SMap: Internet-wide Scanning for Spoofing

Tianxiang Dai
ATHENE Center, Germany
Fraunhofer SIT, Germany

Haya Shulman
ATHENE Center, Germany
Fraunhofer SIT, Germany

ABSTRACT

To protect themselves from attacks, networks need to enforce *ingress filtering*, i.e., block inbound packets sent from spoofed IP addresses. Although this is a widely known best practice, it is still not clear how many networks do not block spoofed packets. Inferring the extent of spoofability at Internet scale is challenging and despite multiple efforts the existing studies currently cover only a limited set of the Internet networks: they can either measure networks that operate servers with faulty network-stack implementations, or require installation of the measurement software on volunteer networks, or assume specific properties, like traceroute loops. Improving coverage of the spoofing measurements is critical.

In this work we present the **Spoofing Mapper (SMap)**: the first scanner for performing *Internet-wide* studies of ingress filtering. SMap evaluates spoofability of networks utilising standard protocols that are present in almost any Internet network. We applied SMap for Internet-wide measurements of ingress filtering: we found that 69.8% of all the Autonomous Systems (ASes) in the Internet do not filter spoofed packets and found 46880 new spoofable ASes which were not identified in prior studies. Our measurements with SMap provide the first comprehensive view of ingress filtering deployment in the Internet as well as remediation in filtering spoofed packets over a period of two years until May 2021.

We set up a web service at https://smap.cad.sit.fraunhofer.de to perform continual Internet-wide data collection with SMap and display statistics from spoofing evaluation. We make our datasets as well as the SMap (implementation and the source code) publicly available to enable researchers to reproduce and validate our results, as well as to continually keep track of changes in filtering spoofed packets in the Internet.

CCS CONCEPTS

• **Security and privacy → Network security**.

KEYWORDS

Ingress Filtering, Spoofing, PMTUD, IPID, DNS

ACM Reference Format:
Tianxiang Dai and Haya Shulman. 2021. SMap: Internet-wide Scanning for Spoofing. In *Annual Computer Security Applications Conference (ACSAC '21), December 6–10, 2021, Virtual Event, USA*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3485832.3485917

ACSAC '21, December 6–10, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8579-4/21/12…$15.00
https://doi.org/10.1145/3485832.3485917

1 INTRODUCTION

Source IP address spoofing allows attackers to generate and send packets with a false source IP address impersonating other Internet hosts, e.g., to avoid detection and filtering of attack sources, to reflect traffic during Distributed Denial of Service (DDoS) attacks, to launch DNS cache poisoning, for spoofed management access to networking equipment and even to trigger services which can only be accessible to internal users [8, 11, 13, 32, 39]. The best way to prevent IP spoofing is by enforcing Source Address Validation (SAV) on packets, a practice standardised in 2000 as BCP38 [19]: *ingress filtering* for blocking inbound packets and *egress filtering* for blocking outbound packets sent from spoofed IP source addresses. In contrast to egress filtering which has been extensively measured in the last 15 years, only a couple of recent studies provided measurements on the extent of ingress filtering.

Ingress filtering. To enforce ingress filtering the networks should check the source address of an inbound packet against a set of permitted addresses before letting it into the network. Otherwise, the attackers using spoofed IP addresses belonging to the network can trigger and exploit internal services and launch attacks. For instance, by spoofing internal source IP addresses the attackers can obtain access to services, such as RPC, or spoofed management access to networking equipment [RFC3704], the attackers can cause DoS amplification by triggering the ICMP error messages from the attacked hosts to other internal hosts whose IP addresses the attacker spoofed. Enforcing ingress filtering is therefore critical for protecting the networks and the internal hosts against attacks. Nevertheless, despite efforts to prevent IP spoofing, it is still a significant problem. Attacks utilising IP spoofing remain widespread [8, 10, 18, 35, 38, 41].

How widespread is the ability to spoof? There are significant research and operational efforts to understand the extent and the scope of (ingress and egress)-filtering enforcement and to characterise the networks which do not filter spoofed packets; we discuss these in Related Work, Section 2. Although the existing studies and tools, such as the Open Resolver [34] and the Spoofer [5–7, 28, 30] projects, provide a valuable contribution for inferring networks which do not enforce spoofing, they are nevertheless insufficient: they provide a meager (often non-uniform) coverage of the Internet networks and are limited in their applicability as well as effectiveness.

SMap (The Spoofing Mapper). In this work we present the first Internet-wide scanner for networks that filter spoofed inbound packets, we call the Spoofing Mapper (SMap). We apply SMap for scanning ingress-filtering in more than 90% of the Autonomous Systems (ASes) in the Internet. The measurements with SMap show that more than 80% of the tested ASes do not enforce ingress filtering (i.e., 72.4% of all the ASes in the routing system), in contrast to 2.4% identified by the latest measurement of the Spoofer Project

[30]. The reason for this significant difference is the limitation of the previous studies of ingress filtering to a small set of networks.

Limitations of filtering studies. The measurement community provided indispensable studies for assessing "spoofability" in the Internet, and has had success in detecting the ability to spoof in some individual networks using active measurements, e.g., via agents installed on those networks [28, 34], or by identifying spoofed packets using offline analysis of traffic, e.g., [29, 30]. The need to install agents on networks or the ability to obtain traces only from some networks limits the studies to non-uniform coverage of the Internet. Therefore it is not clear how representative these statistics are. Unfortunately, this limitation to a small set of networks creates a bias in the assessments of the overall number of spoofable networks. The extrapolation from the small set of networks to the entire Internet typically result in assessment that at least 30% of the Internet networks do not filter spoofed packets [30, 32]. As we show, the number of spoofable networks is above 72% which is significantly higher than what was previous believed.

Requirements on Internet studies. The key requirements for conducting Internet studies upon which conclusions can be drawn include scalable measurement infrastructure, good coverage of the Internet and a representative selection of measurement's vantage points. We summarise the limitations of the previous studies below and in Table 1, and compare to SMap.

• *Limited coverage.* Previous studies infer spoofability based on measurements of a limited set of networks, e.g., those that operate servers with faulty network stack [26] or networks with volunteers that execute the measurement software [5–7, 28, 30, 34], or networks that agree to cooperate and volunteer their traffic logs for offline analysis, e.g., [30]. In contrast, the measurements with SMap use standard protocols supported by almost any network with Internet connectivity, for the first time providing studies of ingress filtering that cover the entire IPv4 space.

• *Limited scalability.* Previous approaches require installing agents, need to reproduce loops in traceroutes, or use misconfigurations in networks which limits their scalability. SMap is more scalable than any previous approach, since it merely exchanges requests/responses with networks using a fixed infrastructure of probers. The measurement infrastructure of SMap is not a function of the measured networks, hence adding more networks to the study does not require extending the measurement infrastructure.

• *Limited representativeness.* Volunteer or crowd-sourcing studies, such as the Spoofer Project [28], are inherently limited due to bias introduced by the participants. These measurements are performed using a limited number of vantage points, which are set up in specific networks, and hence are often not representative of the entire Internet. Increasing the coverage and selecting the networks more uniformly is imperative for collecting representative data; [22] showed that the measured network significantly influences the resulting data as well as the derived conclusions. Since SMap measures almost all the IPv4 networks the results are representative of the entire Internet.

• *Limited stability.* Current measurement studies use unstable infrastructures: volunteers running agents can reinstall computers or move to other networks [34]; misconfigured servers [28] (e.g., with open resolution or with faulty network stack) can be updated – all causing the network to "disappear from the radar" although

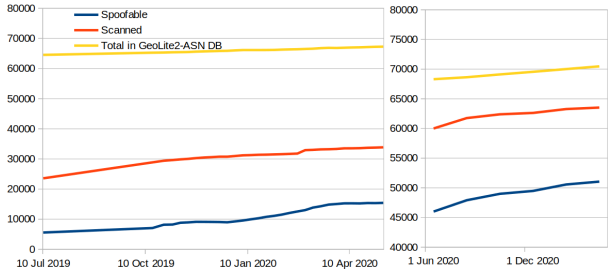

Figure 1: SMap measurements between July'19 and May'21. Domain-based (left) and IPv4-based (right).

it may still be spoofable. Hence, longitudinal studies, such as the Spoofer Project, are biased by the stability of the vantage points, and cannot accurately track deployment of ingress filtering in individual networks. A few works [34] pointed out that the instability of the infrastructure creates discrepancy in the statistics. In particular, repeating the measurements a few weeks later generates other different results.

What SMap improves. The infrastructure of SMap is more stable than those used in previous studies, e.g., we do not risk volunteers moving to other networks. Our measurements do not rely on misconfigurations in services which can be patched, blocking the measurements. The higher stability also allows for more accurate reproduction and validation of our datasets and results, and enables to perform reliable longitudinal studies. We ran ingress filtering measurements with SMap every week over a period of two years (between 10 July 2019 and 10 May 2021). Our results plotted in Figure 1 demonstrate that the number of spoofable ASes is stable and proportionally increases with the growth in the overall number of ASes in the Internet. This is in contrast to previous studies, e.g., [27–29], in which a repeated evaluation even a week later provided different statistics. Our two year long measurements between 2019 and 2021 of more than 90% of Internet's ASes we found 50,023 new ASes that do not enforce ingress filtering, which were not known before, and confirmed all the other ASes that were found spoofable in prior studies.

Ethical Considerations. Internet-wide scans are important for security research [16, 31] and have proven valuable in improving the security landscape of the Internet, including exposing new vulnerabilities, tracking adoption of defences. Nevertheless, Internet-wide scans introduce also ethical challenges. We communicated with network operators to understand and consider the ethical implications of Internet-wide scans. We identified two issues as particularly important for our measurements: *traffic load* and *consent*.

• *Traffic load.* Network scans, such as [16, 26, 31], require exchanging packets with a large number of Internet networks as well as IP addresses inside the networks. To avoid scanning the Internet we periodically download a dataset of a full scan of the Internet done by Sonar.

• *Consent of the scanned.* It is often impossible to request permission from owners of all the tested networks in advance, this challenge similarly applies to other Internet-wide studies [15, 16, 26, 31]. Like the other studies, [15, 16], we provide an option to opt out of our scans. To opt out the network has to provide either its network

block (in CIDR notation), domain or ASN through the contact page at https://smap.cad.sit.fraunhofer.de. Performing security scans is important - the networks that do not enforce filtering of spoofed packets pose a hazard not only to their operators but also to their users, customers and services, as well as other networks. Due to the importance of identifying such networks, in their recent study [30] even make public the ("name-and-shame") lists of providers with missing or misconfigured filtering of spoofed packets; [30] also discuss stronger measures against spoofable networks, including liability for damages, and various types of regulation. Inevitably, due to the risks that such networks pose to the Internet ecosystem, it is of public interest to know who those networks are. We do not make the identity of the networks, that do not filter spoofed packets, publicly available, but inform the general public on the fraction of such networks and provide their characterisation (i.e., size, geo-location, business type) in Section 5.

Undoubtedly, filtering spoofed packets is critical and networks have to deploy best practices, such as BCP38 [19] and BCP84 [3], to ensure security of the Internet ecosystem. Understanding the extent of filtering is also significant for devising future policies, defence mechanisms or estimating threats and risks to attacks.

Organisation. Our work is organised as follows: we compare our study and SMap to related work in Section 2. In Section 3 we present the design and the implementation of SMap and the measurement techniques that it uses. In Section 4 we report on the data collected with SMap and the statistics that we derived from it. We characterise the networks which we found not to enforce ingress filtering in Section 5. We conclude this work in Section 6.

2 OVERVIEW OF SPOOFING STUDIES

2.1 Egress vs. Ingress

Although there are a few studies of ingress filtering, most studies of spoofing focus on egress filtering. What can be inferred from egress filtering on igress filtering and vice versa?

In their recent measurement of ingress and egress filtering [30] conclude that filtering of inbound spoofed packets is less deployed than filtering of outbound packets, despite the fact that spoofed inbound packets pose a threat to the receiving network. [25] analysed the networks from Spoofer and open resolver projects and found that 74% of the networks that do not filter outbound spoofed packets, do not filter inbound spoofed packets. A more recent study [24] of 515 ASes found that ingress filtering of inbound spoofed packets is more widely deployed than egress filtering of outbound packets.

The correlation between egress and ingress filtering in previous work shows that the measurements of ingress filtering also provide a lower bound on the number of networks that enforce egress filtering of spoofed outbound packets. Therefore our results on networks that do not enforce ingress filtering imply that at least as many networks do not perform egress filtering.

2.2 Measurements of Spoofability

Measurements of networks that filter spoofed packets in the Internet was previously done using *network traces* or using *vantage points*. We summarise the results of the previous studies in Table 1, and briefly explain them below.

Vantage Points. Measurement of networks which do not perform egress filtering of packets with spoofed IP addresses was first presented by the Spoofer Project in 2005 [5]. The idea behind the Spoofer Project is to craft packets with spoofed IP addresses and check receipt thereof on the vantage points operated by the *volunteers*, i.e., participants who run a "spoofer" software provided by the authors. Based on the data collected by the Spoofer Project many reports were published providing statistics on the deployment of egress filtering in the Internet [6, 7, 28, 30]; we list the statistics in Table 1.

The downside of this approach is that the Spoofer Project requires users to download, compile and execute a software - which also needs administrative privileges to run - once per measurement. This requires not only technically knowledgeable volunteers that agree to run untrusted code, but also networks which agree to operate such vantage points on their premises. [22] argues that extending the limited coverage of the Spoofer Project is difficult: the operators are unlikely to volunteer or conduct measurements that could leak a negative security posture of their networks, including lack of support of BCP38 [19]. Hence, [22] propose that the most viable method to measure filtering of spoofed packets in more networks is by crowd-sourcing. In 2018 [28] performed a one-time study of the Spoofer Project by renting a 2,000 EUR crowd-sourcing platforms with workers that executed the Spoofer software over a 6 weeks period. Their measurements included additional 342 ASes which were not covered by the Spoofer Project previously. Crowd-sourcing studies, in addition to being expensive, are also limited by the networks in which workers are present and do not provide longitudinal and repetitive studies that can be validated and reproduced.

In a recent longitudinal data analysis by the Spoofer Project [30] the authors observed that despite increase in the coverage of ASes that do not perform ingress filtering in the Internet, the test coverage across networks and geo-locations is still non-uniform.

Closely related to *volunteers* is the vantage points measurements with *faulty or misconfigured servers*. [34] noticed that some DNS resolvers do not change the source IP addresses of the DNS requests that they forward to upstream resolvers and return the DNS responses using the IP addresses of the upstream resolvers - a problem which the authors trace to broken networking implementations. [26] used this observation to measure egress filtering in networks that operate such misconfigured DNS resolvers. Such measurements are limited only to networks which operate DNS servers with broken networking implementations: out of 225,888 networks that [26] measured, they could find such DNS servers only in 870 networks.

Since the Open Resolver and the Spoofer Projects are the only two infrastructures providing vantage points for measuring spoofing - their importance is immense as they facilitated many research works analysing the spoofability of networks based on the datasets collected by these infrastructures. Nevertheless, the studies using these infrastructure, e.g., [22, 30], point out the problems with the representativeness of the collected data of the larger Internet. Both projects (the Spoofer and the Open Resolver) acknowledged the need to increase the coverage of the measurements, as well as the challenges for obtaining better coverage and stable vantage points.

Network Traces. To overcome the dependency on vantage points for running the tests, researchers explored alternatives for

Study	Coverage (scanned ASes)	Spoofable ASes	Type	Year	Longitudinal	Reproducible	Scalable
Spoofer Project [5]	202 of 18,000 (1.1%)	52	Egress	2005	✓	X	X
Spoofer Project [7]	1,586 of 44,000 (3.6%)	390	Egress	2013	✓	X	X
Misconfigured servers [26]	2,692 of 48,000 (5.6%)	870	Egress	2014	X	✓	✓
Traceroute [29]	1,780 of 56,000 (3.2%)	703	Ingress	2017	X	X	(✓)
IXP traces [27]	700 of 56,000 (1.3%)	393	In & Eg	2017	X	X	X
Amazon Turk Spoofer Project [28]	784 of 56,000 (1.4%)	48	Egress	6w. in 2017	✓	X	X
Spoofer Project [30]	5,178 of 66,000 (7.8%)	1,631	In & Eg	2019	✓	X	X
SMap	63,522 of 70,468 (90%)	51,046	Ingress	2019-21	✓	✓	✓

Table 1: Comparison between SMap and other studies.

inferring filtering of spoofed packets. A recent work used loops in traceroute to infer ability to send packets from spoofed IP addresses, [29]. This method detects lack of ingress filtering only on provider ASes (i.e., spoofable customer ASes cannot be detected). The study in [29] identified loops in 1,780 ASes, which is 3.2% of all the ASes, and 703 of the ASes were found spoofable. Although a valuable complementary technique for active probes with vantage points, this approach has significant limitations: in the absence of loops ingress filtering cannot be inferred, alternately a forwarding loop in traceroute does not imply absence of filtering at the edge, since a loop resulting from a transient misconfiguration or routing update can occur anywhere in the network. Therefore, to identify a lack of ingress filtering reliably one needs to detect a border router and, more importantly, the traceroute loops need to be reproduced - a difficult problem in practice. Furthermore, reproducing or validating the dataset after some time is virtually impossible as the odds for failures rapidly increase. Running traceroutes is also challenging: black-holes in traceroutes, whereby the routers do not respond to probes or when routers have a limit for ICMP responses, are common in Internet [33].

[27] developed a methodology to passively detect spoofed packets in traces recorded at a European IXP connecting 700 networks. The limitation of this approach is that it requires cooperation of the IXP to perform the analysis over the traffic and applies only to networks connected to the IXP. Allowing to identify spoofing that defacto took place, the approach proposed in [27] misses out on the networks which do not enforce filtering but which did not receive packets from spoofed IP addresses (at least during the time frame in which the traces were collected).

A range of studies analysed network traces for ingress filtering using IP address characteristics [4, 9, 10, 14, 36], or by inspecting on-path network equipment reaction to unwanted traffic, [44]. In addition to a limited coverage, the studies do not support longitudinal and repeating data collection and analysis, and cannot be reproduced as they do not make the datasets of their studies public.

3 SCANNING FOR SPOOFABLE NETWORKS

3.1 Dataset

SMap architecture consists of two parts: dataset scan and ingress filtering scan. The dataset scan collects the popular services using two methods: domain-based scan and IPv4 based scan. In IPv4 scan to locate the services SMap probes every IP, checking for open ports that correspond to the services that we need; for instance, port 25 for Email, 53 for DNS, 80/443 for Web. To reduce the traffic volume of the scan, instead of probing each IP address for target ports, SMap enables also query of the input domains for services. For every domain, it queries the IP and hostname of the services, e.g., (A, MX) for Email server, A for Web server, (A, NS) for name server.

3.2 Methodology

The measurement methodology underlying SMap uses active probes, some sent from spoofed as well as from real source IP addresses to popular services on the tested networks. The spoofed source IP addresses belong to the tested networks (similarly to the Spoofer Project [5]). The idea behind our methodology is that if the packets with spoofed addresses reach the services in the tested networks, they trigger a certain action. This action can be measured remotely. If the action was not triggered, we conclude that spoofed packets did not reach the service.

We develop three techniques to detect if networks filter spoofed traffic based on our methodology: DNS lookup, IPID and PMTUD based. Using popular services ensures that our measurements apply to as many Internet networks as possible.

SMap consists of the orchestrator which coordinates and synchronises the prober hosts. The prober hosts receive the dataset of networks to be scanned for spoofability from the orchestrator. The probers then run IPID, PMTUD and DNS lookup tests against the services on the dataset list. SMap applies one test at a time for each AS in the dataset. Each successful test indicates that packets from a spoofed IP address reached the destination on the target network, implying that the target AS does not filter spoofed packets. On the other hand, a failed test may indicate that one of the ASes on the path between the probers and the service on the target AS may be filtering spoofed packets.

The results from the tests are stored in the backend database. The GUI displays the results of the measurements at https://smap.cad.sit.fraunhofer.de. We next explain each measurement technique. In our measurements in Section 4 we compare the success and applicability of each technique.

3.3 IPID

Each IP packet contains an IP Identifier (IPID) field, which allows the recipient to identify fragments of the same original IP packet.

The IPID field is 16 bits in IPv4, and for each packet the Operating System (OS) at the sender assigns a new IPID value. There are different IPID assignment algorithms which can be categorised as: random and predictable. Predictable category uses either a global counter or multiple counters per designation IP address, such that the counter is incremented in predictable quotas. Random category selects each IPID value at random from a pool of values.

Recent work showed that even TCP traffic gets fragmented under certain conditions [12]. Fragmentation has long history of attacks which affect both the UDP and TCP traffic [21, 23, 40].

Methodology. We use services that assign globally incremental IPID values. The idea is that globally incremental IPID [RFC6864] [42] values leak traffic volume arriving at the service and can be measured by any Internet host. Given a server with a globally incremental IPID on the tested network, we sample the IPID value (send a packet to the server and receive a response) from the IP addresses controlled by us. We then generate a set of packets to the server from spoofed IP addresses, belonging to the tested network. We probe the IPID value again, by sending packets from our real IP address. If the spoofed packets reached the server, they incremented the IPID counter on the server - an event which we infer when probing the value from our real IP address the second time.

The challenge here is to accurately probe the increments rate of the IPID value (caused by the packets from other sources not controlled by us), in order to be able to extrapolate the value that will have been assigned to our second probe from a real source IP. This allows us to infer if the spoofed packets incremented the IPID counter.

Identifying servers with global IPID counters. We send packets from two hosts (with different IP addresses) to a server on a tested network. We implemented probing over TCP SYN, ping and using requests/responses to Name servers and we apply the suitable test depending on the server that we identify on the tested network. If the responses contain globally incremental IPID values - we use the service for ingress filtering measurement with IPID technique. We located globally incremental IPID in 63.27% of the measured networks. There are certainly more hosts on networks that support globally incremental IPID values, yet our goal was to validate our measurement techniques while keeping the measurement traffic low - hence we avoided scanning the networks for additional hosts and only checked for Web, Email or Name servers with globally incremental IPID counters via queries to the tested domain.

Statistics of IPID values distribution among tested servers are plotted in Figure 2. When ICMP is filtered, it results in ERROR, when run with TCP, the IPID values are often zero (i.e., ZERO IPID in graph) in Figure 2. To improve coverage of the IPID technique we merge the ICMP&TCP and ICMP&UDP results for each server in our measurements.

Measuring IPID increment rate. The traffic to the servers is stable and hence can be predicted, [43]. We validate this by sampling the IPID value at the servers which we use for running the test. One example evaluation of IPID sampling on one of the busiest servers is plotted in Figure 3. In this evaluation we issued queries to a Name server at `69.13.54.XXX` during three minutes, and plot the IPID values received in responses in Figure 3 - the identical patterns demonstrate predictable increment rates. Which means that the traffic to the server arrives at a stable rate.

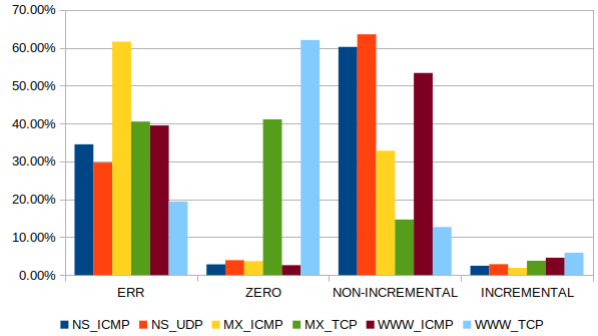

Figure 2: IPIDs on servers in dataset.

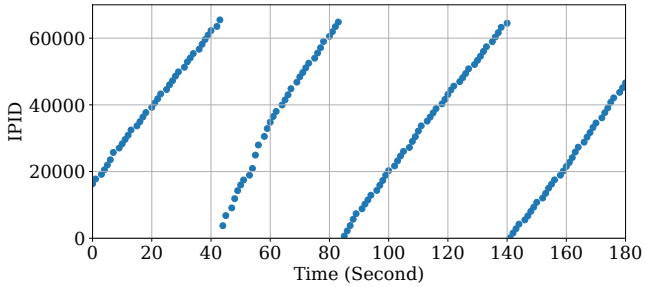

Figure 3: IPID of Name server 69.13.54.XXX during 180sec.

Accuracy of IPID measurements. The IPID techniques are known to be difficult to leverage, requiring significant statistical analyses to ensure correctness. Recently, [17, 37] developed statistical methods for measuring IPID. However, in contrast to our work, the goal in [17, 37] is different - they use IPID to measure censorship and have additional sources of inaccuracy, which do not apply to our measurements: (1) the measurements are applied against client hosts, which results in significantly higher noise than our measurements against servers - the clients move between networks, change IP addresses, the clients are located behind intermediate devices, such as Network Address translators (NAT) and firewalls - which also prevents direct measurements; (2) inaccuracies in geolocation tools, which do not apply to our study since we do not need to know the location to measure ingress filtering, (2) additional network mechanisms (anycast, rerouting, traffic shaping, transient network failures). All these can only cause us to classify the server as not 'testable', but do not impact 'spoofable' outcomes. Furthermore, the IPID measurement methods in prior workss use TCP-RST packets to increment IPID, which are often blocked in firewalls. In contrast, we use packets which are not blocked such as DNS queries or TCP-SYN.

Inferring spoofing. We use the following components: the prober at IP address `7.7.7.7` and a server at IP address `1.2.3.7` that uses globally incremental IPID, illustrated in Figure 4. Using the prober at `7.7.7.7`, we measure the value of the IPID and the rate at which IPID increments. We use linear regression with Ordinary Least Square (OLS) method to estimate the relation between IPID

and timestamp t. Since IPID is incremental, it holds: $IPID = a * t + b + \epsilon, \epsilon \sim N(0, \sigma^2)$

We send N probes to `7.7.7.7` (in step (1)). With N probes, we can estimate a, b and σ using OLS method in step (2). In step (3) in Figure 4 we send a set of $M = 6*\sigma$ packets from a spoofed source IP address `1.2.3.6` (belonging to the probed network). In step (4) at time T_{M+N+1} we sample the IPID value $Z = IPID^{real}_{M+N+1}$ from the server from the prober's real IP address `7.7.7.7` - this is needed in order to receive the response. We check the IPID value Z in step (5) in Figure 4. Taking the linear regression model into consideration, we can calculate $IPID^{esti}_{M+N+1}$ at time T_{M+N+1}. If the M spoofed packets are filtered, according to *3-sigma rule*, there is a 99.73% possibility that: $IPID^{esti}_{M+N+1} - 3 * \sigma \leq Z \leq IPID^{esti}_{M+N+1} + 3 * \sigma$. However, if the spoofed packets are not blocked, a.k.a. there is no ingress filtering, the IPID counter should have an additional increment of M. Thus $Z > IPID^{esti}_{M+N+1} + 3 * \sigma$, which is also $Z > IPID^{esti}_{M+N+1} + M/2$.

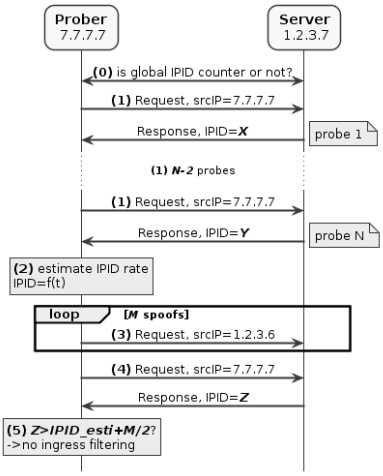

Figure 4: Sequence diagram for IPID technique.

We define outcomes of a test with IPID technique as *spoofable, applicable, non-applicable, N/A*; see Table 2. The IPID technique is not applicable if the IPID counter is constant zero or if the IPID counter is not globally incremental.

Category	IPID	PMTUD	DNS
Spoofable	no filtering	no filtering	no filtering
Applicable	server w/globally incremental IPID	host supports PMTUD	has DNS server
Non-applicable	random IPID or per-dest IPID or IPID=0	(DF≡0 & MF≡0) or (DF≡1 or MF≡1) & no change	no DNS server found
N/A	host unreachable or firewall or packet loss or load balancer	host unreachable or misconfigured service or firewall or packet loss	

Table 2: Outcomes of tests.

3.4 PMTUD

Path Maximum Transmission Unit Discovery (PMTUD) determines the MTU size on the network path between two IP hosts. The process starts by setting the `Don't Fragment (DF)` bit in IP headers. Any router along the path whose MTU is smaller than the packet will drop the packet, and send back an ICMP Fragmentation Needed / Packet Too Big (PTB). The payload of the ICMP packet contains the IP header and the first 8 bytes of the original packet that triggered the error as well as the MTU of the router that sent the ICMP message. After receiving an ICMP PTB message, the source host should either reduce its path MTU appropriately or unset the DF bit.

A study of CAIDA datasets in 2017 found 3M ICMP fragmentation needed packets sent by routers in the Internet, with about 1K routers sending ICMP error message with next hop MTU of less than 500 Bytes [20].

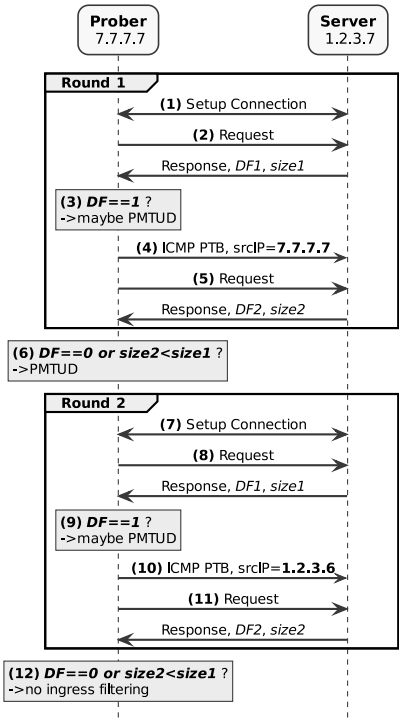

Figure 5: Sequence diagram for PMTUD technique.

Methodology. The core idea of the Path MTU Discovery (PMTUD) based tool is to send the ICMP Packet too Big (PTB) message from a spoofed source IP address, belonging to the tested network, and in the 8 bytes payload of the ICMP to insert the real IP address belonging to the prober. If the network does not enforce ingress filtering, the server will receive the PMTUD message and will reduce the MTU to the IP address specified in the first 8 bytes of the ICMP payload. We first probe the MTU to a service on the tested network, then send ICMP PTB from a spoofed IP address. If the packet arrives at the service, it will reduce the MTU to our prober, and we will identify this event in the next packet from the service -

this event implies that the tested network does not apply ingress filtering.

Identifying servers that support PMTUD. We measured networks that support PMTUD (i.e., do not filter ICMP Fragmentation Needed (Type 3, Code 4) messages), and found that 85.92% of the tested networks support PMTUD.

Inferring spoofing. The PMTUD test is illustrated in Figure 5. We establish a TCP connection to a server on the tested network. Then we send Request1 and receive Response1. If DF bit is not set, the server does not support PMTUD. Otherwise, we send an ICMP PTB with smaller MTU. Following that, we request again and get Response2. If $DF_1 == 1$ *and* $(DF_2 == 0$ *or* $size_2 \leq size_1)$, the server supports PMTUD. Now we can proceed to test if ingress filtering is enforced. We spoof an ICMP PTB with smallest MTU, using server's neighbour IP as source IP address. Once that is done, we make another request. The server is not protected by ingress filtering if following condition applies: $size_3 \leq size_2$ *or* $(DF_2 == 1$ *and* $DF_3 == 0)$.

We define outcomes of a test with PMTUD technique as *spoofable, applicable, non-applicable, N/A*; see rightmost column in Table 2.

3.5 DNS Lookup

DNS provides lookup services to networks. Upon receiving a DNS request, the resolver performs the lookup of the requested domain name and returns the response with the requested record.

Methodology. We send a DNS request to the tested network from a spoofed IP address belonging to the tested network. If the network does not enforce ingress filtering, the request will arrive at the DNS resolver on that network. A query from a spoofed source IP address will cause the response to be sent to the IP address from which the request was sent, i.e., the spoofed IP address. Since we do not control the spoofed IP address, we will not be able to observe this event and hence will not be able to infer if the DNS resolver received our request or if the request was filtered due to spoofing. To obtain insights into the traffic arriving at the resolver in the tested network we utilise the payload of the DNS request: the query contains the domain which we own, set up on Name servers that we control. Namely, eventhough the response from the DNS resolver will be returned to the spoofed IP address and will not be received by us, the DNS request will be issued to our Name servers, which is an indication that the DNS resolver on the tested network received our DNS request, sent from spoofed IP address.

Identifying DNS resolvers. The main challenge here is to locate the DNS resolvers within a domain/network and to trigger a DNS request to our Name servers. We use Email service in the target networks (retrieved via the MX type request in the target domain) to find the DNS resolvers. We send an email to target domain's Email server from one of our unique subdomains with a non-existing recipient set in the destination. This causes the Email server on the tested network to generate a Delivery Status Notification (DSN) error message [RFC3464] to our Email server. To be able to send us the DSN, the Email server will request the resolver on the tested network, to provide it the MX and A/AAAA records of our Email exchanger. At the same time, it may also trigger anti-spam checking, which requests (SPF/TXT, PTR, DKIM, DMARC)-type records in domains under our control. By monitoring the DNS queries at our Name servers, we collect the IP addresses of the resolvers. Using this methodology we identified 49,252 DNS resolvers in 7,141 networks. However, in our regular IPv4 scan, to reduce Email traffic in the Internet, we use the list of servers with UDP port 53 open from Project Sonar as input.

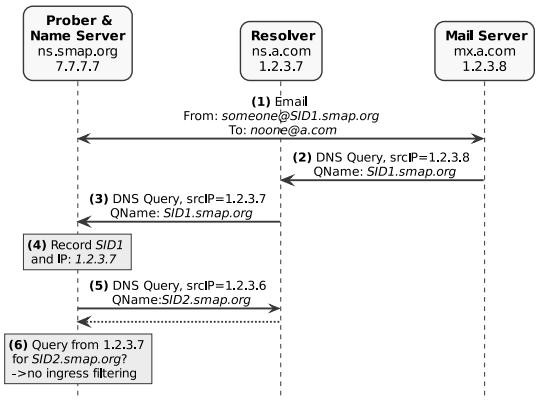

Figure 6: Sequence diagram for DNS lookup technique.

Inferring spoofing. Given a DNS resolver at IP `1.2.3.7`, we send a DNS query to `1.2.3.7` port 53 asking for a record in domain under our control. The query is sent from a spoofed source IP address belonging to the tested network. We monitor for DNS requests arriving at our Name server. If a query for the requested record arrives from `1.2.3.7`, we mark the network as not enforcing ingress filtering. The process is illustrated in Figure 6, steps (1-4) locate the IP address of the DNS resolver, and steps (5,6) test for ingress filtering on that network.

4 INTERNET MEASUREMENTS

In this section we report on our Internet-wide measurement of ingress filtering with SMap. Our dataset collection with SMap has been initiated on July 2019 continually collected data over a period of one year, of over 6M domains and an entire IPv4 address block.

4.1 Dataset

SMap first collects the dataset of services.Our dataset is constructed as follows: we periodically download the entire IPv4 scan from Sonar Project [2]. We use the scan results on UDP port 53 as input for Name servers and DNS resolvers, scan data on TCP port 25 for Mail servers and scan results on TCP port 80 for Web servers. Besides, we also make use of forward DNS responses and reverse DNS responses from Sonar Project to help find hostnames of servers. In the latest dataset from Sonar, we have services hosted in 63,522 ASes (Table 3) with 4,256,598 DNS servers in 38,838 ASes, 16,478,938 Email servers in 38,937 ASes, and 62,455,254 Web servers in 61,535 ASes; see Table 4.

4.2 Ingress Filtering Results

Domain-scan and IPv4-scan both show that the number of spoofable ASes grows with the overall number of the ASes in the Internet, see Figure 1. Furthermore, there is a correlation between fraction

Technique_Service	Spoofable		Applicable		Non-Applicable	N/A	Total ASes
IPID_NS	8,752	23.07%	12,056	31.78%	25,881	25,585	63,522
IPID_MX	4,355	21.48%	6,861	33.84%	13,416	43,245	63,522
IPID_WWW	30,963	51.83%	39,370	63.27%	22,891	2,608	63,522
IPID_ANY	32,248	56.25%	41199	67.52%	22,853	1,299	63,522
PMTUD_NS	9,054	24.16%	11,592	30.93%	25,885	26,045	63,522
PMTUD_MX	23,078	68.69%	27,127	80.74%	6,471	29,924	63,522
PMTUD_WWW	41,959	76.91%	47,524	87.11%	7,034	8,964	63,522
PMTUD_ANY	43,473	75.98%	49,161	85.92%	8,053	6,308	63,522
DNS lookup	25,407	40.00%	44,577	70.18%	-	-	63,522
ANY	51,046	80.90%	58,432	92.61%	4,662	428	63,522

Table 3: Collected data and analysis per AS view.

	Name Server	Email Server	Web Server
#IPs	4,256,598	16,478,938	62,455,254
#Blocks	697,851	748,406	3,207,393
#Prefixes	229,981	217,334	542,983
#ASes	38,838	38,937	61,535

Table 4: Servers in tested networks.

of scanned domains and ASes. Essentially the more domains are scanned, the more ASes are covered, and more spoofable ASes are discovered; see Figure 7. This result is of independent interest as it implies that one can avoid scanning the IPv4 and instead opt for domains-scan, obtaining a good enough approximation. This not only reduces the volume of traffic needed to carry out studies but also makes the study much more efficient.

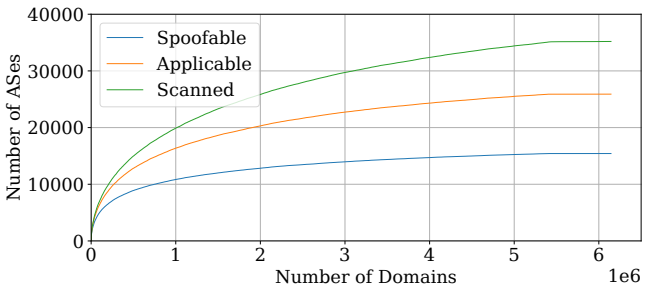

Figure 7: As we scan more domains, we cover more ASes and discover more spoofable ASes.

Further, to avoid single point of failure it is recommended that the Name servers of a domain are hosted in multiple networks. This is also our observation when correlating between domains and ASes. Essentially we find that when testing one domain for each server we can obtain different results, depending on the AS that the server is hosted on.

The results of the ingress filtering measurements with SMap are summarised in Table 3. The techniques that we integrated into SMap (IPID, PMTUD, DNS lookup) were found applicable to more than 92% of the measured ASes. Using SMap we identified 80%

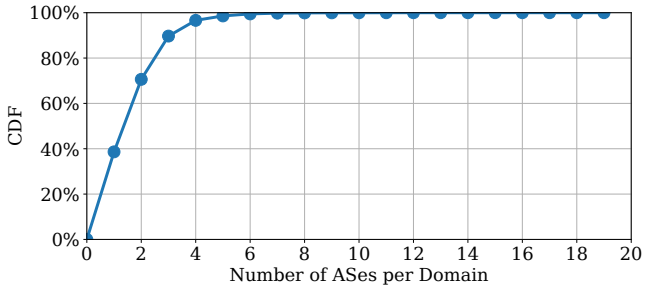

Figure 8: Fraction of domains hosted in multiple ASes. We check how many ASes host services of one domain: 70% of the domains are hosted in one or two ASes.

of the ASes that do not enforce ingress filtering. In what follows we compare the effectiveness of the techniques, explain causes for false negatives and failures. In the rest of this section we explain and analyse the applicability of our results and the success of the different techniques, discuss errors and compare to the results in previous studies.

4.3 Applicability and Success

As can be seen in Table 3 the most applicable technique is PMTUD against Web servers, which applies to a bit more than 87% of the ASes, yielded the highest fraction of spoofable ASes. This is not surprising, since the number of web servers is much larger than the others and it is recommended not to block ICMP to Web servers to allow for path MTU discovery.

We next compare the success and applicability of tests with PMTUD and IPID techniques against Email, Name and Web servers. In order to compare the effectiveness of the PMTUD and IPID measurement techniques as well as their applicability, we define the spoofable and applicable rates, as follows:

$$Rate_{spoofable} = \frac{N_{spoofable}}{N_{total} - N_{NA}}, Rate_{applicable} = \frac{N_{applicable}}{N_{total} - N_{NA}}$$

The spoofable rate reflects the fraction of the networks found not to apply ingress filtering and the applicable rate means applicability of the test technique. The coverage of each of the three techniques for different types of servers (Web, Name, and Email) is plotted in Figure 9.

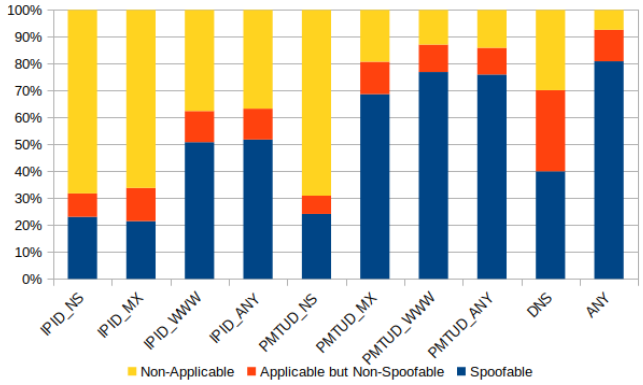

Figure 9: Coverage of the measurement techniques.

Figure 9 shows that PMTUD technique (listed as "PMTUD_ANY" in Figure 9) has a better test rate than either of the IPID and DNS tests, which indicates that PMTUD is still widely supported. Between the other two, DNS test has a slightly higher applicability than IPID test, which shows that globally sequential IPID is less supported now. In Figure 11 we similarly see that the fraction of spoofable networks that can be fonud through IPID and PMTUD is higher than when measured with the other methodologies; Figure 11 plots the networks found spoofable via IPID vs PMTUD excluding "N/A" networks.

In general, tests against Web servers have a higher applicability rate than the tests with Email or DNS servers, regardless of which technique was used (IPID or PMTUD). The number of Web servers is much larger than the others. It is much easier to setup a Web server than Email server or DNS server. Considering that DNS servers and Email servers are more likely to be hosted by providers, they also have higher probability to get new system updates. Furthermore, we find that when a Web server is not available ("N/A"), both Email and DNS servers cannot be tested, either. This also results in much higher N/A outcomes for tests against Email and DNS servers as opposed to Web servers.

The higher applicability of the tests against web servers also correlates with a higher number of spoofable networks. In Figure 10, we show the relationships between the applicability of SMap measurement techniques to different services and the overlap between them.

4.4 Errors

We define the result of SMap evaluation successful (i.e., true positive) if at least one of the three tests outputs that the tested network does *not* filter spoofed packets: either the IPID value on the server in the tested network was incremented as expected (IPID test) or we receive a query at our domain (DNS test) or the server on the tested network reduced the MTU of the packets sent to us (PMTUD test). When either of the three techniques provides a positive result, we mark the network as *not filtering*.

SMap does not make mistakes when reporting a network as not filtering. However, it can have false negatives: when the scan does not report network as not filtering when a network does not filter spoofed packets.

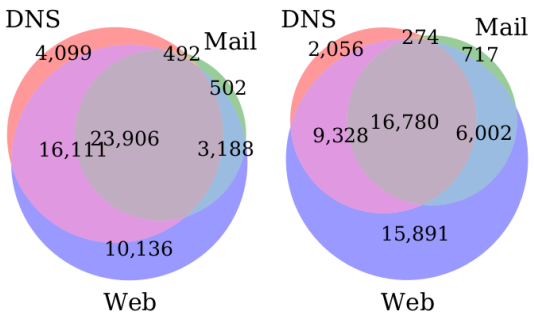

Figure 10: Number of Applicable (left) and Spoofable (right) ASes according to service type.

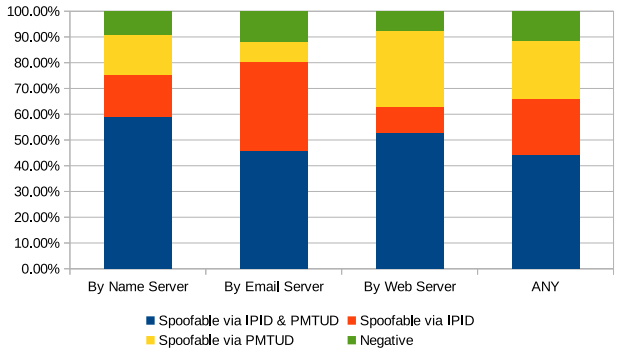

Figure 11: Comparison of spoofability via IPID and PMTUD.

4.4.1 No False Positives. Our techniques are not susceptible to false positives, that is, classification of the tested network as filtering spoofed packets when in fact it does not do so. This is a side effect of our methodology - only when spoofing is not filtered will the "test action" be triggered.

IPID technique. When spoofing is not filtered the counter on the server will be incremented - which is the test action. At the probing phase the counter's value will equal or large than the expected value after the increment phase. The repeated measurements ensure that we do not accidentally interpret noise (i.e., packets from other sources to the same server) as lack of ingress filtering.

DNS technique. When spoofing is not filtered the DNS resolver on the tested network will receive a DNS request from a spoofed IP address to our domain. Hence a query at our domain is the test action that spoofed packets are not filtered.

PMTUD technique. Reduction of the MTU of the packets sent from the test server to our network is the action which indicates that spoofing filtering is not enforced.

4.4.2 False Negatives. False negatives in our measurements mean that a network that does not perform filtering of spoofed packets is not marked as such. We next list the causes of false negatives for each of our three techniques. Essentially the false negatives cannot be resolved, and therefore our measurement results of networks that enforce ingress filtering introduce a a lower bound. The networks that we classify as those that do not apply ingress filtering -

definitely allow packets from spoofed IP addresses into the network. The networks which were not classified as "not enforcing ingress filtering", could still be "not enforcing ingress filtering", but this cannot be determined using our techniques.

IPID technique. Load balancing can introduce a challenge in identifying whether a given network enforces ingress filtering. As a result of load balancing our packets will be split between multiple instances of the server, hence resulting in low IPID counter values. There are different approaches for distributing the load to different instances, e.g., random or round robin, which makes it impossible to identify whether a "load-balanced-server" is on a network which applies ingress filtering or not.

Anycasted server instances can also introduce a challenge in inferring ingress filtering enforcement. We identified such cases by performing traceroutes to the server.

DNS technique. Firewalls, blocking incoming packets on port 53, would as a result generate a similar effect as ingress filtering on our servers: we would not receive any DNS requests to our domain. However, such a setting does not indicate that the tested network actually performs ingress filtering.

PMTUD technique. Firewalls are often configured to block ICMP packets. In such case the evaluation result is similar as when a tested network does not enforce ingress filtering: our PMTUD packets will be blocked by the firewall, but not because they originate from an IP address that belongs to the tested network but because the firewall blocks ICMP packets. This case can be identified by sending ICMP PMTUD packets from an IP address that does not belong to the network. If the ICMP packets are not blocked (but were blocked when the packets were sent from a spoofed IP address) then the network does not block ICMP packets and does enforce IP spoofing filtering. On the other hand if the packets are blocked then one cannot determine if the blocking is done because of ICMP or because of filtering of spoofed IP addresses.

4.5 Comparison with Other Measurements

To understand the effectiveness of our methodologies we compare the results of our measurements with the active measurements of ingress filtering performed by the CAIDA Spoofer Project. These include two types of measurements: *using traceroute* and *using agents*. The spoofer project is the only measurement study that makes the datasets from their scans available online. The traceroute approach and the agents approach are the only two other active measurements of enforcement of ingress filtering (see Related Work Section 2). We crawled all the 217,917 session reports in 2019 of CAIDA Spoofer Project. These included 2,867 ASes with Spoofer Project agents, and 2,500 ASes with Spoofer Project traceroute loops (total of 5,367 ASes). Using our methodologies we measured 63,522 ASes, which is substantially more than the previous studies all together. We compare between our results and the other two methodologies below.

Traceroute Active Measurements. We analyse the datasets from the traceroute measurements performed by the CAIDA Spoofer Project within the last year 2019, [29]. The measurements identified 2,500 unique loops, of these 703 were provider ASes, and 1,780 customer ASes. The dataset found 688 ASes that do not enforce ingress filtering. Out of 688 ASes found with traceroutes by the Spoofer Project, we could not test 4 ASes (none of our tests applied) and 36 ASes were not included in our tests (those ASes could not be located from domain names - due to our attempt to reduce traffic and not to scan IPv4 but to collect the services via domain names). The rest of the ASes agree with our measurement results.

Agents Active Measurements. Agents with active probes found 608 ASes that were found not to be enforcing ingress filtering using the agents approach of the Spoofer Project (these include duplicates with the traceroute loops measurements). Those contain some of the duplicates from traceroute measurements: together both approaches of the Spoofer Project found 1,113 ASes to be spoofable. Apart from 57 ASes not included in our tests, we could not test 9 ASes, the rest were also identified by our tests.

Although the agents provide the optimal setup for testing filtering, with control over the packets that can be crafted and sent from both sides, as we explain in Related Work Section 2, this approach is limited only to networks that deploy agents on their networks. In contrast, SMap provides better coverage since it is potentially applicable to every network that has one of the services that are required in our tests.

In total, our results identified 51,046 ASes to be spoofable, which is more than 80% of the ASes that we tested. This is also 50,023 ASes more than that both the traceroute and the agents approaches found.

These findings show that SMap offers benefits over the existing methods, providing better coverage of the ASes in the Internet and not requiring agents or conditions for obtaining traceroute loops, hence improving visibility of networks not enforcing ingress filtering.

5 NETWORKS ANALYSIS

In order to understand if there are differences in enforcement of ingress filtering between different network types and different countries, we perform characterisation of the networks that we found to not be filtering spoofed packets. Specifically, we ask the following questions: *Does business type of networks or geo-location of networks influence filtering of spoofed packets?*

To derive the geo-location of ASes we used MaxMind GeoLite2 GeoIP database [1]. The results are listed in Table 5. The tested ASes are distributed across different countries, with most ASes being in large countries, like US and Russia. The ration of spoofable ASes ranges between 67% and 84%, with Ukraine leading with the fraction of spoofable networks, with 84%. Surprisingly the ratio between the geolocation and spoofed packets is similar across different countries, with USA and Russia leading with 32% of the networks and 33% of the networks respectively, that do not filter spoofed packets.

We also want to understand the types of networks that we could test via domains-wide scans. To derive the business types we use the PeeringDB. We classify the ASes according to the following business types: content, enterprise, Network Service Provider (NSP), Cable/DSL/ISP, non-profit, educational/research, route server at Internet Exchange Point (IXP)[1] We plot the networks that do not enforce ingress filtering according to business types in Figure 12.

[1]A route server directs traffic among Border Gateway Protocol (BGP) routers.

Country	Tested ASes	Spoofable ASes	Spoofable Ratio
US	16,138	12,385	76.74%
BR	7,692	6,447	83.81%
RU	4,906	4,221	86.04%
PL	2,092	1,739	83.13%
DE	2,171	1,677	77.25%
GB	2,231	1,648	73.87%
UA	1,776	1,547	87.11%
IN	1,970	1,480	75.13%
ID	1,412	1,236	87.54%
AU	1,625	1,234	75.94%
CA	1,484	1,184	79.78%
FR	1,310	1,036	79.08%
NL	1,308	1,026	78.44%
IT	1,013	850	83.91%
ES	1,001	783	78.22%
AR	918	733	79.85%
RO	962	720	74.84%
JP	782	606	77.49%
HK	743	565	76.04%
CZ	673	560	83.21%

Table 5: Top-20 Countries with most tested ASes.

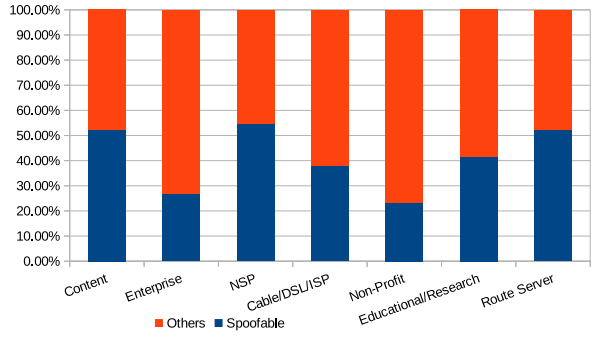

Figure 12: Spoofable ratio across ASes' types. AS type is queried from PeeringDB.

According to our study enterprise and non-profit networks enforce ingress filtering more than other networks. In contrast, NSPs contain the most networks that do not enforce ingress filtering.

There is a strong correlation between the AS size and the enforcement of spoofing, see Figure 13. Essentially, the larger the AS, the higher the probability that our tools identify that it does not filter spoofed packets. The reason can be directly related to our methodologies and the design of our study: the larger the network the more services it hosts. This means that we have more possibilities to test if spoofing is possible: for instance, we can identify a higher fraction of servers with a globally incremental IPID counters, which are not "load balanced". In Figure 14 we plot the statistics of the tested networks according to their size and type. The results show a correlation between the size of the network and its type.

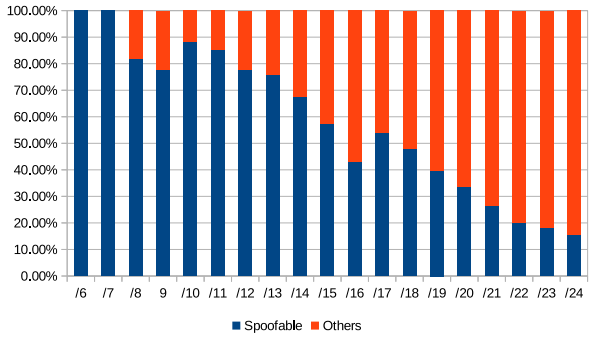

Figure 13: Spoofable ratio according to networks' sizes. Network size is calculated from GeoLite2-ASN database.

For instance, most NSP networks are large, with CIDR/6. This is aligned with our finding that among NSP networks there was the highest number of spoofable networks.

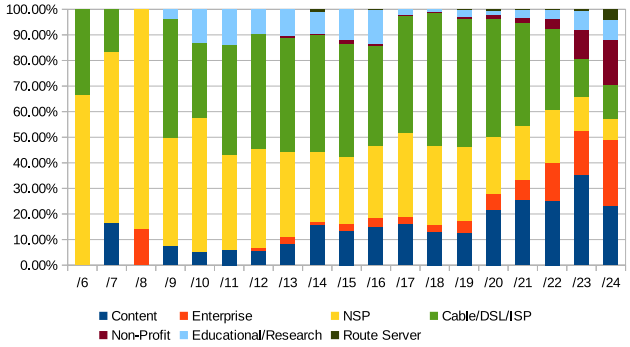

Figure 14: Distribution of networks' sizes vs types.

6 CONCLUSIONS

Much effort is invested to understand the extent of spoofability in the Internet. However, current measurement studies have limited applicability, providing results that apply to a small set of Internet networks.

Our work provides the first comprehensive view of ingress filtering in the Internet. We showed how to improve the coverage of the Internet in ingress filtering measurements to include many more ASes that were previously not studied. Our techniques allow to cover more than 90% of the Internet ASes, in contrast to best coverage so far of 7.5% of the ASes performed by the Spoofer Project. This coverage can be further extended to include 100% of the Internet's ASes by scanning the IPv4 range instead of opting for the dataset of [2], that we used in our study.

The most significant aspect of our methodologies is that they do not require coordination with the scanned networks. SMap can measure spoofability in any TCP/IP network with standard and widely supported services, such as Email and web. We integrated into SMap three techniques for testing ingress filtering: DNS-based,

IPID-based and PMTUD-based. Our experimental comparison of the effectiveness of the techniques demonstrated that DNS-based technique has a wider applicability rate on networks that operate DNS resolvers than the other two techniques, while the detection of the spoofability of networks is more accurate with PMTUD.

We set up SMap as a public service for continuous collection and analysis of the ingress filtering in the Internet at `https://smap.cad.sit.fraunhofer.de`.

ACKNOWLEDGMENTS

This work has been co-funded by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) SFB 1119.

REFERENCES

[1] [n. d.]. MaxMind GeoLite2 Database. https://dev.maxmind.com/geoip/geoip2/geolite2/
[2] [n. d.]. Rapid7 Labs Open Data. https://opendata.rapid7.com/
[3] F. Baker and P. Savola. 2004. Ingress Filtering for Multihomed Networks. http://tools.ietf.org/rfc/rfc3704.txt RFC3704.
[4] Paul Barford, Rob Nowak, Rebecca Willett, and Vinod Yegneswaran. 2006. Toward a model for source addresses of internet background radiation. In *Proc. of the Passive and Active Measurement Conference*.
[5] Robert Beverly and Steven Bauer. 2005. The Spoofer project: Inferring the extent of source address filtering on the Internet. In *Usenix Sruti*, Vol. 5. 53–59.
[6] Robert Beverly, Arthur Berger, Young Hyun, and K Claffy. 2009. Understanding the efficacy of deployed internet source address validation filtering. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*. 356–369.
[7] Robert Beverly, Ryan Koga, and KC Claffy. 2013. Initial longitudinal analysis of IP source spoofing capability on the Internet. *Internet Society* (2013), 313.
[8] Markus Brandt, Tianxiang Dai, Amit Klein, Haya Shulman, and Michael Waidner. 2018. Domain validation++ for MitM-resilient PKI. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2060–2076.
[9] Zesheng Chen, Chuanyi Ji, and Paul Barford. 2008. Spatial-temporal characteristics of internet malicious sources. In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE, 2306–2314.
[10] Jakub Czyz, Michael Kallitsis, Manaf Gharaibeh, Christos Papadopoulos, Michael Bailey, and Manish Karir. 2014. Taming the 800 pound gorilla: The rise and decline of NTP DDoS attacks. In *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, 435–448.
[11] Tianxiang Dai, Philipp Jeitner, Haya Shulman, and Michael Waidner. 2021. The Hijackers Guide To The Galaxy: Off-Path Taking Over Internet Resources. In *30th USENIX Security Symposium (USENIX Security 21)*. 3147–3164.
[12] Tianxiang Dai, Haya Shulman, and Michael Waidner. 2021. DNS-over-TCP considered vulnerable. In *ANRW '21: Applied Networking Research Workshop, Virtual Event, USA, July 24-30, 2021*. ACM, 76–81.
[13] Tianxiang Dai, Haya Shulman, and Michael Waidner. 2021. Let's Downgrade Let's Encrypt. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. ACM.
[14] Alberto Dainotti, Karyn Benson, Alistair King, KC Claffy, Michael Kallitsis, Eduard Glatz, and Xenofontas Dimitropoulos. 2013. Estimating internet address space usage through passive measurements. *ACM SIGCOMM Computer Communication Review* 44, 1 (2013), 42–49.
[15] Zakir Durumeric, Frank Li, James Kasten, Johanna Amann, Jethro Beekman, Mathias Payer, Nicolas Weaver, David Adrian, Vern Paxson, Michael Bailey, et al. 2014. The matter of heartbleed. In *Proceedings of the 2014 conference on internet measurement conference*. 475–488.
[16] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. 2013. ZMap: Fast Internet-wide scanning and its security applications. In *Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13)*. 605–620.
[17] Roya Ensafi, Jeffrey Knockel, Geoffrey Alexander, and Jedidiah R Crandall. 2014. Detecting intentional packet drops on the Internet via TCP/IP side channels. In *International Conference on Passive and Active Network Measurement*. Springer, 109–118.
[18] Paul Ferguson. 2000. Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. (2000).
[19] P. Ferguson and D. Senie. 2000. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. http://tools.ietf.org/rfc/rfc2827.txt RFC2827.
[20] Matthias Göhring, Haya Shulman, and Michael Waidner. 2018. Path MTU Discovery Considered Harmful. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 866–874.
[21] Amir Herzberg and Haya Shulman. 2013. Fragmentation Considered Poisonous: or one-domain-to-rule-them-all.org. In *IEEE CNS 2013. The Conference on Communications and Network Security, Washington, D.C., U.S.* IEEE.
[22] Gokay Huz, Steven Bauer, KC Claffy, and Robert Beverly. 2015. Experience in using mturk for network measurement. In *Proceedings of the 2015 ACM SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big (Internet) Data*. 27–32.
[23] Christopher A Kent and Jeffrey C Mogul. 1987. *Fragmentation considered harmful*. Vol. 17.
[24] Maciej Korczyński, Yevheniya Nosyk, Qasim Lone, Marcin Skwarek, Baptiste Jonglez, and Andrzej Duda. 2020. The Closed Resolver Project: Measuring the Deployment of Source Address Validation of Inbound Traffic. *arXiv preprint arXiv:2006.05277* (2020).
[25] 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.
[26] Marc Kührer, Thomas Hupperich, Christian Rossow, and Thorsten Holz. 2014. Exit from Hell? Reducing the Impact of Amplification DDoS Attacks. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*. 111–125.
[27] 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 2017 Internet Measurement Conference*. ACM, 86–99.
[28] Qasim Lone, Matthew Luckie, Maciej Korczyński, Hadi Asghari, Mobin Javed, and Michel van Eeten. 2018. Using Crowdsourcing Marketplaces for Network Measurements: The Case of Spoofer. In *2018 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 1–8.
[29] Qasim Lone, Matthew Luckie, Maciej Korczyński, and Michel van Eeten. 2017. Using loops observed in traceroute to infer the ability to spoof. In *International Conference on Passive and Active Network Measurement*. Springer, 229–241.
[30] Matthew Luckie, Robert Beverly, Ryan Koga, Ken Keys, Joshua A Kroll, and k claffy. 2019. Network Hygiene, Incentives, and Regulation: Deployment of Source Address Validation in the Internet. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 465–480.
[31] Gordon Fyodor Lyon. 2009. *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure.
[32] Keyu Man, Zhiyun Qian, Zhongjie Wang, Xiaofeng Zheng, Youjun Huang, and Haixin Duan. 2020. DNS Cache Poisoning Attack Reloaded: Revolutions with Side Channels. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, CCS*. ACM.
[33] Pietro Marchetta, Antonio Montieri, Valerio Persico, Antonio Pescapé, Ítalo Cunha, and Ethan Katz-Bassett. 2016. How and how much traceroute confuses our understanding of network paths. In *2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*. IEEE, 1–7.
[34] Jared Mauch. 2013. Open resolver project. In *Presentation, DNS-OARC Spring 2013 Workshop (Dublin)*.
[35] Rui Miao, Rahul Potharaju, Minlan Yu, and Navendu Jain. 2015. The dark menace: Characterizing network-based attacks in the cloud. In *Proceedings of the 2015 Internet Measurement Conference*. ACM, 169–182.
[36] David Moore, Colleen Shannon, Douglas J Brown, Geoffrey M Voelker, and Stefan Savage. 2006. Inferring internet denial-of-service activity. *ACM Transactions on Computer Systems (TOCS)* 24, 2 (2006), 115–139.
[37] Paul Pearce, Roya Ensafi, Frank Li, Nick Feamster, and Vern Paxson. 2017. Augur: Internet-wide detection of connectivity disruptions. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 427–443.
[38] Terrance A. Roebuck. 2005. Network security: DoS vs DDoS attacks. http://www.crime-research.org/articles/network-security-dos-ddos-attacks/5.
[39] Christian Rossow. 2014. Amplification Hell: Revisiting Network Protocols for DDoS Abuse.. In *NDSS*.
[40] Haya Shulman and Michael Waidner. 2014. Fragmentation considered leaking: port inference for dns poisoning. In *International Conference on Applied Cryptography and Network Security*. Springer, 531–548.
[41] Stephen M Specht and Ruby B Lee. 2004. Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures.. In *ISCA PDCS*. 543–550.
[42] J. Touch. 2013. Updated Specification of the IPv4 ID Field. http://tools.ietf.org/rfc/rfc6864.txt RFC6864.
[43] Duane Wessels, Marina Fomenkov, et al. 2003. Wow, that'sa lot of packets. In *Proceedings of Passive and Active Measurement Workshop (PAM)*.
[44] Guang Yao, Jun Bi, and Athanasios V Vasilakos. 2014. Passive IP traceback: Disclosing the locations of IP spoofers from path backscatter. *IEEE Transactions on Information Forensics and Security* 10, 3 (2014), 471–484.