts. A more generic approach would be to send spoofed packets towards a resolver inside the cloud network other than those designed to serve the clients. Thus, our findings represent a lower limit for cloud providers that do not implement DSAV. Also note that while we detect a limited number of providers to be vulnerable to our attack models *C* and *D*, this was to some extent expected as we investigate top cloud providers, which are supposed to be well-provisioned. The number of vulnerable networks might quickly grow by extending our study to providers lower in the ranking.

We note that none of the providers that use private IP addresses for their recursive resolver(s) expose their resolver pool to their clients, which makes these providers secure against both our attack models *C* and *D*. Four providers use public (open) DNS resolvers as recursive resolver(s), which excludes them beforehand (see Section 4.3).

5.3.1 Multiple Regions. Cloud providers typically host their infrastructure in multiple regions, ranging from a few regions for smaller providers, to tens of regions for larger providers in our selection. These providers implement different policies for their recursive resolver addressing. 9 out of 11 providers that use a public IP address for their resolvers, consistently assign the same set of IP addresses over the entire set of regions. We suspect that these providers deploy anycast, but consider confirming this intuition out of scope for this paper. The remaining two providers on the other hand have unique recursive resolver addresses per region. Nevertheless, the resolvers of a region are accesible to clients in all other regions. Considering the resolver pool hosts, 5 out of 6 providers that expose their resolver pool to be directly reachable by clients, also expose their resolver pool to the clients of other regions.

Key Takeaway: *While DSAV is meant to protect networks from external threats, even among top cloud providers there are networks that do not deploy this mechanism. Our findings reveal a lower limit of this issue as we only target the internal DNS infrastructure of cloud networks.*

6 DEFENSE MECHANISMS

This paper describes a number of attack models specific to cloud DNS infrastructure. We reason here about how those attack models can be mitigated. Based on our overview of operational best practices (Section 2.2), it is our observation that a more careful deployment of operational practices is key to mitigate those attacks.

First, to avoid attack model *A*, a provider needs to make sure to patch misconfigured hosts which are exposed as an open resolver. Even though the damage of this attack model is not directly on the core infrastructure of a cloud, it is always a good operational practice to avoid unnecessary exposure of services on the Internet.

Moreover, we suggest that operators further harden their core cloud infrastructure. As we discussed in Section 2.3, clients of a network do not necessarily need a direct access to the resolver pool. Thus, better access control on the resolver pools would already help to prohibit attack models *D* and *E*, even if other mitigation mechanisms are not deployed.

The attack models *B*, *E* and *F* can be easily avoided by deploying OSAV (see Section 2.2.1). Our measurements show that almost all of the top cloud providers that we study properly deploy OSAV. The important detail here, to which we feel more attention should be given, is that OSAV needs to be deployed at different layers of the network and not just at the edge router.

Finally, in order to secure networks against attack models *C* and *D*, providers need to deploy DSAV (see Section 2.2.2). Considering that this mitigation strategy is of direct benefit for providers, there is more incentive for them to put efforts in deploying such a mechanism. Nevertheless, our experiments revealed that even among top providers some still lack DSAV deployment.

7 DISCUSSION AND LIMITATIONS

In this section, we reflect upon the limitations of different aspects of our study, from input data quality to methodological subtleties.

Our methodology to identify and make a selection of cloud providers in Section 4.2 relies on metadata provided by the IP intelligence dataset of IP2Location. Although we cannot verify the accuracy of this metadata, our goal in arriving at a selection was to conduct a proof-of-concept study and not to create a flawless

ranking of top cloud providers. Using this methodology we aim to assess the feasibility of our attack models on the Internet. Considering that we investigate top providers which usually have enough resources to provision their networks securely, our methodology likely largely underestimates the extent of the problem and only sheds lights on the tip of the iceberg. There exist online reports on cloud provider rankings such as the market report by Gartner [12], which could serve us in a similar way. We have, however, made a deliberate choice not to use this type of source data to avoid any potential marketing bias and to devise ranking that can easily be reproduced by others.

In this paper we explore 19 public cloud providers. Assuming that these providers represent a substantial part of the cloud market, one might intuitively expect that they operate a well-designed network. Nevertheless, there also exist a number of so called bullet-proof cloud providers that deliberately offer spoofing capabilities to their clients. We have not included these providers in our study as there is no explicit list to explore such providers and for solidity of our study we decided to choose providers based on the size of the IP address space that they own.

We use the DCH tags from the IP2Location database to infer open resolvers residing in a cloud network in Section 4.1. As we have highlighted earlier in this paper, a DCH tag is not equivalent to an IP address tied to a cloud network, which could be seen as an accuracy-related limitation. However, we argue that all of the three network types included in this category are examples of well-provisioned networks, which is the main focus of this paper, and a precise classification is less important than identifying the attack potential.

Deployment of BCP 38 might be done differently for various parts of an infrastructure. In this study we investigate BCP 38 deployment using VPSes running on cloud platforms. Our findings may or may not apply to the entire network of the same provider. This might for example be the case when a hypervisor running on a host machine already drops spoofed queries originated from virtual machines while there is no ingress filtering at the edge router of the network to filter out spoofed queries issued by dedicated servers. Besides, due to time and cost limitations, we have only investigated 19 top cloud providers. This obviously limits our insight into the full ecosystem of cloud platforms on the Internet as there exist hundreds of such providers.

8 ETHICAL CONSIDERATIONS

In our research we have identified several issues that require ethical considerations: the open DNS resolver scan, sending spoofed traffic, testing for rate limiting, and the disclosure of our findings to vulnerable providers. These are discussed individually below.

8.1 Open Resolver Scans

As part of an ongoing research project we have been running full IPv4 address space scans to detect open DNS resolvers. We use some of the resulting data in this paper (see Section 5.1). To minimize the impact of the scans, we have adhered to the recommended best practices [3] in the following manner. First of all we have limited our scans to once per week. Second, we have distributed our scans randomly over the IPv4 address space. This way we make sure that we are not causing a burst of disruption on a specific network. Additionally, we have instated an opt-out procedure for network operators who are not willing to be scanned by us. This was done by using a query name that makes us recognizable for operators. Also, a PTR record was set for our scanner machine, which points to a web page with details of our project and opt-out procedure.

Historic data of open resolvers exists, but was not suitable for our research. The Open Resolver Project[28] has ceased operating and no longer performs scans. The Shadowserver Foundation[11] performs scans of the entire Internet and is available for querying, but does not share data directly. Censys[6] scans for open resolvers but at the time we requested access they were making changes in their data access policies, and did not allow access in the meantime.

8.2 Spoofed Traffic

Part of our study involves sending DNS queries with spoofed IP addresses both from and towards cloud provider networks. To limit the impact of sending spoofed traffic towards cloud provider networks, we only targeted our own rented VPSes in the cloud providers under consideration and used the IP addresses of those VPSes as the spoofed source IP addresses in our queries (Sections 4.4 and 4.5). This way spoofed traffic (if not dropped) would arrive at a host under our own control. To further minimise the impact, we have only sent a handful of these spoofed queries.

Similarly when sending spoofed traffic from cloud provider networks, we used IP addresses of hosts under our own control (external host as well as another VPS under our control) as spoofed source addresses. Thus, any spoofed query that would leave the cloud network, would finally end up on one of our hosts. We believe the impact of spoofed traffic for transport networks to be minimal.

8.3 Rate Limiting

Deployment of DNS response rate limiting may reduce the impact of the attack models that we have introduced in this paper. As we mention in Section 4.5, the likelihood of such a mechanism being in place is low. Even if RRL is deployed, it can be bypassed in a number of ways. To validate whether vulnerable networks have RRL in place we would need to send a burst of DNS queries towards the DNS servers which are meant to serve clients relying on them. It is also likely that sending this kind of burst traffic would set off detection rules, alerting security teams. Since we wished to avoid any disruption on these systems, we have chosen not to perform such tests in this study.

8.4 Vulnerability Disclosure

Our study identifies a number of cloud providers to be vulnerable to the attack models explored in our research. We have conducted a coordinated vulnerability disclosure procedure to these providers. Almost all of the notified providers responded and worked with us to understand the issue. We have given the affected operators enough time to implement countermeasures before publishing our findings. To minimise harm, this paper describes our findings in a generic way so that they do not directly identify specific providers. A number of providers in question have already responded positively to our disclosure, aiming to deploy fixes in their networks, some have chosen to accept the current situation.

9 RELATED WORK

A number of existing studies have been conducted to classify open DNS resolvers as potential reflectors to be misused in R&A DDoS attacks. Kührer et al. [20] monitored multiple, UDP-based protocols that can be used to bring about amplification attacks. The authors classified amplifiers in terms of architectures and operating systems. Moreover, they investigated SAV adoption using a remote measurement setup that relied on misconfigured DNS proxies. This allowed them to reveal that about 2.7k autonomous systems lacked SAV at the time of their study, without looking into the types of these networks. In a later study, Kührer et al. conducted a large-scale measurement of open DNS resolvers [19]. They studied the landscape of resolvers over time and classified resolvers according to device type and running software, in addition to measuring the authenticity of responses that are returned to clients by open resolvers. As part of their study, the authors also looked at the top AS numbers in terms of open DNS resolver presence, and manually reasoned about the type of some of these networks. The authors found that many of the ASes offer end user services such as broadband, which suggests that at the very least some of the open resolvers in these networks run on consumer devices such as modems and routers. Park et al. [27] performed a study that shares some common ground with [19], as the authors investigated the behavior of open resolvers and quantified resolvers in terms of correct, incorrect, and even malicious responses. Our research for the most part differs from this group of studies. First, we are concerned with the likely capacity of the networks that open resolvers are in and focus on their potential contribution to DDoS attacks. For this reason, our study considers open resolvers that return a correct answer (from an application-layer perspective) and would be more appealing for attackers. Second, while the focus of our paper is on well-provisioned networks in particular, our approach to identifying open DNS resolver network type is more thorough. Third, while we do also look at open DNS resolver software, this is only a small part of our results.

Leverett and Kaplan [22] estimate a lower bound for a global R&A DDoS attack rate considering four UDP-based protocols. They leverage the speed measurements of the Measurement-Lab (MLab) [21] in their research. Our study reasons about the likely well-provisioned state – and hence network throughput capabilities – of reflectors. However, we rely on the IP intelligence dataset of IP2Location since MLab has a number of concerns regarding data quality as Leverett and Kaplan acknowledge [22]. Besides, as MLab relies on volunteers to perform speed tests, there are coverage concerns, which limits its applicability to our study.

The deployment of network ingress filtering on the Internet was studied in previous works, using various methodologies. Deccio et al. [8] studied the adoption of DSAV on the recursive DNS servers that appeared in DNS-OARC's DITL dataset [9] during their study period. The authors used various source IP address types in their measurement, such as multiple addresses from a prefix different than that of the destination, an IP address in the same prefix as the destination, a private IP address [29], same source and destination, and loopback address. They report that roughly half of the 62k autonomous systems that they test lack DSAV. Similarly, Korczyński et al. [18] studied DSAV deployment in the entire IPv4 address space by issuing spoofed DNS queries that use an IP address adjacent to the destination address. They report that more than 32k autonomous systems are vulnerable to spoofing of inbound traffic. We measure DSAV in this paper as well, but with two notable and important differences. First, our focus is on the DDoS potential, whereas the aforementioned studies focus on DNS cache poisoning attacks. Second, we focus on the misuse potential of well-provisioned networks (cloud providers) compared to the generic approach of existing work.

The Spoofer project by Luckie et al. [24] enables inferences about SAV on both the source and destination side. Spoofer relies on participants that voluntarily run the client software on their hosts. We explore SAV deployment by running hosts in the target networks (i.e., cloud providers) of our study, which is more suitable for this specific research. Next to this, we use Spoofer data to get an insight into OSAV adoption in datacenter-based networks. As we mention at the end of Section 5.2, only roughly 3% of datacenter-based networks are measured by the Spoofer project, which introduces a limitation into applicability of Spoofer data to our research.

Deccio et al. [7] conducted a measurement study on the RRL adoption on the authoritative nameservers of the root, TLDs and most-popular Web sites by Statvoo [31]. Conducting a similar study for the DNS infrastructure of concern in our study would give an insight into real-life misuse potential of these hosts. However, for ethical reasons, among others (see Sections-4.5 and 8.3), we have not explored RRL in this paper.

As part of our study, we identify and assess six attack models (largely novel) under which cloud infrastructures can be misused. Although existing works share some common ground with our testing methodology and a subset of the models [8, 18], to the best of our knowledge, no other studies have extensively focused on the role of cloud infrastructure in R&A DDoS attacks.

10 CONCLUSION

Open DNS resolvers have been a persistent source of concern for a long time when dealing with reflection & amplification distributed denial of service attacks. Traditional approaches consider all open resolvers to be equally threatening when it comes to R&A DDoS attacks. We have taken a different approach, postulating that rooting out reflectors in well-provisioned networks is more urgent. As such, we differentiate open DNS resolvers based on their network capacity. Our empirical study shows that roughly 12% of open resolvers are located in datacenters. Our classification of more than 300k of well-provisioned open resolvers stands to benefit more selective take-down efforts.

More importantly, we underpin that open DNS resolvers are only part of the problem to be solved. We identified and formalized six attack models in which the infrastructure of a network (with a focus on well-connected cloud networks) can be misused to bring about reflection-based DDoS attacks. This notably includes DNS infrastructure that is not fully shielded against misuse by external attackers while by design it is only meant to serve the customers internal to the network. To get an insight into the extent to which such threats exist on the Internet, we conducted a proof-of-concept assessment of the attack models on 19 public cloud providers. Our

experiments show that 3 major providers expose their DNS infrastructure to spoofed queries originated from external hosts by not implementing DSAV. This is while all of them to a high extent block spoofed traffic originated in their networks by deploying OSAV. Besides, 6 of the providers in our study expose their DNS resolver pool (which are meant to be a back-end infrastructure) to their clients. We engaged in coordinated vulnerability disclosure with the providers in question. A number of providers have already reacted positively to our disclosure, intending to deploy relevant fixes in their networks.

Our study can be extended in a number of directions which we leave as a future work. First, the resolver pool inference can be improved to extend the impact assessment for each of the applicable attack models. Next, open resolvers inside cloud networks can be leveraged to extend the DSAV deployment experiments of our paper. Finally, while we considered outreach regarding exposed cloud provider DNS infrastructure most-pressing, we also consider communicating open DNS resolvers (customer-operated) to cloud providers.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable feedback on our paper. We gratefully acknowledge CAIDA for providing us access to their infrastructure to carry out part of our experiment. This research is partially funded by the EU H2020 project CONCORDIA (#830927), by the SIDNfonds, and by the Comcast Innovation Fund.

REFERENCES

[1] Yehuda Afek, Anat Bremler-Barr, and Lior Shafir. 2020. NXNSAttack: Recursive DNS Inefficiencies and Vulnerabilities. In *Proceedings of the 29th USENIX Security Symposium*. 631–648.

[2] Amazon. 2013. Amazon Route 53 Developer Guide. Retrieved April 1, 2022 from https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/route53-dg.pdf

[3] Michael Bailey, David Dittrich, Erin Kenneally, and Doug Maughan. 2012. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. *IEEE Security and Privacy* 10, 2, 71–75. https://doi.org/10.1109/MSP.2012.52

[4] Fred Baker and Pekka Savola. March 2004. BCP 84: Ingress Filtering for Multihomed Networks. Online: https://tools.ietf.org/search/bcp84.

[5] CAIDA. 2020. Spoofer Project. Retrieved April 1, 2022 from https://www.caida.org/projects/spoofer/

[6] Censys. [n.d.]. Retrieved April 1, 2022 from "https://censys.io"

[7] Casey Deccio, Derek Argueta, and Jonathan Demke. 2019. A Quantitative Study of the Deployment of DNS Rate Limiting. In *2019 International Conference on Computing, Networking and Communications*. 442–447. https://doi.org/10.1109/ICCNC.2019.8685601

[8] Casey Deccio, Alden Hilton, Michael Briggs, Trevin Avery, and Robert Richardson. 2020. Behind Closed Doors: A Network Tale of Spoofing, Intrusion, and False DNS Security. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*. 65–77. https://doi.org/10.1145/3419394.3423649

[9] DNS-OARC. 2018. DITL Data. Retrieved April 1, 2022 from https://www.dns-oarc.net/oarc/data/ditl/

[10] Paul Ferguson and Daniel Senie. 2000. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827. https://doi.org/10.17487/RFC2827

[11] The Shadowserver Foundation. [n.d.]. Retrieved April 1, 2022 from "https://www.shadowserver.org"

[12] Gartner. [n.d.]. Cloud Infrastructure and Platform Services Reviews and Ratings. Online: https://www.gartner.com/reviews/market/public-cloud-iaas. https://www.gartner.com/reviews/market/public-cloud-iaas Accessed: Dec. 1, 2021.

[13] Google. 2022. Advanced VPC concepts, Google Cloud. Retrieved April 1, 2022 from https://cloud.google.com/vpc/docs/advanced-vpc

[14] Bradley Huffaker, Marina Fomenkov, and K Claffy. 2011. Geocompare: a comparison of public and commercial geolocation databases. *2011 ISMA Workshop on Active Internet Measurements*.

[15] IP2Location. [n.d.]. IP Address to IP Location and Proxy Information. Retrieved April 1, 2022 from https://www.ip2location.com/

[16] Mattijs Jonker, Alistair King, Johannes Krupp, Christian Rossow, Anna Sperotto, and Alberto Dainotti. 2017. Millions of Targets Under Atack: a Macroscopic Characterization of the DoS Ecosystem. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, Vol. Part F131937. 100–113. https://doi.org/10.1145/3131365.3131383

[17] Dan Kaminsky. 2008. Black Ops 2008: It's The End Of The Cache As We Know It. Retrieved April 1, 2022 from kurser.lobner.dk/dDist/DMK_BO2K8.pdf

[18] Maciej Korczyński, Yevheniya Nosyk, Qasim Lone, Marcin Skwarek, Baptiste Jonglez, and Andrzej Duda. 2020. Don't Forget to Lock the Front Door! Inferring the Deployment of Source Address Validation of Inbound Traffic. In *International Conference on Passive and Active Network Measurement*. Springer, 107–121. https://doi.org/10.1007/978-3-030-44081-7_7

[19] Marc Kührer, Thomas Hupperich, Jonas Bushart, Christian Rossow, and Thorsten Holz. 2015. Going Wild: Large-Scale Classification of Open DNS Resolvers. In *Proceedings of the 2015 ACM Internet Measurement Conference - IMC '15*. ACM Press, New York, USA, 355–368. https://doi.org/10.1145/2815675.2815683

[20] Marc Kührer, Thomas Hupperich, Christian Rossow, and Thorsten Holz. 2014. Exit from hell? Reducing the Impact of Amplification DDoS Attacks. In *Proceedings of the 23rd USENIX Security Symposium*. 111–125.

[21] Measurement Lab. 2009. Retrieved April 1, 2022 from https://www.measurementlab.net/

[22] Eireann Leverett and Aaron Kaplan. 2017. Towards estimating the untapped potential: a global malicious DDoS mean capacity estimate. *Journal of Cyber Policy* 2 (2017), 195–208. https://doi.org/10.1080/23738871.2017.1362020

[23] Franziska Lichtblau, Florian Streibelt, Thorben Krüger, Philipp Richter, and Anja Feldmann. 2017. Detection, Classification, and Analysis of Inter-Domain Traffic with Spoofed Source IP Addresses. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, Vol. Part F1319. 86–99. https://doi.org/10.1145/3131365.3131367

[24] Matthew Luckie, Ken Keys, Robert Beverly, Joshua A. Kroll, Ryan Koga, and K. Claffy. 2019. Network Hygiene, Incentives, and Regulation: Deployment of Source Address Validation in the Internet. In *Proceedings of the ACM Conference on Computer and Communications Security*. 465–480. https://doi.org/10.1145/3319535.3354232

[25] Marcin Nawrocki, Mattijs Jonker, Thomas C. Schmidt, and Matthias Waehlisch. 2021. The Far Side of DNS Amplification: Tracing the DDoS Attack Ecosystem from the Internet Core. In *Proceedings of the 2021 ACM Internet Measurement Conference*, Vol. 1. 419–434. https://doi.org/10.1145/3487552.3487835

[26] University of Oregon. [n.d.]. Route Views Project. Retrieved April 1, 2022 from http://www.routeviews.org

[27] Jeman Park, Aminollah Khormali, Manar Mohaisen, and Aziz Mohaisen. 2019. Where Are You Taking Me? Behavioral Analysis of Open DNS Resolvers. In *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 493–504. https://doi.org/10.1109/DSN.2019.00057

[28] Open Resolver Project. [n.d.]. Retrieved April 1, 2022 from https://web.archive.org/web/20200603050044/http://openresolverproject.org/

[29] Yakov Rekhter, B Moskowitz, Daniel Karrenberg, GJ de Groot, and Eliot Lear. 1996. RFC1918: Address Allocation for Private Internets. https://doi.org/10.17487/RFC1918

[30] Christian Rossow. 2014. Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In *Proceedings of the 2014 Network and Distributed Systems Security Symposium*. Internet Society, San Diego, 23–26. https://doi.org/10.14722/ndss.2014.23233

[31] Statvoo. [n.d.]. Website Discovery and Reviews. Retrieved April 1, 2022 from https://statvoo.com/top/sites

[32] Daniel R. Thomas, Richard Clayton, and Alastair R. Beresford. 2017. 1000 days of UDP amplification DDoS attacks. In *2017 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 79–84. https://doi.org/10.1109/ECRIME.2017.7945057

[33] Paul Vixie. 2013. On the Time Value of Security Features in DNS. Online: https://www.circleid.com/posts/20130913_on_the_time_value_of_security_features_in_dns/. https://www.circleid.com/posts/20130913_on_the_time_value_of_security_features_in_dns/ Accessed: Dec. 1, 2021.

[34] Paul Vixie and Vernon Schryver. 2012. *DNS Response Rate Limiting (DNS RRL)*. Technical Report. https://web.archive.org/web/20160307112057/http://ss.vix.su/~vixie/isc-tn-2012-1.txt