Beyond the Horizon: Uncovering Hosts and Services Behind Misconfigured Firewalls

Qing Deng
University of California, Riverside
qdeng010@ucr.edu

Juefei Pu
University of California, Riverside
jpu007@ucr.edu

Zhaowei Tan
University of California, Riverside
ztan@ucr.edu

Zhiyun Qian
University of California, Riverside
zhiyunq@cs.ucr.edu

Srikanth V. Krishnamurthy
University of California, Riverside
krish@cs.ucr.edu

***Abstract*—Public IP addresses can expose devices and services to risks such as port scanning and subsequent cyberattacks. Therefore, firewalls are extensively deployed and play a critical role in enforcing security policies and preventing unauthorized access. However, vulnerabilities can allow firewalls to be bypassed, effectively nullifying the protection.**

In this paper, we present the first comprehensive study of a previously understudied attack surface: firewall misconfigurations that inadvertently expose protected services to the public Internet. Specifically, we demonstrate flawed firewall rules that allow inbound connections from special source ports to bypass the firewall, and explore the prevalence and security implications thereof. To this end, we scan the IPv4 space for 15 commonly high-risk TCP and UDP services from two special source ports. Our measurement reveals the widespread existence of such misconfigurations and identified over 2,000,000 otherwise unreachable services spread over 15,837 autonomous systems, expanding the "observable Internet" for various protocols by up to 12.60%. More importantly, the affected services generally exhibit higher security risks than the publicly accessible ones, like outdated software versions and weak configurations. Despite the severity of this vulnerability, our honeypot experiment provides little evidence of active exploitation in the wild. Our findings offer insights for better security posture and network administration, helping researchers and organizations anticipate and mitigate potential cyber threats emanating from the Internet.

1. Introduction

The public Internet is constantly being scanned [1], [2], [3], [4], [5], [6], often by malicious actors as a preliminary step in cyberattacks [1], [7]. Devices with public IP addresses are subject to such probes, and attackers actively try to compromise reachable hosts by exploiting vulnerabilities like software defects, configuration flaws, and weak passwords [8], [9]. Unsecured hosts may be compromised within minutes [10], [11]. As a result, firewalls are widely deployed to protect devices and their services by blocking unwanted access. Typically, these firewalls block all inbound connections[1] (with a few exceptions if the end device is a server). They may also restrict outbound traffic to a few essential protocols, e.g., HTTP and DNS. Such a firewall is supposed to minimize the attack surfaces of the end devices.

However, this belief is not always true, as misconfigured firewalls can fail silently in response to simple yet carefully crafted network traffic, exposing protected hosts and services to external attackers. For example, when configuring a firewall to allow accessing HTTP websites, the administrator may inadvertently allow any inbound traffic from TCP port 80, creating a loophole that permits any inbound TCP connection from port 80. Similarly, allowing DNS traffic might unintentionally open up the firewall to any inbound UDP datagrams from source port 53. Similar misconfigurations have previously occurred in Windows 2000/XP/2003 [12] and Mac OS X Tiger [13], which allowed connections from certain ports to bypass their built-in firewalls. Although this vulnerability is documented in the Nmap manual [14], it has been long forgotten and overlooked, and no comprehensive study has been conducted. The prevalence, security implications, and real-world exploitation of such misconfigurations in today's Internet remain unknown.

Our Study. We conduct the first comprehensive study of firewall misconfigurations of this type, which mistakenly allow undesired connections to bypass the firewalls. We aim to quantify the prevalence and security implications of this understudied attack surface. To this end, we identify the affected services in the IPv4 address space and analyze their security risks.

Specifically, we scan the IPv4 space targeting 15 commonly high-risk services (e.g., SSH, HTTP, and MongoDB) and identify those that are only accessible from specific unusual source ports. The result confirms the widespread existence of firewall misconfigurations of this kind. By initiating connections from TCP port 80 and UDP port 53, we uncover *over two million* services that would otherwise remain unreachable, including *over 800 thousand* management services (SSH, Telnet, RDP, SNMP, and IPMI). The

1. For the sake of convenience, we refer to the sessions of connectionless protocols like UDP also as *connections*.

affected hosts are distributed across 15,837 autonomous systems (ASes) and 221 countries and regions. With this firewall circumvention technique, we are able to reach up to 12.60% more services, with the ratio varying by protocol, expanding the observable Internet to a great extent.

Notably, we find that the affected generally exhibit higher security risks than public services, such as outdated software versions and weak ciphers. Furthermore, we identified *hundreds* of unprotected SMB shares and MongoDB databases. This indicates that these misconfigured firewalls provide a false sense of security. We also observe *over ten thousand* vulnerable home routers that may be compromised from the public Internet and one public cloud that provisions flawed default firewall rules to virtual machines.

To investigate whether such misconfigurations are actively exploited in the wild, we carry out a honeypot experiment over four months. While our study shows no evidence of large-scale exploitation of this attack surface in the wild, its widespread presence should trigger an alarm for network administrators and security researchers. We hope that the insights in this paper can help improve security posture and benefit various sensitive applications.

Disclosure. We perform responsible disclosure to the 15,837 ASes with affected hosts and set up a website to explain this vulnerability and provide fix suggestions. In total, the website has been viewed by recipients from 1,436 ASes, and we have received inquiries, updates, and letters of thanks from 173 ASes. In the follow-up measurements, we see a noticeable decrease in affected hosts in the most affected ASes. We also disclose this issue to involved parties whose products are found to contain misconfigurations in their host-level firewalls.

Contributions. We make the following contributions:

- We perform the first comprehensive study of an understudied attack surface, firewall misconfigurations that inadvertently allow undesired connections, at the Internet scale.
- We confirm the widespread existence of this issue and identify over two million affected services in the IPv4 space, with a low false positive rate ensured by our multi-pass workflow.
- We analyze the security implications of the affected services and find that they generally exhibit higher security risks than public services.
- While we did not observe any large-scale exploitation in the wild in our honeypot experiment, we conduct responsible disclosure to the affected parties out of an abundance of caution and receive positive feedback.

2. Background and Related Work

2.1. Internet Scanning

Internet-wide scanning is crucial for various cybersecurity applications, like vulnerability assessment and threat tracking [4], [15], [16], [17]. It also helps understand the topology and connectivity across different regions [18], [19]. By probing the accessible hosts and services, it maps the Internet and collects large-scale empirical data for analysis.

Due to the importance of Internet-scale measurements, numerous techniques have been developed to facilitate Internet scanning. ZMap [20] democratized fast Internet-wide surveys. Masscan [21] is capable of scanning the IPv4 space in minutes with 10-gigabit Internet connections. While the immense space of IPv6 precludes full scans, there have been works utilizing heuristics and/or machine learning [22], [23], [24] to pinpoint scannable subspaces.

There are also platforms that spare researchers the burden of managing their own scanners. Search engines like Shodan [25], Censys [26], and FOFA [27] periodically probe the Internet, and users can search for hosts and services of interest. RIPE Atlas [28] is a global network measurement platform where users can distribute measurement tasks to tens of thousands of crowdsourced probes around the world.

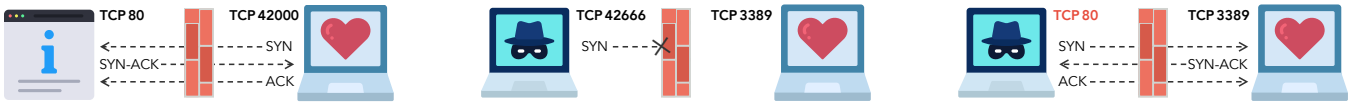
Unfortunately, besides research purposes, Internet scanning is largely abused by attackers for malicious activities. In general, attackers scan ports where vulnerable services reside [7]. After identifying the hosts running the target services, attackers try to exploit them by different means. They may try to compromise the hosts by exploiting software vulnerabilities or brute-forcing [8], [9] and use these machines to mine cryptocurrencies or expand botnets [29], [30]. Attackers may also exploit configuration flaws to launch attacks, e.g., DNS and NTP reflection amplification attacks [31], [32], without fully compromising the hosts.

2.2. The Observable Internet

The more services we can reach, the more empirical data we can collect and analyze. It is also true for attackers – they need to reach a service before they can exploit it. In this paper, we name the set of reachable Internet services *the observable Internet*. Only the services in the observable Internet can be interacted with by actors on the Internet, including performing measurements or launching attacks.

However, despite the huge number of connected devices, hosts and services are sparse even for the relatively small IPv4 space. Bano et al. [33] pointed out that only $\sim$10% of IPv4 addresses respond to ICMP pings. Klick et al. [34] found that only two-thirds of IPv4 addresses are announced and that they can collect 90-99% of the desired responses by scanning at most 75% of the announced IP addresses. The majority of online devices and their services are concealed behind firewalls or NAT gateways.

Researchers have been working on expanding the observable Internet in different ways. Izhikevich et al. [35] investigated services on non-standard ports and found that only 3% of HTTP and 6% of TLS services ran on ports 80 and 443, respectively. They further developed a framework to predict the ports of these misplaced services [36]. Song et al. [37] also proposed a machine learning method to improve the hit rate and intrusiveness of uncovering such services. Wan et al. [38] demonstrated the importance of scanning from different geographic and topological locations.



(a) Normal traffic is allowed. (b) Attacks from high ports are blocked. (c) Attacks from port 80 are overlooked.

Figure 1: Example of Bypassing Misconfigured Firewalls

There are also techniques leveraging vulnerabilities to reach normally unreachable hosts and services. Rytilahti et al. [39] found that application-layer middlebox protocols might be used to bypass NAT gateways and reach internal hosts. Feng et al. [40] utilized the shared-IPID side channel to penetrate NAT devices. The SYN cookie implementation in old Linux versions allowed attackers to bypass firewall rules by guessing cookies [41]. Some firewalls dynamically open ports for multichannel protocols like FTP and SIP, and this feature can be exploited for tunneling any ports [42], [43]. Two vintage firewall flaws in Windows [12] and Mac OS X [13] allowed any inbound traffic from certain remote ports, effectively nullifying the system firewalls.

In conclusion, expanding the observable Internet is critical to network research and can help anticipate and mitigate potential cyber threats. Our work contributes a new perspective by demonstrating the prevalence and significant impacts of firewall misconfigurations.

2.3. Firewall

Firewalls inspect inbound and outbound traffic based on predefined rules, establishing a barrier between the trusted network and the untrusted network (usually the Internet). Firewalls come in different forms and implementations. In the context of this paper, the key lies in whether the firewall is able to, or configured to, track network flows, such as TCP connections and UDP sessions. Based on whether they have this capability, we can categorize firewalls into *stateless firewalls* and *stateful firewalls* [44].

Stateless Firewalls. Stateless firewalls inspect network traffic on a per-packet basis, unaware of network flows. Stateless firewalls only support *stateless rules*, which permit or discard packets with specified properties, such as specific source/destination addresses and ports. A stateless firewall matches each packet against the rules and makes the decision solely based on that packet. Consequently, stateless firewalls are generally incapable of distinguishing between inbound and outbound connections. While many advanced models can filter TCP segments by their flags [45], enabling them to block inbound TCP connections, it does not extend to UDP and is absent in the access control lists (ACLs) of some switches [46]. Stateless firewalls are widely deployed in environments with heavy network traffic, as they offer superior performance and desirable simplicity.

Stateful Firewalls. Stateful firewalls are capable of tracking network flows and filtering the packets based on their corresponding flows. Stateful firewalls support *stateful rules*, which permit or forbid network flows with specific properties. The permitted flows are recorded, and orphan packets that neither belong to any known flow nor create a new flow are discarded. The firewall deletes a flow record when termination signals like FIN and RST segments are found. For connectionless protocols like UDP and ICMP, the record expires after a period of inactivity [47]. Most stateful firewalls, like iptables, also support stateless rules to enhance performance and simplify configuration. With only stateless rules, a stateful firewall operates in the same way as stateless firewalls.

Firewall Misconfiguration. As critical guardians of networks, firewalls mainly rely on manual configurations and a misconfigured one can be a single point of failure. Hence, firewall misconfiguration has been studied from various perspectives, such as automatically modeling and examining firewall rules [48], [49], [50], [51]. Bringhenti et al. [52] proposed an approach to automate firewall configuration. The misconfiguration which unintentionally allows inbound connections from specific source ports has been mentioned in the Nmap manual [53] and some articles available online [13], [54]. However, there has been no comprehensive study on this topic and the security implications thereof in today's Internet remain unknown. Our work demonstrates that such misconfigurations are still widespread and have alarming prevalence and far-reaching impacts.

3. A Firewall's Achilles Heel

A single flawed rule can compromise the effectiveness of the entire firewall. In this section, we demonstrate the mechanism, threat model, and behavior patterns regarding such flawed rules.

3.1. Mechanism

Configuring firewalls may seem intuitive. However, subtle intricacies can render it prone to errors.

Using *stateless rules*, it takes two rules to permit access to external HTTP services: one for user packets to go out and another for server packets to come back. In a simplified scenario where only access to HTTP services is allowed, a network administrator may set up the following rules:

1) Permit TCP segments from internal hosts to port 80 of external hosts.
2) Permit TCP segments from port 80 of external hosts to internal hosts.
3) Discard all other IP packets.

These rules usually work well as operating systems typically use random high ports as source ports by default [55], and connections to internal hosts are supposed to be rejected by

Rule 3, as shown in Figure 1b. However, suppose an attacker initiates TCP connections to the protected hosts from port 80. In that case, the initial SYN segment will be permitted by Rule 2, as the firewall is unable to distinguish the connection direction. Subsequently, the malicious traffic consistently adheres to the first two rules, evading the intention of blocking inbound connections, as illustrated in Figure 1c. Similar misconfigurations can happen when allowing UDP protocols like DNS and NTP, causing a loophole that allows any inbound UDP datagrams from source ports 53 or 123.

It is unlikely for *stateful firewalls with stateful rules* to undergo the same misconfigurations, as they operate on network flows and do not require a complementary rule to permit returning packets. Still, human errors may lead to flawed rules being added. For example, when allowing inbound connections to TCP port 80 of a web server, the network administrator might misspell the destination port as the source port, allowing any inbound connection *from* TCP port 80. In such cases, the examples in Figure 1 still hold.

3.2. Threat Model

We assume an attacker who actively scans the Internet to find and exploit vulnerable services. The attacker is remote and does not have special network capabilities.

Leveraging flawed firewall rules, the attacker can bypass the misconfigured firewalls and reach protected services by simply manipulating the source port of their connections. The attacker can specify a common port like TCP 80 or UDP 53 as the source port for all their scanning activities. The services reached in this manner should be a superset of the result of regular scanning, as scanning from a designated port should not obscure the services reachable from random ports. The attacker can also scan from multiple special ports to circumvent even more misconfigured firewalls.

Consequently, the attacker can expand their observable Internet and obtain more potential targets for their attacks, posing a greater threat to the Internet overall.

3.3. Behavior Pattern

While it is infeasible to inspect the rules of firewalls on the Internet directly, we can distinguish an affected service by initiating multiple connections from different ports and observing the responses.

Assume that the affected service is on port P. Normally, the firewall blocks all inbound connections; however, it erroneously allows inbound traffic from port X. When initiating connections from a random high port R and from port X, we should notice the following differences in responses:

- Port P is irresponsive or responds with errors, e.g., RST segments or ICMP unreachable messages when the connection is initiated from source port R.
- Port P becomes responsive, establishing the connection and responding at the application layer, when the connection is initiated from source port X.

It is worth noting that theoretically, port X may be a high port itself, as the flawed rule may allow any port depending on the exact configuration. There may also be multiple ports from which the inbound traffic can bypass the firewall, as there may be multiple flawed rules.

4. Measurement Framework

While examining a single service is straightforward, it becomes challenging when scaling the scope to the whole IPv4 space. In this section, we discuss how we address the challenges relating to the Internet-scale measurement.

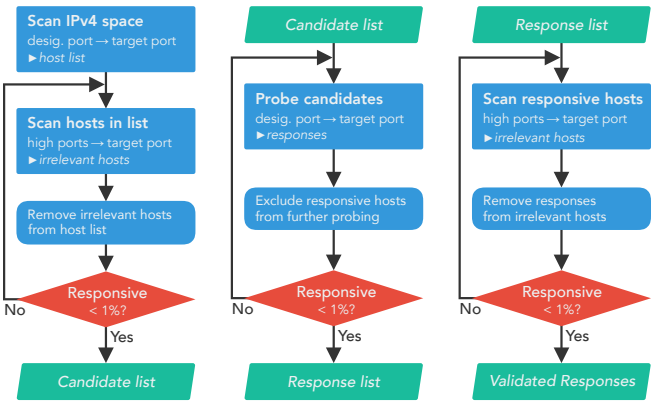
Goals. We aim to study (i) the prevalence and (ii) the security implications of the aforementioned firewall misconfigurations at the Internet scale. To this end, we need to:

1) Identify the affected services in the IPv4 space.
2) Extract the characteristics of the affected services, e.g., software versions and supported cipher suites.
3) Analyze the security implications of exposure.

Challenges. Networks are dynamic and volatile, and the Internet exemplifies these characteristics. Packet loss and region-based filtering rules are commonplace. A host can run any service, and the flawed rule may allow any source port to bypass the firewall inadvertently. To address these complexities, we need a robust workflow complemented by diverse vantage points, carefully selected scope, and proper tools to ensure both accuracy and efficiency.

4.1. Workflow

The core of the workflow is an iterative method to ensure both broad coverage and a low false positive rate (below 1%). We denote the chosen source port as the *designated port* and the port of the target service as the *target port*.



(a) Identification (b) Probe (c) Validation

Figure 2: Three-Phase Workflow

Phase 1: Identify the Affected Hosts. In this phase, we unearth the hosts whose target ports are responsive *only when* we connect from the designated port. The steps are:

1) Scan the IPv4 space targeting the target port from the designated port. The responsive hosts constitute

the initial host list, including both affected hosts and *irrelevant hosts*.
2) Scan the host list targeting the target port from random high ports. The responsive hosts are the irrelevant hosts because they are reachable from high ports[2].
3) Remove the irrelevant hosts from the current host list.
4) Repeat Steps 2 & 3 until the response rate is below 1%, ensuring a low false positive rate. The remaining hosts form the candidate list.

The process is shown in Figure 2a. We adopt the iterative design to circumvent packet loss and other dynamic behaviors. Step 1 incurs most of the cost, which requires a full scan of the IPv4 space. The following steps converge rapidly, typically after three to five rounds of scanning a small number of addresses. We believe that the candidate list is a reliable set of affected hosts.

We note that an open port does not necessarily serve the target service, and hosts may go offline at Step 2, causing false positives. We handle these cases in the next phase.

Phase 2: Probe the Affected Services. In this phase, we send application-layer probes and collect the responses for characteristic extraction. The steps are:

1) Send probes (e.g., HTTP or DNS requests) to the target port of the candidate hosts from the designated port.
2) Record the responses and exclude the responding hosts from further probing.
3) Repeat Steps 1 & 2 until the response rate is below 1%, ensuring high coverage. The aggregated responses constitute the response list.

The process is shown in Figure 2b. The iteration usually finishes in three rounds. The responses come from a subset of the candidates which truly serve the target service and have not gone offline during the measurement.

We recognize that, despite ensuring a low false positive rate in Phase 1, a significant number of affected hosts may either fail to respond to our application-layer probes or respond with an unexpected protocol due to various factors. Consequently, the subset of the hosts responding with valid responses in Phase 2 may be substantially smaller than the candidate list generated in Phase 1, necessitating an additional verification of the false positive rate. We handle this in the next phase.

Phase 3: Validate the False Positive Rate. In this phase, we confirm the desired false positive rate by scanning the affected services from high ports again. The steps are:

1) Scan the services in the response list from random high ports. The responsive hosts are new irrelevant hosts as their services turn out to be reachable from high ports.
2) Remove the responses previously collected from the irrelevant hosts.
3) Repeat Steps 1 & 2 until the response rate is below 1%, ensuring a desired false positive rate. The remaining responses are the validated responses.

2. Here we ignore the negligible possibility that the random high port is the very port that can bypass the firewall.

The process is shown in Figure 2c. The iteration often finishes in three to five rounds. The validated responses have the desired false positive rate below 1%.

After these phases, we parse the validated responses to extract the characteristics of the services, such as software versions and supported cryptographic algorithms. We further analyze the characteristics to conclude security risks.

4.2. Scope Selection

While the IPv4 address space has a relatively manageable size, there are numerous (i) possible services and (ii) unexpectedly allowed source ports. Given the scale, we have to restrict our scope to a small number of target services and designated ports due to feasibility considerations.

Target Service. To illustrate the prevalence and security implications of firewall misconfigurations of this kind, we choose 15 common services that are often vulnerable. We list the services in Table 1. For services supporting both TCP and UDP, we choose the more commonly used protocol, like UDP for DNS. We divide the services into four categories. Management services are for device management; file sharing and database services store user and application data; general services include other often exploitable services. Compromise of these services may lead to device takeover, information leakage, etc. These services have various attack surfaces, such as weak passwords in SSH, Telnet, RDP, etc., and lack of authentication in FTP, SMB, MongoDB, etc. We introduce the attack surfaces of the services in detail in the corresponding sections in §6.

Category	Service	Port
Management	SSH	TCP 22
	Telnet	TCP 23
	RDP	TCP 3389
	IPMI	UDP 623
	SNMPv1	
	SNMPv2c	UDP 161
	SNMPv3	
File sharing	FTP	TCP 21
	SMB	TCP 445
Database	MySQL	TCP 3306
	MongoDB	TCP 27017
General	HTTP	TCP 80
	HTTPS	TCP 443
	DNS	UDP 53
	NTP	UDP 123

TABLE 1: Target Services for Measurement

Designated Ports. Technically, a firewall flaw may mistakenly allow traffic from any source port, depending on the specific erroneous rule. However, it is impractical to enumerate and scan from all 65,535 ports, and we hope to maintain a balance between coverage and footprint as discussed in §11. We select port 80 for TCP-based services and port 53 for UDP-based services because their corresponding protocols (HTTP and DNS) are among the most popular services on the Internet. According to recent studies on Internet traffic [56], [57], the HTTP family of protocols (HTTP,

HTTPS, and QUIC) account for the largest share. HTTP is the fundamental member of this family. DNS accounts for the majority of UDP traffic besides QUIC and is necessary for domain name resolution. Therefore, we presume that TCP port 80 and UDP port 53 are the ports most likely to be allowed by the firewalls, and the results should establish a reliable prevalence. This presumption is also confirmed in small-scale preflight measurements with other popular ports (TCP port 22/443 and UDP port 123/443).

4.3. Experiment Setup

Vantage Points. We employ multiple geographical regions to increase the coverage and mitigate packet loss and region-based filtering. This is important because it has been shown that an individual vantage point may miss up to 18.2% of the target service [38]. Specifically, we deploy five vantage points in the United States, Germany, Singapore, India, and Brazil, respectively. Each vantage point executes every step of the workflow in §4.1, and we merge the results, e.g., responsive hosts, into a union set after each step. The vantage points run Debian 12 with kernel 6.1 and are capable of scanning the entire IPv4 space in about two hours.

Control Group from Public Services. We hypothesize that the affected services face higher security risks because administrators may be misled by the false sense of security provided by firewalls, leading to less stringent maintenance practices. To verify this, we set up control groups for the services for which we can safely measure specific weaknesses (e.g., software versions and weak cipher suites); we do not set up control groups for services that only have general weaknesses (e.g., weak passwords) that require aggressive detection due to ethical considerations. The control groups consist of public services (unprotected by firewalls), whose hosts are randomly picked from the irrelevant hosts in Step 2 of Phase 1 in §4.1. Their sizes are comparable to or larger than the sizes of the affected services. We probe these public services following Phase 2 of the workflow, except that we initiate connections from random high ports.[3] Finally, we parse the responses and compare the security risks of the affected services and public services.

Tools. We leverage two open-source tools for scanning the Internet and sending probes to the target services.

- ZMap [20], [58]: ZMap is a fast network scanner for Internet-wide surveys. It is capable of specifying the source port[4]. We use it to detect the status of TCP ports and send probes to UDP services.
- ZGrab 2.0 [59]: ZGrab is a fast application-layer network scanner. It supports interacting with various protocols, and we enhance it to allow specifying the source port and support additional protocols like RDP. We use it to send probes to TCP services.

3. Not all picked hosts respond, possibly due to factors like network volatility. Hence, the final control group sizes vary (and thus, may appear random).

4. When no source port is provided, ZMap uses random source ports from 32768 to 61000, the default ephemeral port range of Linux.

We also develop our own tools in Python to analyze the output of ZMap and ZGrab. We store the data in MongoDB and ElasticSearch for indexing and querying.

4.4. Threats to Validity

We put in our best effort to get reliable and accurate results in our Internet-wide measurement. We employ the iterative and multi-pass approach in §4.1 to increase coverage and constrain false positives. However, the results may still be affected by the following factors:

Network Volatility. Despite measuring from five vantage points iteratively, we cannot fully eliminate the impact of packet loss. An affected host may be offline for part or all of our measurement period, which is out of our control.

Firewall Strategies. Our probes may suffer from region-based filtering rules despite multiple geographical locations.

Limited Scope. Most services do not run on their default ports [35], and firewalls may be misconfigured to allow packets from ports other than the designated ports we select as discussed in §4.2.

Special Networks. Certain networks may route packets via multiple paths (e.g., ECMP routing), which may affect the observation of middleboxes [60], [61]. Port forwarding may also introduce a bias on the number of observed hosts.

It is hard to quantify the exact impact of these factors as their effects are intertwined. Nevertheless, we try our best to keep the observed false positive rate below 1% as discussed in §4.1, and give a reliable set of affected services.

5. Overview of Measurement Results

We performed our primary Internet-wide measurement in August 2024. In this section, we showcase the prevalence and distributions of the aforementioned firewall misconfigurations. We also highlight the security implications of the affected services.

Service	Port	Count	Public Services	%
SSH	TCP 22	234,984	25,307,484	0.93%
Telnet	TCP 23	50,820	2,504,330	2.03%
RDP	TCP 3389	7,931	3,504,675	0.23%
IPMI	UDP 623	4,242	59,565	7.12%
SNMPv1		42,894		
SNMPv2c	UDP 161	36,753	19,360,968	2.51%[5]
SNMPv3		465,033		
FTP	TCP 21	32,172	7,713,688	0.42%
SMB	TCP 445	19,419	1,174,742	1.65%
MySQL	TCP 3306	19,456	3,853,048	0.50%
MongoDB	TCP 27017	338	198,470	0.17%
HTTP	TCP 80	222,539	163,503,752	0.14%
HTTPS	TCP 443	193,630	158,488,284	0.12%
DNS	UDP 53	334,358	5,287,835	6.32%
NTP	UDP 123	824,389	6,545,301	12.60%

TABLE 2: Number of Affected Services

5. Calculated based on unique IP addresses as Shodan has no SNMP version filter.

Country	Count
Italy	303,446
USA	290,848
China	216,647
Austria	130,331
Hungary	123,294
Poland	108,134
Japan	104,464
Brazil	85,869
Syria	74,144
Israel	69,440

TABLE 3: Top Countries

ASN	Country	Type	Count
1267	Italy	ISP	231,316
8447	Austria	ISP	125,994
5483	Hungary	ISP	121,853
4837	China	ISP	79,811
29256	Syria	ISP	75,124
16509	USA	Cloud	75,050
12741	Poland	ISP	54,586
1680	Israel	ISP	47,127
4134	China	ISP	39,303
15802	UAE	ISP	34,514

TABLE 4: Top ASes

Prevalence. We found a total of 2,488,958 services on 2,147,229 unique IP addresses affected by the previously discussed firewall misconfigurations. We display the detailed numbers relating to each service in Table 2. In addition, we queried Shodan [25] for the numbers of the corresponding publicly accessible services[6]. We list the figures and the ratios of the affected services to the public services. It turns out that by initiating connections from only two designated ports, we can reach up to 12.60% more services.

According to the statistics, the ratios of affected services to public services are higher for UDP services than for TCP services, especially for the more popular protocols like DNS and NTP. The reason for this phenomenon may be due to the prevalence of insecure firewall rules for UDP services on the Internet, which we discuss in §10.

Distributions. We identified the locations and ASes of the affected hosts by their IP WHOIS information. The affected hosts are located in 221 countries and regions and belong to 15,837 different ASes. We also list the ten most affected countries in Table 3 and the ten most affected ASes in Table 4. The results suggest that ISPs are most prone to such firewall misconfigurations. The complete geographic distribution is shown in Appendix A.1.

Besides ISPs and clouds, we also find that many well-known enterprises have thousands of affected hosts in their ASes, such as Apple, Google, Starlink, Alibaba, and Yandex. This further demonstrates the prevalence of firewall misconfigurations of this type. We list these noteworthy ASes in Appendix A.2.

We also notice a significant long tail in the distribution of the affected hosts in the ASes. Among the 15,837 ASes, only 202 (1.3%) have more than 1,000 hosts, accounting for 83.07% of affected hosts; only 960 (6.1%) have more than 100 hosts, accounting for 94.27% of affected hosts.

Security implications. After we extract the characteristics as described in §4.1, such as software versions and supported cryptographic algorithms, we analyze the potential security risks. We find the affected services are susceptible to various kinds of attack vectors. We highlight the key security risks in Table 5. For example, we find that about 30% of the affected SSH services employ weak cryptographic algorithms and that 31 Telnet services present shells without any authentication. We will describe the details relating to each service in §6. Note that we did not perform any aggressive measurements (e.g., guessing weak passwords) due to ethical considerations.

6. We specified the port number and filters like `-hash:0` to exclude invalid results.

Service	Security Risks
SSH	70,739 (30.10%) with weak cryptographic algorithms; 2,695 known to be vulnerable to CVE-2024-6387 [62].
Telnet	31 directly present shells without any authentication.
RDP	2,324 (29.31%) run end-of-life Windows versions.
IPMI	1,264 (29.8%) can lead to full control of servers; 2,772 (65.35%) support insecure authentication.
SNMP	47,117 devices may leak configurations, including servers, routers, switches, printers, VoIP devices, etc.; 465,033 devices may be fingerprinted or brute-forced.
FTP	1,833 known to run outdated software; two known to allow anonymous login.
SMB	202 unprotected file shares; 642 (63.44% of Windows hosts) run end-of-life Windows versions; 10,890 Linux hosts are vulnerable routers.
MySQL	10,522 (54.08%) run end-of-life versions.
MongoDB	790 unprotected databases with sizes up to 1 terabyte.
HTTP(S)	20,681 (53.54% of those on mainstream web servers) run end-of-life software; 172,622 (89.15%) HTTPS services serve internal websites; 189,334 (97.78%) HTTPS services use insecure certificates.
DNS	212,886 (63.67%) allow ANY queries, possibly exploited for reflection amplification attacks.
NTP	22,244 (2.70%) allow monlist, possibly exploited for reflection amplification attacks.

TABLE 5: Security Risks of Affected Services

6. Analyzing Security Risks by Service

In this section, we present the detailed results from our primary measurements, focusing on the security implications of firewall misconfigurations on various services.

6.1. SSH Services

SSH provides secure remote access over insecure networks. We uncovered 234,984 SSH services behind misconfigured firewalls and picked 753,769 publicly accessible SSH services as the control group.

Measurement. We record the banners and handshakes of the SSH services. We determine their software versions by parsing banners. We also extract the supported cryptographic algorithms in the handshakes, as weak cryptographic algorithms can compromise the confidentiality and integrity of SSH channels as found in the Terrapin Attack [63].

Analysis. We list the most common vendors of the affected services in Table 6. The result shows that OpenSSH is predominant (94.8%) while some indicate embedded or network devices like `dropbear`, `ROSSSH`, `Cisco`, `HUAWEI`, and `DOPRA`. OpenSSH accounts for 76.0% of the control group services.

Given its dominant usage share, we compare the distributions of OpenSSH versions between the affected services and public services as in Table 7. Overall, the affected

Software	Count	%
OpenSSH	222,878	94.85%
AWS_SFTP	4,955	2.11%
dropbear	1,545	0.66%
ROSSSH	1,264	0.54%
Cisco	1,095	0.47%
HUAWEI	564	0.24%
DOPRA	232	0.10%
WeOnlyDo	92	0.04%
Others	2,451	1.04%

TABLE 6: SSH Software

Version	Affected	Public
2.x	0.11‰	0.02‰
3.x	0.16%	0.08%
4.x	0.34%	0.52%
5.x	2.81%	3.20%
6.x	3.77%	2.58%
7.x	54.07%	34.84%
8.x	32.25%	45.31%
9.x	6.55%	13.28%
Others	0.03%	0.20%

TABLE 7: OpenSSH Versions

services tend to run older versions, suggesting a higher chance of older operating systems and other outdated and vulnerable software.

We then check the patch progress of CVE-2024-6387, which is a recently discovered remote code execution (RCE) vulnerability in OpenSSH. We select the services within the affected version range and examine whether they are patched based on the patch level in the version. We list the results in Table 8, where *unidentified* means that the patch level is missing or there are no release notes. We note that a smaller portion of the affected services are known to be patched, and 47.92% are unidentified. Since OpenSSH in many distros like Debian [64], Ubuntu [65], and FreeBSD [66] provides patch levels and has explicit release notes, unidentified services might be customized versions and less secure.

	Vulnerable	Patched	Unidentified
Affected	4.71% (2,695)	47.36% (27,085)	47.92% (27,405)
Public	16.34% (28,919)	62.23% (110,131)	21.43% (37,918)

TABLE 8: Patching Progress of CVE-2024-6387

We inspect the most vulnerable cryptographic algorithms and list the results in Table 9. Compared with public SSH services, a larger portion of affected services support weak encryption or message authentication code algorithms. The usage share of weak key exchange algorithms is similar.

	Affected	Public
Encryption Algorithm		
arcfour{,128,256}	6.82% (16,022)	4.15% (31,266)
aes{128,192,256}-cbc	28.25% (66,383)	24.73% (186,392)
3des-cbc	26.94% (63,304)	21.04% (158,599)
blowfish-cbc	22.16% (52,071)	17.19% (129,541)
cast128-cbc	21.62% (50,801)	16.02% (120,768)
Message Authentication Code Algorithm		
hmac-md5	9.27% (21,780)	7.41% (55,870)
hmac-md5-96	8.28% (19,452)	4.31% (32,509)
Key Exchange Algorithm		
dh-group1-sha1	25.20% (59,224)	25.32% (190,818)
dh-group-exchange-sha1	26.51% (62,289)	25.23% (190,162)

TABLE 9: Weak Cryptographic Algorithms in SSH

Besides these attack vectors we have measured, SSH is also vulnerable to weak passwords [9]. Although we did not verify the prevalence of weak passwords due to ethical considerations, these affected services may be brute-forced and taken over if the firewalls are circumvented.

6.2. Telnet Services

Telnet allows users to access remote systems. We uncovered 50,820 Telnet services behind misconfigured firewalls and fetched their banners.

Measurement. We record the banners of Telnet services. Telnet usually prompts login when connected, and insecure systems may directly prompt shells without authentication.

Analysis. Upon connection, 98.06% (50,192) of affected services prompt login. While most of the other affected services return error messages or unidentifiable data, 31 directly present shells without any authentication.

Telnet does not support encryption and transmits all data in plaintext. However, it is still widely used for managing network devices and IoT devices [15], [67]. Consequently, attackers actively exploit Telnet to spread malware by leveraging brute-forcing or software vulnerabilities [68], [69]. While we did not verify these attack vectors due to ethical considerations, the compromise of network devices like routers and switches can constitute severe threats to infrastructures.

6.3. RDP Services

RDP allows users to manage Windows hosts via a graphical interface. We uncovered 7,931 RDP services behind misconfigured firewalls and picked 222,200 publicly accessible RDP services as the control group.

Measurement. We record the negotiation process of RDP services. We identify the OS version by the build number in the NTLMSSP [70] response during negotiation [53]. If NTLMSSP is unavailable, often due to old versions, we match the magic numbers in the responses [71].

Analysis. We list the OS distribution in Table 10. In total, 29.31% (2,324) of affected hosts run end-of-life (EOL) Windows versions, while the percentage for the publicly accessible hosts is only 9.70%.

OS	Affected	Public	EOL
Windows 2000/XP	0.52% (41)	0.15% (344)	✓
Windows 2003	9.51% (754)	3.17% (7,048)	✓
Windows 7/2008	19.28% (1,529)	6.38% (14,174)	✓
Windows 2012	10.68% (847)	30.13% (66,949)	
Windows 10/2016/2019	39.76% (3,153)	36.76% (81,688)	
Windows 11/2022	15.77% (1,251)	21.19% (47,083)	
Unidentified	4.49% (356)	2.21% (4,914)	

TABLE 10: OS Distribution of RDP Services

RDP has had a number of critical vulnerabilities [72], [73], [74], [75], mainly in the old Windows versions. Outdated Windows versions also have various vulnerabilities beyond the RDP service and can be compromised in minutes when connected to the Internet [10].

Furthermore, RDP is also prone to weak passwords, and compromise can lead to takeover, which we did not measure

due to ethical concerns. Still, exposure of RDP services can constitute a threat to these devices.

6.4. IPMI Services

IPMI is the industry standard protocol for out-of-band remote management, mostly for servers. It enables users to control the power, view the screen, and use the keyboard and mouse over the network. We uncovered 4,242 IPMI services behind misconfigured firewalls and picked 43,031 publicly accessible IPMI services as the control group.

Measurement. We probe IPMI services by sending a "Get Channel Authentication Capabilities" command, which reveals the authentication configuration.

Analysis. We examine and list the prevalence of three types of weak configurations [76], [77], [78] in Table 11. A large portion of affected services allow NONE authentication, which enables anyone to gain full control of the servers without any authentication. A relatively small but noticeable percentage of affected services allow the anonymous user, which is a low-privilege account without a password. Null usernames may also lead to anonymous login.

Type	Affected	Public
NONE Authentication	29.8% (1,264)	2.8% (1,185)
Anonymous Login	1.5% (64)	1.1% (469)
Null Usernames	8.7% (369)	23.5% (10,091)

TABLE 11: IPMI Weak Configuration Prevalence

IPMI is not intended for public access. Many IPMI implementations are flawed or shipped with fixed default passwords [79]. While we did not verify the vulnerabilities due to ethical considerations, the exposure of IPMI services is a serious attack surface per se.

6.5. SNMP Services

SNMP is widely used for telemetry and remote management, mainly for servers, routers, and switches. It exposes the device status and configuration in the form of variables.

SNMP has three main versions: SNMPv1, SNMPv2c, and SNMPv3. The first two are similar and insecure, while SNMPv3 has major improvements.

6.5.1. SNMPv1 and SNMPv2c. We uncovered 42,894 SNMPv1 services and 36,753 SNMPv2c services on 47,117 unique IP addresses behind misconfigured firewalls. We discuss the two services together and regard each IP address as a unique device.

Measurement. SNMPv1 and SNMPv2c have a weak security model, where the "community string" serves as the password. Most vendors use `public` as the default community string for read access, and `private` as the default community string for read-write access [80]. We probe SNMPv1 and SNMPv2c services by requesting the system description field with the community string `public`, which can reveal the device type and model.

Analysis. We classify the affected devices based on their system descriptions and list the results in Table 12. "Network device" includes switches, routers, firewalls, modems, etc.; "Appliance" contains other connected devices including printers, IP cameras, and VoIP gateways; "Empty Field" means that the system description field is empty; "Unidentified" means that we cannot identify the exact type of the device. The result shows that the firewall misconfigurations can affect a wide variety of devices.

Type	Count	%
Server	12,096	25.67%
Network Device	20,044	42.54%
Appliance	2,175	4.62%
Empty Field	11,260	23.90%
Unidentified	1,542	3.27%

TABLE 12: SNMPv1 & SNMPv2 Device Type

It is worth noting that SNMPv1 and SNMPv2 respond only when the community string is accepted, and hence, anyone can use the community string `public` to read all the variables of these affected devices. The variables may contain sensitive information like the device passwords and logs [81], varying by specific model and implementation. Due to ethical considerations, we did not use community strings other than `public`.

6.5.2. SNMPv3. We uncovered 465,033 SNMPv3 services behind misconfigured firewalls.

Measurement. SNMPv3 enhances security by supporting cryptographic authentication. We probe SNMPv3 services by sending an empty "get request" without a password. The response includes the device's engine ID, which indicates the device manufacturer and contains additional data, often the media access control (MAC) address.

Analysis. The engine ID can be used to fingerprint the device [82] and even brute force the password [83]. We examine the responses and identify the device manufacturers by the enterprise ID in the engine ID. We list the results in Table 13. Furthermore, 163,827 devices give away their MAC addresses without any authentication, enabling further fingerprinting and attacks.

Manufacturer	Category	Count	%
Cisco	Network Device	199,304	42.86%
Net-SNMP	Server	99,623	21.42%
Juniper	Network Device	51,346	11.04%
Nokia	Network Device	45,330	9.75%
MikroTik	Network Device	17,218	3.70%
Kyle Fox	Unknown	14,303	3.08%
SNMP Research	Server	10,691	2.30%
Others		27,218	5.85%

TABLE 13: SNMPv3 Manufacturers

6.6. FTP Services

FTP is an ancient protocol for file transfer. We uncovered 32,172 FTP services behind misconfigured firewalls.

Measurement. We record the banners of FTP services, which may provide the software versions and the capability of anonymous login.

Analysis. The banners of FTP servers are obscured and only 2,588 (8.04%) contain the software version. We list the identifiable servers in Table 14. In total, 70.83% of these servers run software that is end-of-life or outdated[7], which poses a high security risk. We also find that two affected services explicitly state in their banners that anonymous login is allowed. Furthermore, the affected services may be brute-forced if the firewalls are bypassed.

Software	Supported	EOL or Obsolete
FileZilla	0	13.49% (349)
MikroTik	17.39% (450)	10.36% (268)
ProFTPD	0	12.67% (328)
Serv-U FTP	0	1.31% (34)
vsFTPd	11.79% (305)	33.00% (854)

TABLE 14: Identifiable FTP Services

6.7. SMB Services

SMB is primarily used for file sharing on Windows. We uncovered 19,419 SMB services behind misconfigured firewalls and picked 41,483 publicly accessible SMB services as the control group.

Measurement. ZGrab supports SMB but only has limited functionalities. To study the SMB services better, we adopt smbmap [84]. We customize it to add support for specifying source ports and enhance its performance and stability. We try to establish anonymous SMB sessions and enumerate the unprotected SMB shares.

Analysis. We find 13,220 (68.08%) of affected services allowing anonymous sessions, while only 10,129 (24.42%) of the control group allow anonymous sessions. We discuss only the services allowing anonymous sessions below.

In total, 96 of the affected services have 202 unprotected shares, allowing anonymous read. This poses a great threat as the data is supposed to be protected by firewalls. We did not access any files in the shares due to ethical considerations.

We further analyze the affected hosts based on OS types.

Windows Hosts. We find that 1,012 (7.66%) of the affected services and 6,209 (61.30%) of the public services run Windows. We list the OS distributions in Table 15. In total, 63.44% of affected hosts run end-of-life Windows versions, while the percentage for the public services is only 44.40%. This suggests that the affected hosts may be susceptible to various vulnerabilities like EternalBlue [85] which enables zero-click RCE though we did not verify prevalence.

Non-Windows Hosts. SMB is supported by third-party software like Samba and can run on non-Windows platforms. We find 12,208 (92.34%) affected hosts run non-Windows operating systems, which is an unexpectedly high

7. We deem a software version "outdated" if it was released more than three years ago and we see no sign of maintenance.

OS	Affected	Public	EOL
Windows 2000/XP	1.19% (12)	0.02% (1)	✓
Windows 2003	4.74% (48)	0.06% (4)	✓
Windows 7/2008	44.37% (449)	32.08% (1,992)	✓
Windows 8/8.1	13.14% (133)	12.24% (760)	✓
Windows 2012	0	0.05% (3)	
Windows 10/2016/2019	26.79% (271)	41.78% (2,594)	
Windows 11/2022	9.78% (99)	13.77% (855)	

TABLE 15: OS Distribution of SMB Services on Windows

figure. After investigation, we find 10,890 services with hostnames starting with `LINKSYS`, which appear to be Linksys routers. Further analysis reveals that these routers are customized and have an RCE vulnerability which can be exploited from the Internet. We discuss this case in §8.1.

6.8. MySQL Services

MySQL is a widely used relational database management system. It has a highly compatible fork, MariaDB, using the same protocol. We regard both as MySQL services. We uncovered 19,456 MySQL services behind misconfigured firewalls and picked 249,207 publicly accessible MySQL services as the control group.

Measurement. We record the banners of the MySQL services, which reveal the server software and versions. If a server only allows access from trusted hosts, our connection will be aborted with an error message without a banner.

Analysis. We examine the distribution of software versions and list the results in Table 16. Overall, 54.08% of the affected services run end-of-life MySQL or MariaDB versions, whereas the percentage in public services is only 35.60%. In addition, a lower portion of affected services deny our connections ("No Access" row in the table), which indicates more lax access control.

	Affected	Public
MySQL		
End-of-Life Versions	27.17% (5,286)	28.62% (71,321)
Supported Versions	2.81% (547)	14.92% (37,182)
MariaDB		
End-of-Life Versions	26.91% (5,236)	6.98% (17,396)
Supported Versions	13.94% (2,712)	11.87% (29,575)
Total		
End-of-Life Versions	54.08% (10,522)	35.60% (88,717)
Supported Versions	16.75% (3,259)	26.79% (66,757)
No Access	29.17% (5,675)	37.61% (93,733)

TABLE 16: MySQL Version Distribution

Like any other database product, MySQL is vulnerable to software flaws and weak passwords, and compromise can lead to information leakage and even local privilege escalation [86]. Exposure and outdated software versions significantly increase the risks.

6.9. MongoDB Services

MongoDB is a widely adopted NoSQL database product. We uncovered 338 MongoDB services behind misconfigured firewalls and picked 15,917 publicly accessible MongoDB services as the control group.

Measurement. We try to fetch the banner and database list of the MongoDB services, which is common practice for search engines like Shodan. MongoDB disables authentication by default and is a frequent target of ransomware [87].

Analysis. We list the statistics in Table 17. The results show that a much higher percentage of affected MongoDB services have no protection at all. This indicates that the administrators might assume that the services are protected by firewalls and do not properly secure them. Importantly, we note that most (76.33%) unsecured public MongoDB services have been compromised by ransomware, with a database named `READ__ME_TO_RECOVER_YOUR_DATA` or so as an indicator of compromise (IOC). However, the percentage for affected MongoDB services is only 2%, whose firewalls may be set up after compromise. This suggests that this firewall bypass technique has not been actively exploited against MongoDB services in the wild.

	No Authentication	Compromised
Affected	29.59% (100)	2 (2.00% of the subset)
Public	8.49% (1,352)	1,032 (76.33% of the subset)

TABLE 17: MongoDB Service Statistics

We further analyze the 100 affected and unprotected MongoDB services. We find a total of 790 databases, among which four have over 1 TB of data, 37 have over 100 GB of data, and 98 have over 1 GB of data. The failure of firewalls may impose disastrous impacts on these services.

With ethics in mind, we only list the names and sizes of databases for analyzing IOCs and impacts. We did not retrieve any data inside the databases.

6.10. HTTP(S) Services

HTTP(S) is the fundamental protocol to access websites. We uncovered 222,539 HTTP services and 193,630 HTTPS services behind misconfigured firewalls and picked 851,011 public HTTP services and 708,968 public HTTPS services as the control group.

Measurement. Since the protected websites may contain sensitive information, due to ethical considerations, we send HTTP HEAD requests to get the headers without web page content, which reveal the server software. In addition, we record the TLS handshakes with the HTTPS services, which include the certificates of the websites.

Analysis. We first analyze the TLS certificates used by the HTTPS services and list the risk factors in Table 18. Most affected services appear to serve internal websites with certificates containing internal names, e.g., private IP addresses, illegal domain names, and private top-level domains like `.lan` and `.localdomain`. Compared the public services, a significantly larger portion of affected services use insecure certificates, such as self-signed certificates and certificates valid for more than 397 days. While the affected services are less likely to use expired certificates, shorter keys, or weaker cryptographic algorithms, the exposure of internal services already constitutes a serious security risk.

Risk Factor	Affected	Public
Internal Names	89.15% (172,622)	15.46% (109,595)
Self-Signed Cert.	41.24% (79,856)	16.64% (117,979)
Long Validity	73.25% (163,010)	21.00% (148,883)
Expired Cert.	10.22% (19,788)	11.33% (80,306)
Short RSA Key Size	1.11% (2,158)	2.82% (19,993)
SHA-1 Signature	5.57% (10,794)	9.38% (66,485)

TABLE 18: Security Risks Based on Certificates

We then identify the server software and versions via the `Server` header in the responses. We list the results in Table 19. Overall, the software is highly diverse. Compared with the public services with regard to three mainstream web servers, a higher percentage of affected services use end-of-life versions; fewer affected services conceal their version information, potentially offering clues to attackers. A significant portion (50.21%) of affected services run small-footprint servers such as lighttpd, thttpd, and mini_httpd. This high prevalence suggests that they are likely to be embedded devices, including network devices and industrial control systems. The exposure of these services may impact critical infrastructures.

	Affected	Public
Apache	**4.21% (17,512)**	**14.13% (220,366)**
End-of-Life Versions	50.69% (8,877)	7.47% (16,462)
Supported Versions	33.79% (5,918)	40.38% (88,984)
Unknown Versions	15.52% (2,717)	52.15% (114,920)
NGINX	**4.82% (19,221)**	**21.17% (330,308)**
End-of-Life Versions	59.69% (11,473)	34.87% (115,177)
Supported Versions	0.59% (113)	2.39% (7,889)
Unknown Versions	39.72% (7,635)	62.74% (207,242)
Microsoft IIS	**0.45% (1,892)**	**3.09% (48,225)**
End-of-Life Versions	17.49% (331)	12.07% (5,820)
Supported Versions	82.51% (1,561)	87.90% (42,391)
Unknown Versions	0	0.03% (14)
Squid	23.56% (98,042)	0.49% (7,597)
lighttpd	6.57% (27,353)	1.02% (15,923)
thttpd	22.73% (94,587)	0.13% (2,047)
mini_httpd	20.91% (87,019)	0.09% (1,423)
CDN Providers	0	18.65% (290,929)
Other	16.95% (70,543)	41.23% (643,162)

TABLE 19: Server Software of HTTP(S) Services

6.11. DNS Services

DNS provides domain name resolution. We uncovered 334,358 DNS services behind misconfigured firewalls. The number is as high as 6.32% of all public DNS servers. We also picked 475,592 public DNS servers as the control group.

Measurement. We host a name server and set up a domain name for the measurement. We probe DNS services by sending A queries for our domain name, and the RA flag in the responses shows whether they support recursive queries. We also send ANY queries for our domain name to check whether the servers respond because the ANY query may be exploited to launch reflection amplification attacks [31] and was effectively deprecated by RFC8482 [88] in 2019.

Analysis. The results show that 87.40% (292,229) of affected services support recursive queries, while the percentage for public services is only 27.99% (133,102). DNS servers that support recursive queries have higher risks [89], e.g., reflection amplification attacks and cache poisoning.

The result of ANY queries further proves this. 63.67% (212,886) of affected services respond to ANY queries, whereas only 15.37% (73,119) public services respond. This suggests that a high percentage of DNS services behind misconfigured firewalls may be leveraged to launch attacks if they are exposed to attackers.

Various DNS servers are vulnerable to DNS cache poisoning [90], [91], [92], which can compromise the integrity of DNS responses. However, we did not verify these specific vulnerabilities due to ethical considerations.

6.12. NTP Services

NTP is used for time synchronization. We uncovered 824,389 NTP services behind misconfigured firewalls. The number is as high as 12.60% of all public NTP servers. We also picked 978,751 publicly accessible NTP services as the control group.

Measurement. We probe NTP services by sending time requests. We also send monlist queries to check whether the servers respond because misconfigured NTP servers supporting monlist can be used for amplification attacks [32].

Analysis. The result shows that 2.70% (22,244) of affected NTP services respond to monlist queries, which is higher than the number in public NTP servers, i.e., 1.63% (15,993). By bypassing the firewalls, attackers can expand their arsenal considerably.

7. Current Exploitation Status in the Wild

The measurement reveals the widespread existence of firewall misconfigurations that mistakenly allow inbound connections from certain ports. To investigate whether such misconfigurations are actively exploited in the wild, we set up honeypots to capture connections from special ports.

Experiment Setup. We deploy honeypots in three locations: the United States, Japan, and Hong Kong. Each honeypot hosts a program listening on the ports of our selected services, as detailed in §4.2. This program records all received connections and data. To simulate a misconfigured firewall, we configure iptables with stateless rules that only allow inbound packets from specific TCP ports (22, 80, 443) and UDP ports (53, 123), which correspond to popular services and are likely to be exploited by potential attackers. We verify that the honeypots work correctly with netcat. We also exclude their IP addresses from our measurements to prevent data contamination. The experiment lasted four months, from mid-June 2024 to mid-October 2024.

Result and Analysis. During the four-month period, the honeypots received:

- 3 SNMPv1 requests at UDP port 161 from port 53.
- 65 DNS queries at UDP port 53 from port 53.
- 104 NTP requests at UDP port 123, 6 from port 53, and 98 from port 123.

The result shows very limited scanning activities. We further investigate the source of such scanning and find that most are likely benign and/or unintentional.

Datagrams from Port 53. We find that all of the 74 datagrams from port 53 are from cybersecurity companies or researchers, judging by the source addresses, rDNS records, and payloads. Besides, the choice of source port may be unintentional. For example, scapy is a popular Python library for creating network packets, and 53 is the default source port for UDP datagrams [93].

Datagrams from Port 123. We find that 30 of the 98 datagrams from port 123 are from cybersecurity companies, judging by the source address and rDNS records. We investigate the remaining senders and find that they are all home IP addresses in East Asia and have been marked as NTP scanners on AbuseIPDB, probably from a botnet.

Summary. The honeypots received no TCP connections and only 172 UDP datagrams from cybersecurity companies, researchers, and one botnet. We cannot verify the reason for their source port choice, and the traffic is negligible compared to the everyday scanning activities on the Internet. Hence, we conclude that there is no large-scale exploitation in the wild and do not suspect any intentional exploitation.

8. Case Studies

Besides the primary measurement at the Internet scale, we find two products shipped with the firewall misconfigurations studied in this paper. We elaborate these two cases and their security implications in this section.

8.1. Linksys Home Router

In our measurement of SMB services, we find 10,890 affected services with hostnames starting with `LINKSYS`. Linksys is a popular network hardware manufacturer and many of its home routers support sharing external storage devices via SMB. This indicates that a large number of Linksys routers are affected by the firewall misconfiguration that allows inbound TCP connections initiated from port 80.

Investigation. We check the IP addresses of the affected Linksys routers and find that they all belong to an ISP named Truespeed in the United Kingdom. According to the official websites [94], [95], Truespeed partners with Linksys to provide its customers with customized Linksys routers based on Linksys Velop AX4200, which is a mesh router.

Linksys mesh routers were known to contain an RCE vulnerability in its inter-AP communication protocol [96].

While the firmware of the customized model is not publicly available, we analyze the firmware of its original model, Linksys Velop AX4200, instead. To our surprise, the vulnerability still exists in the latest firmware. We further confirm that the port of the vulnerable service, TCP 6060, becomes reachable when initiating connections from port 80. This suggests that this vulnerability may be remotely exploited.

We do not find any explanation for this firewall misconfiguration in the firmware of the original model. Considering the fact that we do not find other vulnerable Linksys devices outside of this ISP, it is likely that only the customized model has this flawed rule, probably in iptables.

Disclosure. We disclosed the misconfiguration to Linksys with a detailed report in September 2024. However, as of the time of writing, we have received no substantial response from Linksys, and the affected routers still show up in follow-up measurements.

Lessons Learned. Firewall misconfigurations in this work may make local vulnerabilities remotely exploitable. Furthermore, embedded devices may be shipped with flawed firewall rules, which can be difficult to patch afterward.

8.2. Oracle Cloud Ubuntu Image

We research several mainstream public clouds for their built-in firewall rules. While most clouds provision services with no firewall rule or secure firewall rules, the Ubuntu images on Oracle Cloud come with an insecure iptables rule which allows any inbound UDP datagram from port 123.

Investigation. We find that the official Ubuntu images on Oracle Cloud forbid inbound connections by default but contain a flawed iptables rule: `-A INPUT -p udp -m udp -sport 123 -j ACCEPT`. It opens a loophole allowing any UDP datagram from port 123 and is preinstalled on all virtual machines using the official Ubuntu images.

Based on the port number, the rule in question should be intended for NTP responses. The secure version should be stateful, allowing inbound packets only if the host has initiated a UDP session first: `-A INPUT -p udp -sport 53 -m state -state ESTABLISHED -j ACCEPT`.

Oracle Cloud has hundreds of thousands of customers, including big companies and governments [97], and Ubuntu is one of the most popular Linux distros. There may be a large number of hosts affected by this misconfiguration.

Disclosure. We disclosed the misconfiguration to Oracle with a detailed report in October 2024. As of the time of writing, Oracle Cloud has removed the flawed rule in its Ubuntu images.

Lessons Learned. Flawed firewall rules still may come with operating systems, even in the cloud era. Preloading firewalls with flawed rules may be more dangerous than no firewall at all, as it may create a false sense of security.

9. Responsible Disclosure

In this section, we focus on disclosures relating to the affected hosts and services.

Recipient Selection. It is extremely difficult to determine the actual owner of a host given only the IP address, especially considering our avoidance of collecting sensitive information. We choose to disclose the affected hosts and services to the contact emails of their corresponding ASes and kindly request them to forward the message to the actual owners if they are not. We obtain the contact emails via WHOIS. For the few ASes without contact emails, we choose the support or feedback emails on their websites.

Form of Disclosure. We send emails to the recipients, which include (i) a brief introduction of the vulnerability and potential security implications, (ii) the list of affected hosts and services in the AS, and (iii) a link to our website, providing technical details and possible fixes. We also explain the reason for scanning and provide methods to opt out of this study.

Feedback. We started the disclosure in mid-October and finished in one month. As of the time of writing, the website has been viewed by recipients from 1,962 ASes. We have received inquiries, updates, and letters of thanks from 290 ASes, many of which report that they have successfully patched the vulnerability or that they have forwarded the message to their customers. Some indicate that they will update their security tools to cover this attack surface. We received one bounty from a company for discovering their misconfiguration. Our disclosure also led to the discovery of a high-severity CVE [98] in Open Virtual Network, a network virtualization platform.

Some recipients share the root causes of their misconfigurations, and the most common reason is flawed stateless rules induced by carelessness. One special case is that the recipient is aware of the misconfiguration but has to keep it because of a firmware bug in their Cisco stateless firewall. The bug prevents TCP flag filtering from working, and they can only deploy flawed rules instead. They have submitted a ticket to Cisco but have not heard back.

We received and honored five opt-out requests in the replies to our disclosure.

Effectiveness of Disclosure for Top ASes. We perform follow-up measurements before disclosure and every two weeks after disclosure. We scan the top 202 ASes with at least 1,000 affected hosts, which account for 83.07% of the total, from one vantage point in the United States. This reduced scope can decrease our scanning traffic by ∼93% but still accomplish very high coverage. We adopt the same workflow and target services in §4 and count the numbers of affected hosts. The results are shown in Table 20. The first row is the number of the affected hosts from the primary measurement in §5 with only the data collected at the US vantage point, serving as a benchmark. We also list the percentage changes compared to the previous measurement.

	Primary (US)	Week 0	Week 2	Week 4
Total	1,718,659	1,706,378	1,626,073	1,617,107
Change (%)	N/A	-0.71%	-4.71%	-0.55%

TABLE 20: Follow-up Measurements after Disclosure

From the results, we see a small but noticeable drop in the number of affected hosts between Week 0 and Week 2. However, the number of Week 4 only has a negligible decrease, probably due to network volatility. The measurement results show that only 11 of the 202 ASes have fixed the issue (having less than 10% of the original number of affected hosts). We hypothesize that the most affected ASes have a lot of IP addresses and the corresponding devices are managed by their customers instead of the recipients.

Summary. We perform responsible disclosure and receive positive feedback. Our emails and website have helped many ASes patch their firewalls. Some update their security tools accordingly. We also learn the causes of the misconfigurations in some ASes, including carelessness and flawed firmware. In the follow-up measurements, we see a small but noticeable decrease in the number of affected hosts in the most affected ASes.

10. Discussion

The IPv6 Space. With the growing adoption of IPv6, it is expected that every device gets a public IPv6 address. This shift will make the problem of firewall misconfiguration even more important. We leave it to future research to conduct a comprehensive measurement in the IPv6 space.

Bad Advice on the Internet. Search engines and Q&A platforms are common sources of information, including firewall configurations. However, suggestions for insecure firewall rules are widespread and are often most upvoted. This is particularly common with regard to allowing UDP protocols like DNS [99] and NTP [100]. Stateless rules are favored because of their simplicity, and stateful rules may appear obscured for UDP. While rare, flawed rules for TCP ports also exist [101]. Such information sources on the Internet may have contributed to the prevalence of firewall misconfigurations and need remediation.

Mitigations and Best Practices. Stateful rules should be used whenever circumstances permit, as stateless rules are more prone to misconfigurations by nature. If stateless firewalls/rules must be used, it is recommended to utilize TCP flag filtering to block inbound connections (if the firewall supports it) and whitelist only necessary UDP traffic to and from trusted servers. Another workaround is to specifically block the traffic to and from the ports of sensitive services. However, these methods may produce side effects or have omissions. Last but not least, it is important to regularly review and update the firewall rules.

11. Ethical Considerations

We follow the best practices as in prior works [20], [38], including providing opt-out options and setting up a website explaining the study. Anyone accessing the HTTP port of our scanners will be redirected to the website. We also attach the opt-out instruction in our probes when the protocol permits, like in the `User-Agent` header of HTTP. During our study, we received and honored 17 opt-out requests. To minimize network pressure and avoid overwhelming Internet hosts, we focus on a small number of services and source port numbers, and we use scanning tools that have been tested and repeatedly used by prior studies [20]. Our scanners always follow protocol specifications, and we immediately close connections after a response is received (both during port scanning and the application-layer probing).

In addition, during our measurements, we collected minimum data that allows us to analyze the services' characteristics and security risks. We avoid collecting any potentially sensitive data like web pages, files, or database content. We did not perform any aggressive activities, such as cracking passwords or exploiting vulnerabilities.

Despite our efforts, our probe traffic may still be deemed suspicious by automated systems like intrusion detection systems and honeypots. We received 25 complaints during our measurement, mostly only because of sending packets to their IP addresses. We confirmed that no harm was done and properly handled all the complaints by immediately stopping any measurement of the complaining IP addresses.

We also ensure responsible disclosure. After our measurements, we notified all involved parties about the firewall misconfigurations in their networks or products. The procedure and feedback are detailed in §8 and §9.

12. Conclusion

In this paper, we reveal the widespread existence of a previously understudied attack surface, namely firewall misconfigurations that inadvertently allow inbound connections from certain ports. We further analyze the security implications and exploitation status in the wild. We also perform responsible disclosure to the involved parties. Finally, we discuss future directions and best practices for mitigating this attack surface.

Acknowledgment

The authors would like to thank the reviewers for their valuable feedback during the revision process. This research was sponsored by the OUSD(R&E)/RT&L and was accomplished under Cooperative Agreement Number W911NF-20-2-0267. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ARL and OUSD(R&E)/RT&L or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes, notwithstanding any copyright notation herein.

References

[1] Z. Durumeric, M. Bailey, and J. A. Halderman, "An Internet-Wide view of Internet-Wide scanning," in *23rd USENIX Security Symposium (USENIX Security 14)*. San Diego, CA: USENIX Association, Aug. 2014, pp. 65–78. [Online]. Available: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/durumeric

[2] H. Griffioen, G. Koursiounis, G. Smaragdakis, and C. Doerr, "Have you syn me? characterizing ten years of internet scanning," in *Proceedings of the 2024 ACM on Internet Measurement Conference*, ser. IMC '24, 2024, p. 149–164.

[3] J. Mazel, R. Fontugne, and K. Fukuda, "Profiling internet scanners: Spatiotemporal structures and measurement ethics," in *2017 Network Traffic Measurement and Analysis Conference (TMA)*, 2017, pp. 1–9.

[4] R. Hiesgen, M. Nawrocki, A. King, A. Dainotti, T. C. Schmidt, and M. Wählisch, "Spoki: Unveiling a new wave of scanners through a reactive network telescope," in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 431–448. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/hiesgen

[5] A. Anand, M. Kallitsis, J. Sippe, and A. Dainotti, "Aggressive internet-wide scanners: Network impact and longitudinal characterization," in *Companion of the 19th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT 2023, 2023.

[6] W. K. Leong, A. Kulkarni, Y. Xu, and B. Leong, "Unveiling the hidden dangers of public ip addresses in 4g/lte cellular data networks," in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '14. New York, NY, USA: Association for Computing Machinery, 2014. [Online]. Available: https://doi.org/10.1145/2565585.2565599

[7] P. Richter and A. Berger, "Scanning the scanners: Sensing the internet from a massively distributed network telescope," in *Proceedings of the Internet Measurement Conference*, ser. IMC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 144–157. [Online]. Available: https://doi.org/10.1145/3355369.3355595

[8] L. Bilge and T. Dumitraş, "Before we knew it: an empirical study of zero-day attacks in the real world," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 833–844. [Online]. Available: https://doi.org/10.1145/2382196.2382284

[9] S. K. Singh, S. Gautam, C. Cartier, S. Patil, and R. Ricci, "Where the wild things are: Brute-Force SSH attacks in the wild and how to stop them," in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. Santa Clara, CA: USENIX Association, Apr. 2024, pp. 1731–1750. [Online]. Available: https://www.usenix.org/conference/nsdi24/presentation/singh-sachin

[10] E. Parker, "What happens if you connect windows xp to the internet in 2024?" https://www.youtube.com/watch?v=6uSVVCmOH5w, may 2024, (Accessed on 09/03/2024).

[11] M. Boddy, "Exposed: Cyberattacks on cloud honeypots," https://assets.sophos.com/X24WTUEQ/at/rgbjvgnx6qwwj7wvx764rmbn/sophos-exposed-cyberattacks-on-cloud-honeypots-wp.pdf, arp 2019, (Accessed on 09/03/2024).

[12] Microsoft, "Ipsec default exemptions can be used to bypass ipsec protection in some scenarios," https://web.archive.org/web/20140530061933/https://support.microsoft.com/kb/811832, (Accessed on 09/07/2024).

[13] "Discovering mac os x weaknesses and fixing them with the new bastille os x port," https://bastille-linux.sourceforge.net/jay/dc14.pdf, (Accessed on 09/07/2024).

[14] G. Lyon, "Bypassing firewall rules," https://nmap.org/book/firewall-subversion.html, 2008, (Accessed on 09/08/2024).

[15] A. Cui and S. J. Stolfo, "A quantitative analysis of the insecurity of embedded network devices: results of a wide-area scan," in *Proceedings of the 26th Annual Computer Security Applications Conference*, ser. ACSAC '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 97–106. [Online]. Available: https://doi.org/10.1145/1920261.1920276

[16] B. Zhao, S. Ji, W.-H. Lee, C. Lin, H. Weng, J. Wu, P. Zhou, L. Fang, and R. Beyah, "A large-scale empirical study on the vulnerability of deployed iot devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1826–1840, 2022.

[17] M. Hastings, J. Fried, and N. Heninger, "Weak keys remain widespread in network devices," in *Proceedings of the 2016 Internet Measurement Conference*, ser. IMC '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 49–63. [Online]. Available: https://doi.org/10.1145/2987443.2987486

[18] L. Quan, J. Heidemann, and Y. Pradkin, "Trinocular: understanding internet reliability through adaptive probing," in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, ser. SIGCOMM '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 255–266. [Online]. Available: https://doi.org/10.1145/2486001.2486017

[19] R. Fontugne, J. Mazel, and K. Fukuda, "An empirical mixture model for large-scale rtt measurements," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2470–2478.

[20] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast internet-wide scanning and its security applications," in *22nd USENIX Security Symposium (USENIX Security 13)*. Washington, D.C.: USENIX Association, Aug. 2013, pp. 605–620. [Online]. Available: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric

[21] R. Graham, "robertdavidgraham/masscan: Tcp port scanner, spews syn packets asynchronously, scanning entire internet in under 5 minutes." https://github.com/robertdavidgraham/masscan, (Accessed on 09/05/2024).

[22] B. Hou, Z. Cai, K. Wu, T. Yang, and T. Zhou, "6scan: A high-efficiency dynamic internet-wide ipv6 scanner with regional encoding," *IEEE/ACM Transactions on Networking*, vol. 31, no. 4, pp. 1870–1885, 2023.

[23] G. Song, J. Yang, L. He, Z. Wang, G. Li, C. Duan, Y. Liu, and Z. Sun, "AddrMiner: A comprehensive global active IPv6 address discovery system," in *2022 USENIX Annual Technical Conference (USENIX ATC 22)*. Carlsbad, CA: USENIX Association, Jul. 2022, pp. 309–326. [Online]. Available: https://www.usenix.org/conference/atc22/presentation/song

[24] G. Williams, M. Erdemir, A. Hsu, S. Bhat, A. Bhaskar, F. Li, and P. Pearce, "6sense: Internet-Wide IPv6 scanning and its security applications," in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 2281–2298. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/williams

[25] "Shodan search engine," https://www.shodan.io/, (Accessed on 09/06/2024).

[26] "Censys search," https://search.censys.io/, (Accessed on 09/06/2024).

[27] "Fofa search engine," https://en.fofa.info/, (Accessed on 09/06/2024).

[28] "Ripe atlas," https://atlas.ripe.net/, (Accessed on 09/06/2024).

[29] T.-F. Tu, J.-W. Qin, H. Zhang, M. Chen, T. Xu, and Y. Huang, "A comprehensive study of mozi botnet," *International Journal of Intelligent Systems*, vol. 37, no. 10, pp. 6877–6908, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/int.22866

[30] C. Cimpanu, "A crypto-mining botnet has been hijacking mssql servers for almost two years," https://www.zdnet.com/article/a-crypto-mining-botnet-has-been-hijacking-mssql-servers-for-almost-two-years/, arp 2020, (Accessed on 09/06/2024).

[31] Cloudflare, "Dns amplification ddos attack," https://www.cloudflare.com/learning/ddos/dns-amplification-ddos-attack/, (Accessed on 09/06/2024).

[32] ——, "Ntp amplification ddos attack," https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/, (Accessed on 09/06/2024).

[33] S. Bano, P. Richter, M. Javed, S. Sundaresan, Z. Durumeric, S. J. Murdoch, R. Mortier, and V. Paxson, "Scanning the internet for liveness," *SIGCOMM Comput. Commun. Rev.*, vol. 48, no. 2, p. 2–9, may 2018. [Online]. Available: https://doi.org/10.1145/3213232.3213234

[34] J. Klick, S. Lau, M. Wählisch, and V. Roth, "Towards better internet citizenship: Reducing the footprint of internet-wide scans by topology aware prefix selection," in *Proceedings of the 2016 Internet Measurement Conference*, ser. IMC '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 421–427. [Online]. Available: https://doi.org/10.1145/2987443.2987457

[35] L. Izhikevich, R. Teixeira, and Z. Durumeric, "LZR: Identifying unexpected internet services," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 3111–3128. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/izhikevich

[36] ——, "Predicting ipv4 services across all ports," in *Proceedings of the ACM SIGCOMM 2022 Conference*, ser. SIGCOMM '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 503–515. [Online]. Available: https://doi.org/10.1145/3544216.3544249

[37] G. Song, L. He, T. Zhao, Y. Luo, Y. Wu, L. Fan, C. Li, Z. Wang, and J. Yang, "Which doors are open: Reinforcement learning-based internet-wide port scanning," in *2023 IEEE/ACM 31st International Symposium on Quality of Service (IWQoS)*, 2023, pp. 1–10.

[38] G. Wan, L. Izhikevich, D. Adrian, K. Yoshioka, R. Holz, C. Rossow, and Z. Durumeric, "On the origin of scanning: The impact of location on internet-wide scans," in *Proceedings of the ACM Internet Measurement Conference*, ser. IMC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 662–679. [Online]. Available: https://doi.org/10.1145/3419394.3424214

[39] T. Rytilahti and T. Holz, "On using application-layer middlebox protocols for peeking behind nat gateways," in *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. The Internet Society, 2020. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/on-using-application-layer-middlebox-protocols-for-peeking-behind-nat-gateways/

[40] X. Feng, S. Chen, and H. Wang, "An internet-wide penetration study on nat boxes via tcp/ip side channel," 2023. [Online]. Available: https://arxiv.org/abs/2311.17392

[41] "Cve-2001-0851," https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0851, mar 2002, (Accessed on 09/17/2024).

[42] Éric Leblond, "Playing with network layers to bypass firewalls' filtering policy – to linux and beyond !" https://home.regit.org/2012/03/playing-with-network-layers-to-bypass-firewalls-filtering-policy/, mar 2012, (Accessed on 09/16/2024).

[43] S. Kamkar, "Nat pinning," https://samy.pl/natpin/, jan 2010, (Accessed on 09/16/2024).

[44] S. Sahay, "Stateful vs. stateless firewall: Differences explained," https://www.connectwise.com/blog/cybersecurity/stateful-vs-stateless-firewall, jun 2023, (Accessed on 09/10/2024).

[45] Cisco, "Configure commonly used ip acls," https://www.cisco.com/c/en/us/support/docs/ip/access-lists/26448-ACLsamples.html, nov 2023, (Accessed on 09/10/2024).

[46] D. US, "Dell emc networking command line reference guide for the s5048f–on system 9.14.2.6," [Online; accessed 2025-04-01]. [Online]. Available: https://www.dell.com/support/manuals/en-us/dell-emc-os-9/s5048f-on-9.14.2.6-cli-pub/deny-tcp-for-extended-ip-acls?guid=guid-a610d67e-86bb-41de-902d-3c6e23fa6ff7&lang=en-us

[47] Cisco, "Cli book 2: Cisco asa series firewall cli configuration guide, 9.12," https://www.cisco.com/c/en/us/td/docs/security/asa/asa912/configuration/firewall/asa-912-firewall-config/conns-connlimits.html, mar 2019, (Accessed on 09/10/2024).

[48] L. Yuan, H. Chen, J. Mai, C.-N. Chuah, Z. Su, and P. Mohapatra, "Fireman: a toolkit for firewall modeling and analysis," in *2006 IEEE Symposium on Security and Privacy (SP'06)*, 2006, pp. 15 pp.–213.

[49] T. E. Uribe and S. Cheung, "Automatic analysis of firewall and network intrusion detection system configurations," in *Proceedings of the 2004 ACM Workshop on Formal Methods in Security Engineering*, ser. FMSE '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 66–74. [Online]. Available: https://doi.org/10.1145/1029133.1029143

[50] H. Hu, G.-J. Ahn, and K. Kulkarni, "Detecting and resolving firewall policy anomalies," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 3, pp. 318–331, 2012.

[51] H. Pan, Z. Li, P. Zhang, P. Cui, K. Salamatian, and G. Xie, "Misconfiguration-free compositional sdn for cloud networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 2484–2499, 2023.

[52] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, and J. Yusupov, "Automated firewall configuration in virtual networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1559–1576, 2023.

[53] "nmap/scripts/rdp-ntlm-info.nse · nmap/nmap," https://github.com/nmap/nmap/blob/e263e648207995e491b6ffe3a94e24c3fbd4faba/scripts/rdp-ntlm-info.nse#L153, (Accessed on 09/29/2024).

[54] GlaDiaT0R, "Bypassing firewalls using an ftp," 3 2010, [Online; accessed 2025-04-01]. [Online]. Available: https://www.exploit-db.com/papers/13651

[55] Dataplane.org, "Ephemeral source port selection strategies," https://dataplane.org/analysis/ephemeralports.html, (Accessed on 09/10/2024).

[56] I. Tsareva, T. V. Doan, and V. Bajpai, "A decade long view of internet traffic composition in japan," in *2023 IFIP Networking Conference (IFIP Networking)*, 2023, pp. 1–9.

[57] L. Schumann, T. V. Doan, T. Shreedhar, R. Mok, and V. Bajpai, "Impact of evolving protocols and covid-19 on internet traffic shares," 2022. [Online]. Available: https://arxiv.org/abs/2201.00142

[58] "zmap/zmap: Zmap is a fast single packet network scanner designed for internet-wide network surveys." https://github.com/zmap/zmap/, (Accessed on 09/16/2024).

[59] "zmap/zgrab2: Fast go application scanner," https://github.com/zmap/zgrab2/, (Accessed on 09/16/2024).

[60] A. Bhaskar and P. Pearce, "Many roads lead to rome: How packet headers influence DNS censorship measurement," in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 449–464. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/bhaskar

[61] ——, "Understanding routing-induced censorship changes globally," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 437–451. [Online]. Available: https://doi.org/10.1145/3658644.3670336

[62] "Cve-2024-6387," https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-6387, jun 2024, (Accessed on 09/27/2024).

[63] F. Bäumer, M. Brinkmann, and J. Schwenk, "Terrapin attack: Breaking SSH channel integrity by sequence number manipulation," in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 7463–7480. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/b%C3%A4umer

[64] "debian/rules · 2df9bff12640a33749f0f20ae806b6efac327116 · debian ssh maintainers / openssh · gitlab," https://salsa.debian.org/ssh-team/openssh/-/blob/2df9bff12640a33749f0f20ae806b6efac327116/debian/rules#L36, (Accessed on 09/28/2024).

[65] "debian/rules - ubuntu/+source/openssh," https://git.launchpad.net/ubuntu/+source/openssh/tree/debian/rules?h=ubuntu/noble-security&id=284288abcc5d0d41f890460e9a60af4472c42627#n36, (Accessed on 09/28/2024).

[66] "security/openssh-portable/files/extra-patch-version-addendum - ports - freebsd ports tree," https://cgit.freebsd.org/ports/tree/security/openssh-portable/files/extra-patch-version-addendum?id=6014ebaef2fc946f5fc971cd4c6d882acca60098#n5, (Accessed on 09/28/2024).

[67] T. Sasaki, T. Noma, Y. Morii, T. Shimura, M. v. Eeten, K. Yoshioka, and T. Matsumoto, "Who left the door open? investigating the causes of exposed iot devices in an academic network," in *2024 IEEE Symposium on Security and Privacy (SP)*, 2024, pp. 2291–2309.

[68] H. Heo and S. Shin, "Who is knocking on the telnet port: A large-scale empirical study of network scanning," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ser. ASIACCS '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 625–636. [Online]. Available: https://doi.org/10.1145/3196494.3196537

[69] L. Metongnon and R. Sadre, "Beyond telnet: Prevalence of iot protocols in telescope and honeypot measurements," in *Proceedings of the 2018 Workshop on Traffic Measurements for Cybersecurity*, ser. WTMC '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 21–26. [Online]. Available: https://doi.org/10.1145/3229598.3229604

[70] "[ms-nlmp]: Nt lan manager (ntlm) authentication protocol," https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-nlmp/b38c36ed-2804-4868-a9ff-8dd3182128e4, aug 2022, (Accessed on 10/14/2024).

[71] "nuclei-templates/network/detection/rdp-detect.yaml · projectdiscovery/nuclei-templates," https://github.com/projectdiscovery/nuclei-templates/blob/d6003d408f5d1dd71ca16d1baf8f5fd00e332e58/network/detection/rdp-detect.yaml, (Accessed on 09/29/2024).

[72] "Cve-2019-0708," https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-0708, may 2015, (Accessed on 09/27/2024).

[73] "Cve-2018-0886," https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-0886, mar 2018, (Accessed on 09/27/2024).

[74] "Cve-2012-0002," https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0002, mar 2012, (Accessed on 09/27/2024).

[75] "Cve-2005-1794," https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-1794, jun 2005, (Accessed on 09/27/2024).

[76] "Default accounts : Ipmi anonymous login enabled," https://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.103836, (Accessed on 09/29/2024).

[77] "General : Ipmi no auth access mode enabled," https://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.103837, (Accessed on 09/29/2024).

[78] "General : Ipmi null usernames allowed," https://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.103838, (Accessed on 09/29/2024).

[79] H. Moore, "A penetration tester's guide to ipmi and bmcs," https://www.rapid7.com/blog/post/2013/07/02/a-penetration-testers-guide-to-ipmi/, jul 2013, (Accessed on 09/30/2024).

[80] D. Hobbs, "Snmp… strings attached!" https://www.blackhillsinfosec.com/snmp-strings-attached/, dec 2022, (Accessed on 09/30/2024).

[81] D. Heiland, "Snmp data harvesting during penetration testing," https://www.rapid7.com/blog/post/2016/05/05/snmp-data-harvesting-during-penetration-testing/, may 2016, (Accessed on 10/01/2024).

[82] T. Albakour, O. Gasser, R. Beverly, and G. Smaragdakis, "Third time's not a charm: exploiting snmpv3 for router fingerprinting," in *Proceedings of the 21st ACM Internet Measurement Conference*, ser. IMC '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 150–164. [Online]. Available: https://doi.org/10.1145/3487552.3487848

[83] S. Thomas, "Brute forcing snmpv3 authentication," https://applied-risk.com/resources/brute-forcing-snmpv3-authentication, jun 2020, (Accessed on 10/02/2024).

[84] "Shawndevans/smbmap: Smbmap is a handy smb enumeration tool," https://github.com/ShawnDEvans/smbmap, (Accessed on 10/04/2024).

[85] "Eternalblue exploit: What it is and how it works | sentinelone," https://www.sentinelone.com/blog/eternalblue-nsa-developed-exploit-just-wont-die/, (Accessed on 10/04/2024).

[86] N. Abulhul, "Privilege escalation with mysql user defined functions," https://medium.com/r3d-buck3t/privilege-escalation-with-mysql-user-defined-functions-996ef7d5ceaf, oct 2021, (Accessed on 10/14/2024).

[87] C. Cimpanu, "Hacker ransoms 23k mongodb databases and threatens to contact gdpr authorities," https://www.zdnet.com/article/hacker-ransoms-23k-mongodb-databases-and-threatens-to-contact-gdpr-authorities/, jul 2020, (Accessed on 10/11/2024).

[88] J. Abley, Ólafur Guðmundsson, M. Majkowski, and E. Hunt, "Providing Minimal-Sized Responses to DNS Queries That Have QTYPE=ANY," RFC 8482, Jan. 2019. [Online]. Available: https://www.rfc-editor.org/info/rfc8482

[89] Cloudflare, "What is recursive dns?" https://www.cloudflare.com/learning/dns/what-is-recursive-dns/, (Accessed on 11/15/2024).

[90] A. Herzberg and H. Shulman, "Fragmentation considered poisonous, or: One-domain-to-rule-them-all.org," in *2013 IEEE Conference on Communications and Network Security (CNS)*, 2013, pp. 224–232.

[91] K. Man, Z. Qian, Z. Wang, X. Zheng, Y. Huang, and H. Duan, "Dns cache poisoning attack reloaded: Revolutions with side channels," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1337–1350. [Online]. Available: https://doi.org/10.1145/3372297.3417280

[92] X. Li, C. Lu, B. Liu, Q. Zhang, Z. Li, H. Duan, and Q. Li, "The maginot line: Attacking the boundary of DNS caching protection," in *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association, Aug. 2023, pp. 3153–3170. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/li-xiang

[93] "scapy/scapy/layers/inet.py · secdev/scapy," https://github.com/secdev/scapy/blob/6f0faf38597080daca367d741903a99464e32760/scapy/layers/inet.py#L821, (Accessed on 10/11/2024).

[94] Truespeed, "New router & mesh wi-fi system launched," https://www.truespeed.com/news/truespeed-launches-new-router-mesh-wi-fi-system/, (Accessed on 10/09/2024).

[95] "Linksys spnmx42ts-uk faqs - linksys support," https://support.linksys.com/kb/article/3713-en/, (Accessed on 10/09/2024).

[96] X. Zhou, Q. Deng, J. Pu, K. Man, Z. Qian, and S. V. Krishnamurthy, "Untangling the knot: Breaking access control in home wireless mesh networks," in *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, 2024.

[97] "Cloud customer successes | oracle," https://www.oracle.com/customers/, (Accessed on 10/10/2024).

[98] "Cve-2025-0650," https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-0650, jan 2025, (Accessed on 04/01/2024).

[99] "Centos iptables open port 53 - server fault," https://serverfault.com/questions/508661/centos-iptables-open-port-53/508682#508682, (Accessed on 11/10/2024).

[100] "ubuntu - what are the iptables rules to permit ntp? - super user," https://superuser.com/questions/141772/what-are-the-iptables-rules-to-permit-ntp/141795#141795, (Accessed on 10/10/2024).

[101] "iis - can i use iptables on my varnish server to forward https traffic to a specific server? - server fault," https://serverfault.com/questions/442708/can-i-use-iptables-on-my-varnish-server-to-forward-https-traffic-to-a-specific-s/442734#442734, (Accessed on 10/10/2024).

[102] D. Maez, "Ip address allocation by country," [Online; accessed 2025-04-10]. [Online]. Available: https://impliedchaos.github.io/ip-alloc/

[103] "Windtre semplifica connessioni, energia, assicurazioni," [Online; accessed 2025-04-10]. [Online]. Available: https://www.windtre.it/

Appendix A. Misconfiguration Distributions

A.1. Geographical Distribution

The geographical distribution of affected hosts is illustrated in 3.

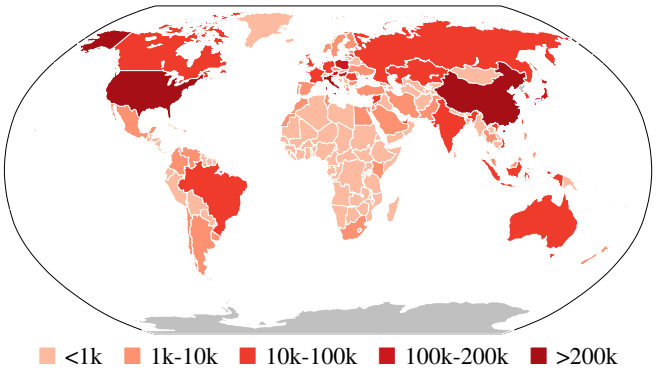


Figure 3: Affected Host Distribution

As stated in Table 3, Italy has the most affected hosts, more than the United States and China, the two countries with the most IP address allocations [102]. The reason is that AS1267 alone, which belongs to an Italian telecommunications company [103], accounted for 231,316 affected hosts, as shown in Table 4.

We list the service distribution in AS1267 in Table 21. Please note that one host may have more than one affected service. We suspect that the affected hosts are mainly the servers and network devices of this company.

Service	Count	% (of Hosts)	Note
NTP	225,654	97.55%	None allows monlist.
SNMPv3	4,062	1.76%	3,804 have the engine enterprise ID of Nokia (6527); 241 have that of Cisco (9).
SNMPv1	3,906	1.69%	3,750 run TiMOS by Nokia.
SNMPv2	3,889	1.68%	3,749 run TiMOS by Nokia.
DNS	1,833	0.79%	159 allow ANY queries.
HTTPS	1,358	0.59%	1,354 have the common name `dsldevice.lan` in their server certificates.
Others	91	0.04%	

TABLE 21: Service Distribution in AS1267

A.2. Organizational Distribution

We highlight some affected ASes belonging to well-known enterprises in Table 22. This demonstrates the high susceptibility of firewall misconfigurations of this kind. The data also indicates that UDP services are particularly vulnerable, probably due to the connectionless nature of UDP.

ASN	AS Name	Count	Main Services
714	APPLE-ENGINEERING	8,348	DNS, NTP
14593	SPACEX-STARLINK	5,571	SNMPv3, NTP
36692	CISCO-UMBRELLA	4,914	DNS, SNMPv3
45090	TENCENT-NET-AP	3,770	NTP, SSH, DNS
15169	GOOGLE	2,127	HTTPS, HTTP
13238	YANDEX	1,841	NTP, SNMPv3
132892	CLOUDFLARE	857	SNMPv3, NTP
24429	Taobao	567	NTP, SNMPv3
202623	CLOUDFLARENET-CORE	265	NTP, SNMPv3
13335	CLOUDFLARENET	131	NTP, SNMPv3
6185	APPLE-AUSTIN	130	DNS
44534	yandex-office	117	NTP, SNMPv3
19527	GOOGLE-2	60	HTTPS
109	CISCOSYSTEMS	46	IPMI
1216	ORACLE-OCI-IAD1	31	SNMPv3
394161	TESLA	25	SNMPv3, NTP
36040	YOUTUBE	14	SNMPv3, NTP

TABLE 22: Noteworthy Affected ASes

Appendix B.
Meta-Review

The following meta-review was prepared by the program committee for the 2025 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

B.1. Summary

This work measures a firewall misconfiguration where firewall rules allow traffic in either direction on a default server port, allowing external actors to scan the network using the server port as the source port. The authors scan for these firewall rules using multiple vantage points across different countries, finding this misconfiguration common in the wild.

B.2. Scientific Contributions

- Identifies an Impactful Vulnerability
- Provides a Valuable Step Forward in an Established Field

B.3. Reasons for Acceptance

1) This paper identifies an impactful vulnerability. The authors demonstrate that in practice, many firewalls that have rules to allow traffic to a server port will also unintentionally allow traffic from a source port set to the server port. Thus despite the firewall rules, external actors can still scan a network. The authors demonstrate that this misconfiguration occurs often across the Internet.
2) This paper identifies a valuable step forward in an established field. This work provides guidance on improving firewall deployments, and also a technique that may be helpful for Internet scanning, both well-established fields.

B.4. Noteworthy Concerns

1) Broader evaluations of more source ports likely would have observed a wider set of accessible services.