A Horizontal Study on the Mixed IPID Assignment Vulnerability in the Linux Ecosystem

Ao Wang*, Xuewei Feng†, Qi Li†, Yuxiang Yang†, Ke Xu†
Southeast University* Tsinghua University†
Email: wangao@seu.edu.cn; fengxw06@126.com; yangyx22@mails.tsinghua.edu.cn; {qli01, xuke}@tsinghua.edu.cn

2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS) | 979-8-3503-5012-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/IWQoS61813.2024.10682845

***Abstract*—The off-path TCP hijacking attack poses a significant threat to Internet security, allowing attackers to manipulate various upper-layer applications and causing severe real-world damage. In this paper, we undertake a horizontal study on a critical TCP hijacking attack affecting Linux servers, which was reported in November 2020 (CVE-2020-36516). This attack has the potential to compromise over 20% of popular websites on the Internet. Our study particularly focuses on determining the extent to which the developed stack patches, designed to address this vulnerability, have been effectively deployed in the real world and whether they have successfully mitigated the identified attack. In our horizontal study, we thoroughly examine the current status of the vulnerability, covering upstream and downstream components of the Linux ecosystem. This study encompasses 12 mainstream Linux distributions, 296 images from 7 leading cloud vendors, 2.92 million IPs from 301 network segments belonging to 6 major CDN vendors, as well as the top 1 million websites from 3 datasets. Our study unveils a notable disparity in the patching of the vulnerability in the Linux ecosystem, spanning various ISPs and vendors, which leaves the vulnerability open to potential exploitation and poses a serious threat to the Internet.**

***Index Terms*—TCP Hijacking, Linux Vulnerability, Stack Patches, Horizontal Study**

I. Introduction

The development of TCP/IP has been accompanied by a variety of security vulnerabilities that cause various attacks such as information inference [1], [2], connection termination [3], and session hijacking [4]–[7]. Among these attacks, the TCP hijacking attack is considered particularly threatening, which allows attackers to manipulate upper-layer applications that rely on TCP. There are many real-world attacks constructed based on TCP hijacking, e.g., SSH connections terminating [6] and web traffic tampering [4]–[8].

However, the successful execution of a TCP hijacking attack typically requires specific prerequisites, including the knowledge of port numbers, sequence numbers, and acknowledgment numbers, which are not easily obtainable for off-path attackers. Previous attacks often rely on specific network environments [9] or malware [4], [5], [8] to assist in obtaining these prerequisites. For example, some attacks require both the attacker and the victim to connect to the same Wi-Fi network [9], [10], while some other attacks require the victim's running of unprivileged malware [4], [5]. Consequently, while the TCP hijacking attack can pose a significant threat to Internet security, its practical feasibility is markedly restricted.

In 2016 and 2020, Cao et al. and Feng et al. made significant discoveries regarding TCP hijacking attacks [6], [7]. Cao et al. uncovered the "challenge ACK rate limit vulnerability", while Feng et al. uncovered the "mixed IPID assignment vulnerability". These are two severe vulnerabilities, as they enable attackers to manipulate TCP sessions completely off-path, without any additional assistance, bringing TCP hijacking attacks to a more practical level.

After the disclosure of the challenge ACK rate limit vulnerability, a subsequent 6-month longitudinal study was carried out on the Alexa top 1 million websites to track their patching behavior [11]. This study provided a valuable data point for studying the patching of webservers running the Linux TCP stack. However, this study focuses on the patching behavior of webservers within a 6-month window, rather than the stable patching status, which could have continued to evolve beyond that timeframe. Moreover, the study did not offer a horizontal view across the Linux ecosystem, such as upstream distributions, cloud and Content Delivery Network (CDN) services, which are widely adopted in practice. Therefore, the study lacks a comprehensive understanding of the patching status concerning TCP stack vulnerabilities in the Linux ecosystem.

In this paper, we thoroughly investigate the impacts of the critical mixed IPID assignment vulnerability uncovered 4 years ago [7], particularly the propagation of the developed patches in the Linux ecosystem. In contrast to the study on the challenge ACK rate limit vulnerability, which primarily concentrated on webservers' patching behavior, we aim at conducting a horizontal study within the Linux ecosystem. Specifically, we delve into the upstream Linux kernel and distributions, the midstream cloud and CDN, as well as the downstream websites, i.e., a comprehensive study along the patch propagation path. The goal of our study is to offer an in-depth perspective on the patching status concerning critical stack vulnerabilities in the entire Linux ecosystem and serve as an alarm for the located vulnerable links.

Furthermore, unlike the challenge ACK rate limit vulnerability that solely exploits rate limit in the challenge ACK mechanism, the mixed IPID assignment vulnerability we studied represents a more comprehensive exploitation of

This work was supported in part by the Science Fund for Creative Research Groups of the National Natural Science Foundation of China under No. 62221003, the Key Program of the National Natural Science Foundation of China under No. 61932016 and No. 62132011, the National Science Foundation for Distinguished Young Scholars of China under No. 62425201.

both the challenge ACK mechanism and the fundamental Linux IPID assignment strategy. We delve into the essential components employed in performing the TCP hijacking attack based on the mixed IPID assignment vulnerability, including the implementation status of RFC 5961 and IPID assignment strategy for different types of packets. For the Linux kernel and distributions, we conduct tests on a local testbed to identify their patching time and vulnerability exposure time. For cloud and CDN, we test a wide range of Virtual Private Servers (VPS) and CDN servers from well-known vendors. For websites, we investigate the vulnerability and patching status of top 1 million websites from three datasets.

After a thorough study spanning upstream to downstream of the Linux ecosystem (i.e., covering 12 distributions, 296 images from 7 cloud vendors, and 2.92 million IPs from 301 network segments belonging to 6 CDN vendors), we reveal significant disparities and delay in the patching of the vulnerability among various network scenarios and different vendors, which offer an in-depth perspective on the patching behavior of TCP stack vulnerabilities in the Linux ecosystem. The study aims to incentivize the community, vendors and service providers to expeditiously patch and update their kernel implementations and ancillary products, which is instrumental for mitigating prospective protocol stack vulnerabilities and fostering a more robust Linux ecosystem.

Contributions. Our main contributions are as follows:

- Our horizontal study reveals that the critical mixed IPID assignment vulnerability within the Linux ecosystem (originating from the kernel and extending to various Linux distributions) experiences substantial patching delays, which can extend up to two years. This substantial delay exposes a vulnerable window for potential attackers to launch the severe off-path TCP hijacking attack.
- We uncover that widely used real-world infrastructure in Linux ecosystem, such as cloud services, remains significantly impacted by the vulnerability, allowing off-path attackers to manipulate various applications on the Internet. Furthermore, we observe that the patch deployment status enforced by Internet ISPs and various vendors is notably slow and lacks consistency.
- We identify two vulnerabilities concerning the handling of ICMP error massages. The first vulnerability affects three widely-used cloud platforms and could result in session failures. The second vulnerability, on the other hand, enables potential attackers to bypass checks on carefully crafted ICMP error massages.

II. Background

In this section, we provide a concise background of the mixed IPID assignment vulnerability and the patches developed by the Linux community to mitigate the vulnerability.

A. Blind in Window Attacks and RFC 5961

To deal with brute-force guess attacks from blind (off-path) attackers [12], RFC 5961 recommends a more strict check on sequence and acknowledge numbers for incoming packets, known as the challenge ACK mechanism. Specifically, for:

- **Spoofed SYN**: If an attacker tries to interrupt the target connection with a spoofed `SYN` packet, a challenge `ACK` packet will be sent and an `RST` packet with correct sequence number is expected.
- **Spoofed RST**: If an attacker tries to interrupt the target connection with a spoofed `RST` packet, even if the guessed sequence number falls within the receive window, as long as it is not exactly equal to the next expected sequence number (RCV.NXT), a challenge `ACK` packet will be sent and an `RST` packet with correct sequence number is expected.
- **Spoofed Data**: If an attacker tries to insert data packet into the target connection, even if the guessed sequence number falls within the receive window and the guessed acknowledge number falls within the ACK window, as long as the acknowledge number does not fall within a smaller challenge ACK window, a challenge `ACK` packet will be sent and a data packet with correct acknowledge number is expected.

In earlier Linux implementations, the challenge `ACK` packet was assigned a global rate limit for performance considerations, which could be used by attackers to infer secrets as a side channel and ultimately hijack TCP sessions [6].

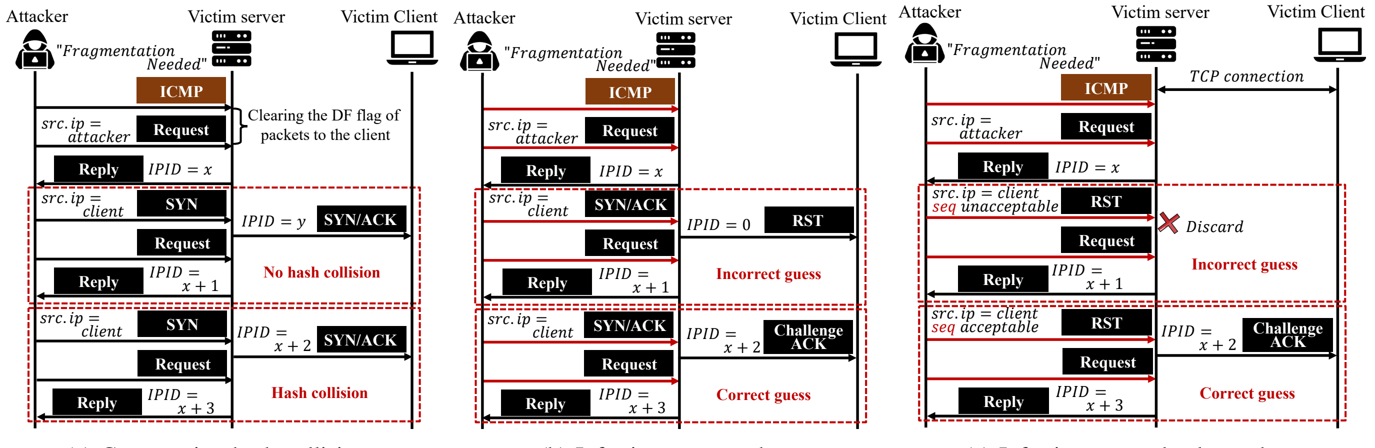
B. IPID Assignment Strategy in Linux

IPID (IP Identification) is a 16-bit field in the IP header that is used for packet fragmentation and reassembly. Linux currently employs two strategies for IPID assignment: per-socket and hash-based. The per-socket strategy maintains an increasing counter as IPID while the hash-based strategy selects IPID from 2048 hash counters maintained by the kernel. The selection of the hash counter depends on the source IP, destination IP, protocol number, and a random number generated during system boot. Destinations with the same hash value share the same hash counter.

$$hash(Src\_IP, Dest\_IP, Protocol\_num, Boot\_random) \quad (1)$$

Linux kernel employs a mixed IPID assignment strategy for TCP sockets before 5.16, incorporating both hash-based strategy and per-socket strategy. Feng et al. uncovered that when the `DF` (Don't Fragment) flag of TCP packets can be set to `False` (by default to `True`), the IPID assignment for the packets will be downgraded from per-socket strategy to hash-based strategy [7]. Additionally, they discovered that the `DF` flag of TCP packets can be cleared by triggering fragmentation through a forged ICMP error message ("Fragmentation Needed", type=3, code=4) embedded with ICMP echo reply, which can deceive the originator's check.

Specially, `RST` and `SYN/ACK` packets are processed independently from other TCP socket packets in the Linux kernel and adopt a separate IPID assignment strategy. `SYN/ACK` packets employ a mixed IPID assignment strategy that can be downgraded from 0 IPID to hash-based IPID. `RST` packets employ only 0 IPID from kernel version 4.18 [2].



(a) Constructing hash collisions (b) Inferring port numbers (c) Inferring seq and ack numbers

Fig. 1: The attack process of the mixed IPID assignment vulnerability

C. Off-path Attacks Utilizing Mixed IPID as a Side-channel

Based on the aforementioned challenge ACK mechanism and the IPID assignment strategy, an off-path TCP hijacking attack can be constructed. Firstly, the attacker downgrades the IPID assignment strategy of the victim server to the hash-based IPID assignment strategy using a carefully constructed ICMP "Fragmentation Needed" message. Then, through hash collision, the attacker identifies a hash counter shared by the attacker and the victim client. This hash counter serves as a side channel, enabling the attacker to determine whether the guessed value is right or fall within the correct range. Fig.1 illustrates the detailed process of exploiting the mixed IPID assignment vulnerability when fragmentation is triggered:

- **Constructing Hash Collisions**: As shown in Fig.1a, the attacker sends request to the victim server with its own IP and sends spoofed `SYN` packet to the victim server with the victim client's IP. The victim server will reply a `SYN/ACK` packet to the victim client with an IPID selected from one of the 2048 hash counters. The shared hash counter is found once the attacker observes an IPID jump change (more than 1) in the replies. If not found, the attacker changes its IP address and repeats the process.
- **Inferring Port Numbers**: As shown in Fig.1b, the attacker sends request to the victim server with its own IP and sends spoofed `SYN/ACK` packet to the victim server with the victim client's IP. The victim server will return a `RST` packet with 0 IPID if the target connection does not exist [13]. Otherwise, the victim server will return a challenge `ACK`, which shares the same hash counter with replies. The attacker can determine whether a challenge `ACK` is triggered by observing the IPID in replies.
- **Inferring Sequence and Acknowledge Numbers**: As shown in Fig.1c, the attacker sends request to the victim server with its own IP and sends spoofed `RST` (or `ACK`) packet to the victim server with the victim client's IP. Similar to the process of guessing the port number, only the packet within the receive window (challenge ACK window) can trigger a challenge `ACK`, and then a jump change of IPID in replies can be observed.

We refer readers to the original paper [7] for more details.

D. Patches for the Mixed IPID Assignment Vulnerability

Two patches have been implemented in the Linux kernel to deal with the vulnerability.

- **Patch 1: Clearing the IPID of SYN/ACK**: Starting from kernel version 5.16.5, the Linux kernel applies a patch that sets `SYN/ACK` packets' IPID to 0 and `DF` flag to 1 if it is smaller than IPV4_MIN_MTU, otherwise, a random IPID is used [14].
- **Patch 2: Assigning IPID Based on the Protocol Field**: The mixed IPID assignment vulnerability comes from the side channel formed by the shared hash counter. To completely eliminate this side channel, kernel version 5.16.5 and beyond adopt only per-socket IPID assignment strategy for TCP sockets [15].

III. Measurement Method

In this section, we present the target object, basic idea, and measurement method.

A. Target Object: Linux Ecosystem

Linux systems are widely used to provide various Internet services, and Linux is the world's most used server operating system [16]. The deployment of Linux patches on actual servers in use involves a propagation chain that encompasses the entire Linux ecosystem. First, the Linux kernel releases patches for various vulnerabilities. Then, distributions integrate these patches into their own operating systems according to their release cycles and policies. Next, ISPs update their images with the latest patched versions of the distributions. Finally, downstream websites access services from ISPs. Our study aims to conduct a horizontal study of the whole Linux ecosystem and evaluate the patching status at each stage of the propagation chain.

B. Basic Idea

In order to assess the security status of the targets, we employ two metrics: **vulnerability rate** and **patch rate**. The former denotes the proportion of samples that are vulnerable to the attack, whereas the latter indicates the proportion of samples that have been patched against the vulnerability. We note that some samples are neither vulnerable nor patched, since they are either non-Linux systems or Linux systems with kernel versions below 4.18, which are immune to the vulnerability. We exclude these samples from the calculation of the patch rate.

We examine the adherence of RFC 5961 on the target server (Section III-C) and its IPID allocation policy (Section III-D) to differentiate among vulnerable, patched, and otherwise unaffected machines. All tests are performed on self-built connections, through which we transmit customized requests and analyze the responses. Our self-built connections enable us to construct packets that satisfy specific requirements (e.g., with in-window sequence numbers or acknowledgment numbers). The potential biases exist in the method are discussed in Section VII-A and Section VII-B.

C. RFC 5961 Test

We commence by examining the target server's compliance of RFC 5961, which encompasses blind SYN test, blind RST test, and blind data test, as depicted in Fig. 2. All vulnerable and patched machines ought to conform to RFC 5961.

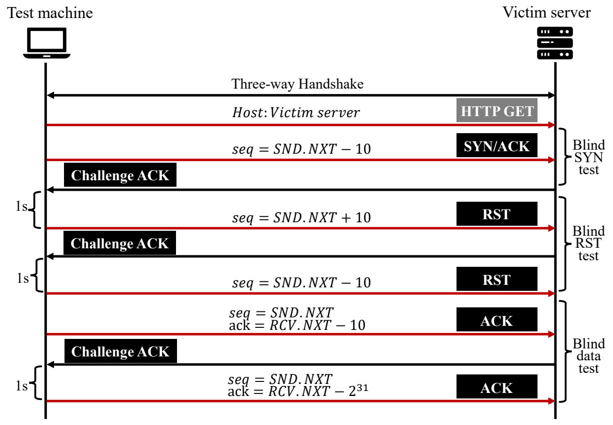


Fig. 2: RFC 5961 test

Blind SYN Test. Here we test whether a challenge `ACK` packet can be triggered by a `SYN/ACK` packet, which is exploited by off-path attackers to verify whether the guessed port number is correct. In the test, a `SYN/ACK` packet is sent, and a challenge `ACK` packet is expected as a response. Given that some servers have only implemented the challenge ACK mechanism for in-window `SYN/ACK` packets [17], our actual test assigns the sequence number of `SYN/ACK` packets to SND.NXT - 10 to ensure that `SYN/ACK` with out-of-window sequence numbers can also provoke challenge `ACK` packets.

Blind RST Test. Subsequently, we verify whether `RST` packets with in-window sequence numbers can provoke challenge `ACK` packets, while `RST` packets with out-of-window sequence numbers cannot, which is exploited by off-path attackers to determine whether the guessed sequence number is within the receive window. We send two `RST` packets: one with an in-window sequence number and one with an out-of-window sequence number. We expect only the former packet to provoke a challenge `ACK` packet and the latter to get no response.

Blind Data Test. Lastly, we examine whether data packets with acknowledge number in the challenge ACK window can elicit challenge `ACK` packets, while data packets with acknowledge number out of the challenge ACK window cannot, which is leveraged by off-path attackers to test whether the guessed acknowledge number is within the challenge ACK window. We send a data packet in the challenge ACK window expecting a challenge `ACK` packet as reply, and one data packet out of the challenge ACK window expecting no response.

Two additional optimizations are introduced to address problems encountered in the test. Firstly, we insert a 1s delay after finalizing the 3-way handshake prior to the test. This delay accommodates some servers, such as some Akamai servers, that tend to discard packets that arrive shortly after the handshake. Second, some servers will terminate the connection immediately upon receiving an unexpected request (TCP deferred accept), so we send an `HTTP GET` request to alter the connection's state to `Established` after the handshake.

D. IPID Assignment Strategy Test

Here we present the measurement method for the IPID assignment strategy of the target server. Three tests have been conducted to identify the IPID assignment policy for `RST` packets, `SYN/ACK` packets, and TCP socket packets, respectively, as illustrated in Fig.3.

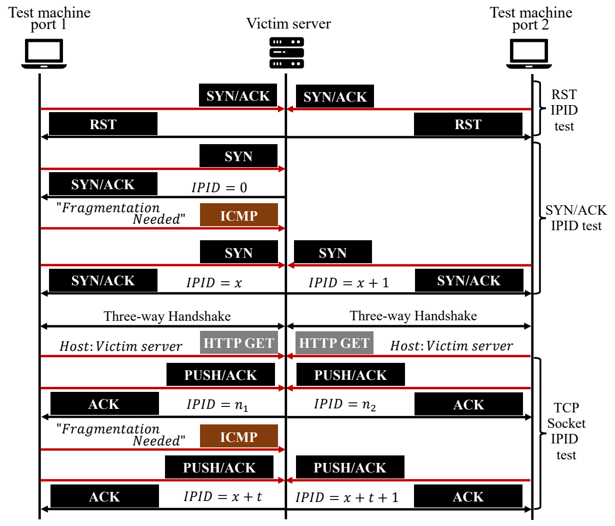


Fig. 3: IPID assignment strategy test

RST IPID Test. Only when the 0 IPID assignment strategy is employed for `RST` packets, can attackers distinguish them from challenge `ACK` packets that employ the hash-based IPID assignment strategy. Ideally, we can send a `SYN/ACK` packet to an open port with no connection to the target server and check whether the returned `RST` packet has 0 IPID. However, we observe that many webservers do not adhere this Linux kernel modification [18] and assign a constant IPID to `RST` packets in practice. Therefore, we send two `SYN/ACK` packets to the target server and compare IPIDs of the two received

RST packets. If the IPIDs are identical, the server employs a constant IPID for RST, which is equivalent to 0 IPID.

SYN/ACK IPID Test. The IPID assignment strategy of SYN/ACK packets may be exploited to determine whether a shared hash counter is used. We send a SYN packet to the target server and check the IPID of the returned SYN/ACK packet. If it is 0, an ICMP "Fragmentation Needed" message is sent to attempt to downgrade the target server's IPID assignment strategy. Subsequently, two SYN packets are sent to the target server and IPIDs of the returned SYN/ACK packets are recorded. If the IPIDs are not 0 or random number, the IPID assignment strategy has been successfully downgraded. If fragmentation cannot be triggered (i.e., DF = 1) and the IPID of the SYN/ACK packet is still 0, Patch 1 has been applied.

TCP Socket IPID Test. The final crux of the attack is that the IPID assignment strategy of TCP sockets can be downgraded from per-socket IPID to hash-based IPID. We establish two TCP connections with the target server and transmit a PUSH/ACK packet through each connection in a short time interval. The target server should return two ACK packets. If the IPIDs of the two returned packets are non-sequential, the target server employs the per-socket IPID assignment strategy for TCP sockets. Then, we trigger fragmentation on the target server, dispatch another PUSH/ACK packet through each connection, and compare the IPID of the two returned ACK packets. If the IPIDs are sequential, the IPID assignment strategy has been successfully downgraded. Otherwise, Patch 2 has been applied.

Other OSes and middleboxes may influence the results of the measurement. In Section VII-A and Section VII-B, we discuss the potential measurement bias that may be introduced by other OSes and middleboxes and the respective mitigation.

E. Implementation

All tests are conducted on a local testbed for kernel and distributions, with a test machine loaded with Ubuntu 20.04, and a tested machine loaded with OSes under test. Both machines are equipped with an Intel Xeon E3-1230 V2 CPU (4 cores, 8 threads), 8G memory, and 5Mbps bandwidth. For cloud, CDN, and webservers that deployed worldwide, a local test is unachievable. To minimize the impact of filtering strategies of different ISPs [19] and mitigate possible packet loss, we deploy three vantage points in Frankfurt, Hongkong, and Toronto and repeat each test three times.

Ethical Considerations: All measurements are performed on our own TCP connections, without affecting any other network traffic. We do not launch actual attacks by sending a large number of probe packets, but mainly examine the features required for a successful attack with trivial traffic load (about 0.4kbps) on the tested targets.

IV. Kernel and Distribution

Being the foundation of the Linux ecosystem, vulnerability status of the kernel and distributions directly affects the security of upper-layer services. In this section, we reveal substantial patching delays existing in kernel and distributions.

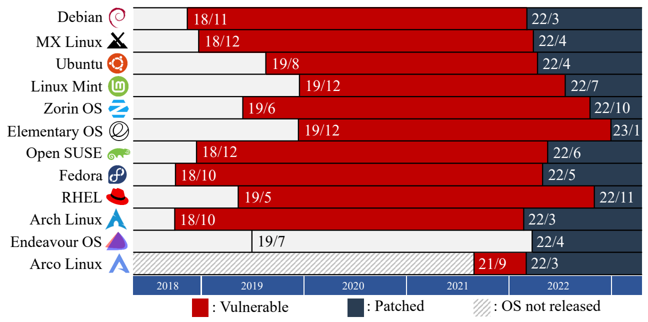


Fig. 4: Timeline for Linux distributions patch

A. Linux Kernel Patch

Feng et al. presented their findings on the mixed IPID assignment vulnerability during the ACM Conference on Computer and Communications Security (CCS'20) in November 2020 [7]. In response to this discovery, the Linux community underwent subsequent development in February 2022 to address the issue, resulting in the implementation of two patches.

The first patch addresses the vulnerability by clearing or randomizing the IPID of SYN/ACK packets. Commencing with kernel version 5.16.5, the Linux kernel incorporates a modification that sets the IPID of SYN/ACK packets to 0 and the DF flag to 1 if it is smaller than IPV4_MIN_MTU, otherwise, random IPID will be used. Consequently, this prevents attackers from constructing hash collisions, thereby thwarting their ability to identify a victim client. The second patch, also applicable to Linux kernel version 5.16.5 and beyond, involves assigning IPIDs for TCP packets based on the protocol field. Specifically, this strategy adopts the per-socket IPID assignment method for TCP packets without taking into account the DF flag of the packets. As a result, off-path attackers will be unable to manipulate the IPID assignment methods of the target Linux server by crafting an ICMP error message to clear the DF flag of the server's TCP packets.

B. Linux Distributions Patch

Subsequent to the Linux kernel patch, various Linux distributions adopted patches for the mixed IPID assignment vulnerability. We test different versions of 12 Linux distributions in a local testbed to assess their patching latency and vulnerability exposure time, depicted in Fig.4. As of January 2023, all examined distributions were updated with the patches.

The upstream-downstream relationship determines the patching sequence of Linux distributions, with the upstream ones patching earlier. Among the tested distributions, Debian, Open SUSE, Fedora, and Arch Linux are the direct downstream of the Linux kernel. They exhibit the shortest patching latency, within 4 months after the kernel patch. MX Linux and Ubuntu are downstream of Debian, which patched their kernel about 5 months after the Linux kernel patch. Linux Mint, Zorin OS, and Elementary OS are downstream of Ubuntu, and they patched later. Elementary OS patched even after a year since the original patch. The discrepancy in patching latency between upstream and downstream distributions creates an exploitable attack window for attackers.

Regarding the vulnerability exposure time, the Linux kernel cleared `RST` packets' IPID in September 2018 and patched the mixed IPID assignment vulnerability in February 2022, exposing the vulnerability for a period of 3 years and 5 months. This period may vary in different distributions. Among them, Fedora has the longest exposure time of 3 years and 7 months, while Linux Mint has the shortest exposure time of 2 years and 7 months.

None of the versions of Endeavor OS are susceptible to the vulnerability, as its firewall proactively intercepts `SYN/ACK` and in-window data packets and thereby precludes the response of corresponding challenge `ACK` packets.

C. Summary of Findings for Kernel and Distributions

In this section, we study the patching status of the Linux kernel and 12 distributions. We discover that despite being patched, the kernel and distributions suffer from a substantial patching latency. The kernel was patched more than a year after the vulnerability was disclosed, while distributions add an additional delay of 4 months to 1 year on top of the kernel patch, which results in a total delay of up to 2 years, leaving an exploitable window for potential attackers.

V. Cloud and CDN

Open port services are widely deployed on cloud and CDN platforms. Cloud and CDN vendors typically maintain some system images themselves, allowing users to deploy services on them. The security of these images is crucial for the security of the deployed services. We uncover the disparity in the patching status of different ISPs and vendors.

A. Targets

We test 296 images from seven cloud vendors: Google Cloud, Amazon EC2, Microsoft Azure, Alibaba Cloud, Tencent Cloud, Huawei Cloud, and Tianyi Cloud. We purchase VPSs with special images deployed directly from these vendors. Except for Azure, we evaluate all official Linux-based images from these cloud vendors. For Azure, as it only offers third-party images, we evaluate 30 popular third-party images.

We study 6 major CDN vendors, including Akamai, Cloudflare, CloudFront, Azure CDN, Fastly, and Imperva. Previous work maps the top 10k IPs to the corresponding CDN corporations using WHOIS [6], which has two limitations: Firstly, the tested samples are limited to the top 10k IPs, which might not fully represent the CDN servers. Secondly, CDN vendors often optimize content delivery by utilizing their partners' IPs, making it challenging to accurately identify CDN vendors through WHOIS. To overcome these challenges and ensure a more comprehensive evaluation, we leverage the publicly available whitelists for our study, which CDN vendors offer to prevent webservers from incorrectly filtering traffic from CDN servers [20].

B. Cloud Vulnerability

We present the investigation results of clouds in Fig.5. As of July 2023, Alibaba Cloud, AWS EC2, and Tencent Cloud

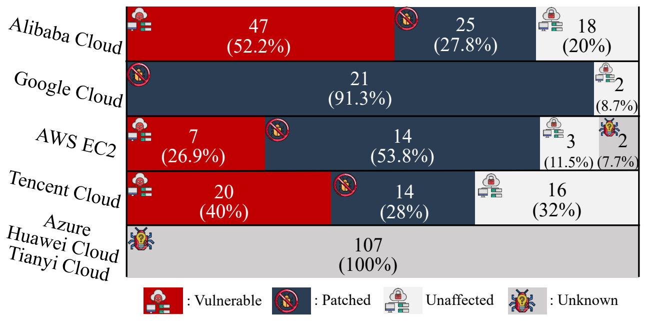


Fig. 5: Vulnerability and patching status of 7 cloud vendors

still offer vulnerable images. Among them, Alibaba Cloud exhibits the highest vulnerability rate (approximately 52.2%), whereas all Google Cloud images are not affected by the vulnerability. There is also a considerable discrepancy in the patch rate among different vendors. When considering only vulnerable and patched images, the patch rate of Google Cloud images attains 100%, while the patch rate of Tencent Cloud and Alibaba Cloud are below 50%. Furthermore, each cloud vendor supplies a number of unaffected images that adopt Linux kernel versions prior to 4.18 that are neither vulnerable nor patched, which are excluded when computing the patching rate. As an exceptional case, we discovered that RHEL 7 and SUSE 15 SP4 provided by AWS EC2 cannot be fragmented appropriately, which hindered us from assessing their vulnerability and patching status. Owing to space constraints, we only display the detailed results of Alibaba Cloud in Table II.

TABLE I: Categorized vulnerable status for 4 cloud vendors

	Alibaba Cloud	Google Cloud	AWS EC2	Tencent Cloud
No patch provided	11/47	0/0	3/7	12/20
Patch provided	36/47	0/0	4/7	8/20

We attempt to understand why there is such a significant disparity in the vulnerability rate and patch rate among different cloud vendors. As depicted in Table I, we classify the vulnerable images into two groups: No patch provided and Patch provided. The "No patch provided" denotes that the image is vulnerable since the distribution does not offer any patched images, such as Debian 8 and RHEL 7. The "Patch provided" implies that the distribution provides patches, but the cloud vendor has not updated the images accordingly, such as Ubuntu Xenial in all cloud vendors except Google Cloud.

We find two factors account for the disparity: 1) Some cloud vendors offer images that exceed the security support period, such as Ubuntu Trusty, Fedora 33 and CentOS 8. Unless users manually apply patches, these images remain vulnerable. 2) Many images that should have been patched are not patched promptly. Among the 47 vulnerable images in Alibaba Cloud, 36 images have not been patched timely.

Middlebox Error: Gateway Translates ICMP Error Packet Incorrectly. We find no fragmentation can be triggered in images provided by Microsoft Azure, Huawei Cloud, and Tianyi Cloud. As depicted in Fig.6, by capturing packets on the VPS side, we find that the source IP of the ICMP

TABLE II: Measurement results for Alibaba Cloud

Distribution(Number)	**Version(Minor version)**	**Kernel base**	**IP**	**Location**	**RFC 5961?**	**0 IPID for RST?**	**Fragment triggered?**	**SYN/ACK IPID downgraded?**	**Socket IPID downgraded?**
Alibaba Cloud Linux	2.1903	4.19	123.56.x.x	CN	●	●	●	○	○
Alibaba Cloud Linux	3.2104	5.10	123.56.x.x	CN	●	●	●	○	○
Debian(2)	Jessie(9/11)	3.16	39.106.x.x/39.107.x.x	CN	●	○	●	●	●
Debian(5)	Stretch(6/8/9/11/12)	4.9	8.209.x.x	JP	●	●	●	●	●
Debian	Stretch(13)	4.9	123.56.x.x	CN	●	●	●	○	○
Debian(10)	Buster(2-11)	4.19	47.91.x.x	AE	●	●	●	●	●
Debian(2)	Buster(12/13)	4.19	47.91.x.x	AE	●	●	●	○	○
Debian(3)	Bullseye(0-2)	5.10	8.208.x.x	GB	●	●	●	●	●
Debian(4)	Bullseye(3-6)	5.10	47.254.x.x	DE	●	●	●	○	○
Ubuntu Server	Trusty	4.4	47.95.x.x	CN	●	○	●	●	●
Ubuntu Server	Xenial	4.4	39.106.x.x	CN	●	○	●	●	●
Ubuntu Server	Bionic	4.15	39.105.x.x	CN	●	●	●	○	○
Ubuntu Server	Focal	5.4	39.106.x.x	CN	●	●	●	○	○
Ubuntu Server	Jammy	5.15	182.92.x.x	CN	●	●	●	○	○
CentOS(8)	7(2-9)	3.10	8.213.x.x	KR	●	○	●	●	●
CentOS(6)	8(0-5)	4.18	8.213.x.x	KR	●	●	●	●	●
CentOS Stream	8	4.18	8.209.x.x	JP	●	●	●	○	○
CentOS Stream	9	5.14	47.74.x.x	AU	●	●	●	○	○
RHEL(2)	Maipo(8/9)	3.10	147.129.x.x/147.139.x.x	ID	●	●	●	●	●
RHEL(7)	Ootpa(0-6)	4.18	147.129.x.x/147.139.x.x	ID	●	●	●	○	○
RHEL	Plow	5.14	147.139.x.x	ID	●	●	●	○	○
Fedora	33	5.11	8.208.x.x	GB	●	●	●	●	●
Fedora	34	5.16	47.115.x.x	CN	●	●	●	○	○
Fedora	35	5.18	116.62.x.x	CN	●	●	●	○	○
Rocky Linux(2)	Green Obsidian(5/6)	4.18	47.74.x.x	AU	●	●	●	●	●
Rocky Linux	Green Obsidian(7)	4.18	47.74.x.x	AU	●	●	●	○	○
Rocky Linux	Blue Onyx(0)	5.14	8.138.x.x	TH	●	●	●	●	●
Rocky Linux	Blue Onyx(1)	5.14	8.213.x.x	TH	●	●	●	○	○
OpenSUSE	42.3	4.4	47.250.x.x	MY	●	○	○	●	●
OpenSUSE(3)	15(1-3)	4.12/5.3	147.139.x.x	IN	●	●	●	●	●
OpenSUSE	15(4)	5.14	8.219.x.x	SG	●	●	●	○	○
SUSE Enterprise	12(SP3)	4.4	47.89.x.x	US	●	○	●	●	●
SUSE Enterprise	12(SP4)	4.4	47.89.x.x	US	●	●	●	●	●
SUSE Enterprise	12(SP5)	4.12	47.252.x.x	US	●	●	●	○	○
SUSE Enterprise(2)	15(SP1/SP2)	4.12/5.3	47.252.x.x/198.11.x.x	US	●	●	●	●	●
SUSE Enterprise	15(SP3)	5.3	198.11.x.x	US	●	●	●	○	○
Anolis OS(2)	7(7/9)	3.10	121.41.x.x/8.217.x.x	CN	●	○	●	●	●
Anolis OS(2)	8(2/4)	4.18	8.130.x.x/39.104.x.x	CN	●	●	●	●	●
Anolis OS(2)	8(6/8)	5.19	47.104.x.x	CN	●	●	●	○	○
AlmaLinux(2)	8(5/6)	4.18	47.90.x.x	US	●	●	●	●	●
AlmaLinux	8(7)	4.18	47.251.x.x	US	●	●	●	○	○
AlmaLinux	9	5.14	47.251.x.x	US	●	●	●	●	●
AlmaLinux	9(1)	5.14	147.139.x.x	ID	●	●	●	○	○

●means that the feature is implemented, while ○means that the feature is not implemented.

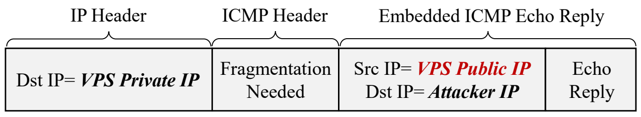


Fig. 6: ICMP error message with mistranslated padding

echo reply embedded in the received ICMP "Fragmentation Needed" message is not the private IP of the VPS, but the public IP of the VPS. That is, the cloud gateway device does not translate the padding of the ICMP error message correctly, which is required by RFC 792 [21]. This translation error results in the kernel rejecting the ICMP error message, thus preventing fragmentation from being triggered and may eventually result in network session failure. We have reported the error to affected vendors.

C. CDN Vulnerability

We obtain 301 network segments belonging to 6 CDN vendors through whitelists. Since not all IPs in these network segments are deployed with CDN services, we use the Zmap tool [22] to scan the IPs with port 80/443 open, resulting in 2.92 million IPs that are potential CDN servers. We conducted a comprehensive investigation on these IPs and summarized the results in Table III. We find that the patching status of CDN vendors is extremely consistent. CDN vendors have a high level of patch rate (over 99% of all the tested vendors) and almost none of their servers exhibited the vulnerability. Both as ISP, the patching status of CDN is far better than cloud.

TABLE III: Patching status of 6 CDN vendors

Provider	Network segment	Valid IP	Patching rate
Akamai	16	1,168,815	99.1%
Cloudflare	15	109,210	100%
CloudFront	142	1,418,291	99.9%
Azure CDN	98	19,499	99.7%
Fastly	19	125,389	100%
Imperva	11	76,606	99.9%

D. Summary of Findings for Cloud and CDN

In this section, we show the disparity in the patching of the vulnerability among intermediary ISPs. We find that while almost all CDN servers are well patched, cloud images are poorly patched, with 74 out of 296 cloud images vulnerable. Furthermore, there are discrepancies in the patching status among different vendors. More than 50% of Alibaba Cloud images exhibit mixed IPID assignment vulnerability, whereas none of Google Cloud images has this vulnerability.

VI. Websites Vulnerability

Web services often expose public ports and IPs, making them the most common targets for off-path TCP hijacking [4], [6], [7]. In this section, we demonstrate that the mixed IPID assignment vulnerability is prevalent in the top websites.

A. Targets

Since Alexa no longer provided its website ranking service since May 2022, we opt for the top 1 million websites in the Chrome User Experience Report as the dataset, which is the most accurate list according to Ruth et al. [23]. Other two datasets (Majestic [24] and Tranco [25]) are also tested for comparison. We obtain IP of target websites through domain name resolution. In case of multiple IP due to load balancing or CDN, we choose the first IP returned by the DNS resolver as the target and test duplicate IP only once. After excluding duplicate IP, active server disconnections, and test timeouts, we get 341,395 IPs belonging to 876,026 websites for Google datasets.

B. Bypass Checks on Crafted ICMP Error Packets

We find that only 55% of tested webservers correctly accepted the crafted ICMP ”Fragmentation Needed” packet and cleared the DF bits of subsequent packets. Various reasons may contribute to the fragmentation failure, such as hosts ignoring ICMP messages, the middlebox filtering, and the gateway mistranslation (Section V-B). Here we find a vulnerability which allows attackers to bypass checks on crafted ICMP “Fragmentation Needed” packet by partial firewalls. We find that sending a crafted ICMP echo request to the target server before sending crafted ICMP “Fragmentation Needed” packet can increase the success rate of triggering fragmentation. It exploits the fact that although the Linux kernel does not check whether the ICMP echo reply embedded in the ICMP “Fragmentation Needed” message originated from itself, some firewalls will do so. By inducing the target server to send an ICMP echo reply that passed through the firewall, we make the ICMP “Fragmentation Needed” packet pass some firewall’s check, since its payload is indeed a packet sent by the target server. With this trick, we improved the probability of successfully triggering fragmentation to 72.8%.

C. Basic Results

Fig.7 illustrates the measurement results of Google top 1 million websites regarding the mixed IPID assignment vulnerability. We study their patching status by a hierarchical representation of their features. The leaf nodes denote the fraction of websites that adopt a specific feature, whereas the parent nodes denote the fraction of websites that adopt both/at least one feature of their child nodes (intersection set/union set). By manipulating the DF flag, we are able to downgrade the IPID assignment strategy for TCP sockets and SYN/ACK packets for 101.3k (11.6%) and 104.9k (12.0%) websites, respectively. Among them, 93.3k (10.6%) websites are concurrently susceptible to both downgrades, implying that they do not implement any patches. Moreover, 62.3% of the websites have fully implemented the challenge ACK mechanism and 484.2k (55.3%) of the websites have cleared their RST IPID. By combining all features implemented, we identify 23.0k websites that are susceptible to the vulnerability with a vulnerability rate of 2.6%. Fig.8 displays the geographical distribution of the vulnerable websites.

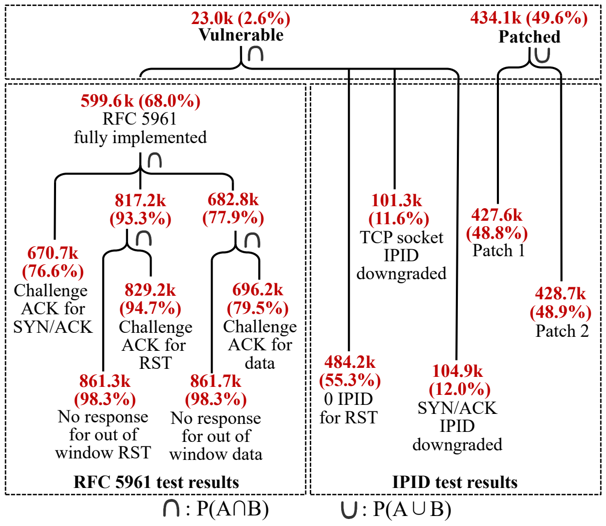


Fig. 7: Measurement results of Google top 1 million websites

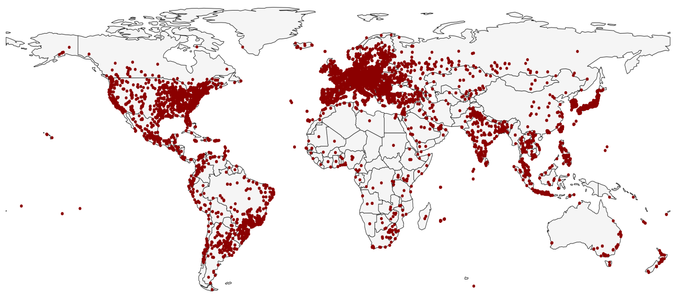


Fig. 8: Geographical distribution of vulnerable websites

Our results demonstrate that the IPID assignment strategy of SYN/ACK of 427.6k (48.8%) websites is immune to downgrade with Patch 1 applied, and the IPID assignment strategy of TCP sockets of 428.7k (48.9%) websites is immune to downgrade with Patch 2 applied. Ultimately, 434.1k (49.6%) websites have implemented at least 1 patch, with a patching rate of 95.1% ($\frac{434.1k}{22.5k+434.1k} \times 100\%$).

D. Comparison of Different Rank of Websites

Fig.9 shows the vulnerability rate and patch rate of different rank websites. For comparison, we also present the results of the Majestic and Tranco. The results of the three datasets are analogous and essentially exhibit the same tendency. Regarding vulnerability, the proportion of websites with vulnerability in the top 1k to the top 1M websites of the three datasets does not surpass 3%. Although the ratio is comparatively low, considering that there are hundreds of millions of websites on the Internet, a 3% vulnerability rate implies that millions of websites are susceptible to this vulnerability. We detect this

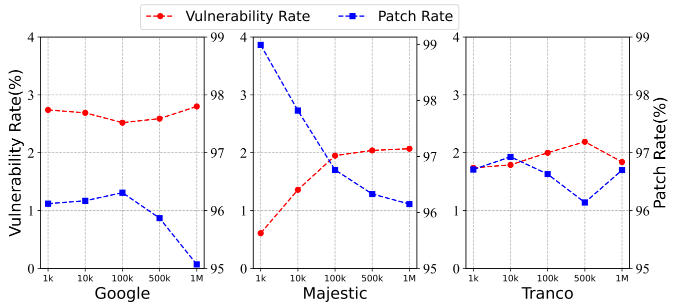


Fig. 9: Vulnerability rate and patch rate of different rank in 3 datasets

vulnerability in well-known websites, comprising but not restricted to news (www.munet.com), communication (vk.com), e-commerce (jd.com), etc. Regarding patch rate, websites with higher rank typically patch more than websites with lower rank, and the patching status deteriorates as the rank declines.

E. *Summary of Findings for Websites*

In this section, we study the top 1 million websites and discover that the vulnerability is prevalent in webservers. We identify 23k websites with vulnerable webservers in Google dataset. Moreover, we find that the vulnerability rate and patch rate are consistent across three website ranking datasets.

VII. Discussion

A. *Interference of Other OSes*

We utilize the RFC 5961 implementation and IPID assignment policy to distinguish among vulnerable, patched and unaffected targets. In this section, we discuss the potential deviation caused by other OSes, that is, whether other unaffected OSes/versions manifest the same feature and lead to false positive. We investigate the RFC 5961 implementation and IPID assignment strategy of four prevalent server OSes (Linux, Windows, FreeBSD, and OpenBSD), as depicted in Table IV. The results demonstrate that other OSes have only trivial impact on our measurement.

We can easily filter out vulnerable targets from a diversity of operating systems and versions. Linux (3.2 and above) is the sole OS that fully implements the challenge ACK mechanism. Hence, RFC 5961 can be utilized to filter out all non-linux OSes and earlier Linux (before 3.2). For the remaining Linux versions, the only deviation arises in kernel versions 3.2 and 3.3, which adopt a hybrid of per-socket IPID assignment strategy and per-destination IPID assignment strategy and produce the same IPID test results as vulnerable machine. We consider this deviation to be negligible since these are very early versions of Linux (released 12 years ago) and they switched to hash-based IPID assignment after only four months.

For Patch 1, the IPID of `SYN/ACK` packets for FreeBSD is also 0 and cannot be downgraded, except for patched Linux (5.16 and above). This is because FreeBSD cannot be fragmented with ICMP “Fragmentation Needed” embedded with ICMP echo reply. We exclude FreeBSD servers by observing if `DF` bits can be cleared. For Patch 2, only the patched Linux adopts the pure per-socket IPID assignment policy, and no other systems interfere.

B. *Interference of Middleboxes*

Measurements are inevitably affected by middleboxes, we summarize 4 potential impacts and possible mitigation. First, middleboxes may introduce packet loss, which is unavoidable in network transmission. To reduce the impact of packet loss, we adopt multiple vantage points and repeat each test three times. Second, middleboxes may apply specific filtering policies. For instance, we observed that the Tsinghua campus network filters partial challenge ACK packets. To minimize the impact of filtering policies, we select vantage points from different locations and networks. Third, middleboxes may filter out forged packets, and we discuss a trick to bypass the check on middleboxes in Section VI-B. Finally, as discussed in Section V-B, misconfiguration of some middleboxes may affect our measurements.

VIII. Related Work

Off-Path TCP Hijacking. The first TCP hijacking attack using predictable sequence numbers was proposed by Morris in 1985 [26]. Mitnick successfully hijacked Shimomura’s host in 1995 [27], raising public awareness of TCP security. Exploiting the TCP window mechanism, Watson constructed a blind in-window reset attack in 2003 [12], which could be performed without knowing the exact sequence number. RFC 5961 [28] proposed the challenge ACK mechanism in 2010, which increased the difficulty for attackers to carry out blind in-window attacks. Qian et al. exploited firewall middleboxes and some system counters to infer the sequence number and constructed two off-path TCP hijacking attacks in 2012 [4], [5]. Gilad et al. also proposed four off-path hijacking attacks in 2014 [29]. However, these attacks relied on malware running on the victim to obtain the port number. Cao et al. and Feng et al. constructed two pure off-path TCP hijacking attacks exploiting challenge ACK rate limit vulnerability and mixed IPID vulnerability in 2016 and 2020 respectively [6], [7]. Additionally, some TCP hijacking attacks occur in specific scenarios such as NAT [3] and WiFi [9], [10], [30].
Protocol Stack Large-Scale Measurements. Durumeric et al. proposed Zmap in 2013 [22], a scanning tool that can scan the entire IPv4 address space within 45 minutes. Luckie et al. measured the resilience of top websites, routers, and switches to blind attacks in 2015 [17]. They found a surprisingly high level of vulnerabilities in the TCP stack in use, which illustrates the fragility of the Internet. Quach et al. tracked the patch status of Challenge ACK rate limit vulnerability in 2016 [11]. They tracked the Top 1M websites for half a year since the release of the patch, illustrating the TCP stack vulnerability patching behavior of websites over time. Pan et al. measured the deployment of inbound source address validation (ISAV) and the reachability between arbitrary Internet nodes via the ICMP rate limiting channel in 2023 [31].

TABLE IV: RFC 5961 implementation and IPID assignment for four mainstream server OSes

OS	RFC 5961	RST IPID	SYN/ACK IPID	TCP socket IPID
Linux (prior to 3.2)	Not implemented	0	0/per-destination	per-socket/per-destination
Linux (3.2-3.3)	Fully implemented	0	0/per-destination	per-socket/per-destination
Linux (3.4-4.17)	Fully implemented	hash-based	0/hash-based	per-socket/hash-based
Linux (4.18-5.15)	Fully implemented	0	0/hash-based	per-socket/hash-based
Linux (5.16 and above)	Fully implemented	0	0/random	per-socket
Windows	Partially implemented	hash-based	hash-based	hash-based
Earlier FreeBSD *	Partially implemented	random	random	random
FreeBSD *	Partially implemented	0	0	0
OpenBSD*	Partially implemented	random	random	random
Old Linux/Windows	Not implemented	global	global	global

* FreeBSD and OpenBSD cannot be fragmented with ICMP "Fragmentation Needed" embedded with ICMP echo reply.

IX. Conclusion

In this work, we study a critical protocol stack vulnerability within the scope of the Linux ecosystem. After a horizontal study on the vulnerability against different Linux components, we discover that there is a substantial delay in the patching of the stack vulnerability, and the patching status varies among different ISPs and vendors. We find that the kernel and distributions do not release patches promptly, with delays of up to 2 years. And while all CDN vendors are well patched, we find more than 1/3 of the images provided by the cloud vendors are vulnerable. The patching status of different cloud vendors also varies greatly. Besides, we identify 22.5k websites that still have vulnerability among the top 1 million websites.

References

[1] J. Knockel and J. R. Crandall, "Counting packets sent between arbitrary internet hosts," in *4th USENIX Workshop on Free and Open Communications on the Internet (FOCI)*, 2014.

[2] G. Alexander, A. M. Espinoza, and J. R. Crandall, "Detecting tcp/ip connections via ipid hash collisions," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 4, 2019.

[3] X. Feng, Q. Li, K. Sun, Z. Qian, G. Zhao, X. Kuang, C. Fu, and K. Xu, "Off-Path network traffic manipulation via revitalized ICMP redirect attacks," in *31st USENIX Security Symposium (USENIX Security)*, 2022, pp. 2619–2636.

[4] Z. Qian, Z. M. Mao, and Y. Xie, "Collaborative tcp sequence number inference attack: how to crack sequence number under a second," in *Proceedings of the 2012 ACM conference on Computer and communications security (CCS)*, 2012, pp. 593–604.

[5] Z. Qian and Z. M. Mao, "Off-path tcp sequence number inference attack-how firewall middleboxes reduce security," in *2012 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2012, pp. 347–361.

[6] Y. Cao, Z. Qian, Z. Wang, T. Dao, S. V. Krishnamurthy, and L. M. Marvel, "Off-path tcp exploits: Global rate limit considered dangerous," in *25th USENIX Security Symposium (USENIX Security)*, 2016, pp. 209–225.

[7] X. Feng, C. Fu, Q. Li, K. Sun, and K. Xu, "Off-path tcp exploits of the mixed ipid assignment," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, p. 1323–1335.

[8] X. Feng, Q. Li, K. Sun, K. Xu, B. Liu, X. Zheng, Q. Yang, H. Duan, and Z. Qian, "Pmtud is not panacea: Revisiting ip fragmentation attacks against tcp," in *Proceedings of the Network & Distributed System Security Symposium (NDSS)*, 2022, pp. 24–28.

[9] X. Feng, Q. Li, K. Sun, Y. Yang, and K. Xu, "Man-in-the-middle attacks without rogue ap: When wpas meet icmp redirects," in *2023 IEEE Symposium on Security and Privacy (SP)*, 2022, pp. 694–709.

[10] D. Schepers, A. Ranganathan, and M. Vanhoef, "Framing frames: Bypassing wi-fi encryption by manipulating transmit queues," in *32th USENIX Security Symposium (USENIX Security)*, 2023.

[11] A. Quach, Z. Wang, and Z. Qian, "Investigation of the 2016 linux tcp stack vulnerability at scale," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 1, pp. 1–19, 2017.

[12] P. Watson, "Slipping in the window: Tcp reset attacks," 2004.

[13] J. Postel, "Internet protocol," Internet Requests for Comments, Internet Engineering Task Force, document RFC 791, Sep. 1981. [Online]. Available: http://www.rfc-editor.org/rfc/rfc791.txt

[14] E. Dumazet, "ipv4: tcp: send zero ipid in synack messages," 2022. [Online]. Available: https://lore.kernel.org/all/20220126200518.990670-2-eric.dumazet@gmail.com/

[15] ——, "ipv4: avoid using shared ip generator for connected sockets," 2022. [Online]. Available: https://lore.kernel.org/all/20220126200518.990670-3-eric.dumazet@gmail.com/

[16] "Usage statistics of operating systems for websites." [Online]. Available: https://w3techs.com/technologies/overview/operating_system

[17] M. Luckie, R. Beverly, T. Wu, M. Allman, and k. claffy, "Resilience of deployed tcp to blind attacks," in *Proceedings of the 2015 Internet Measurement Conference (IMC)*, 2015, pp. 13–26.

[18] E. Dumazet, "ipv4: tcp: send zero ipid for rst and ack sent in syn-recv and time-wait state," 2018. [Online]. Available: https://lore.kernel.org/all/20180822203045.76928-1-edumazet@google.com/

[19] R. S. Raman, A. Stoll, J. Dalek, R. Ramesh, W. Scott, and R. Ensafi, "Measuring the deployment of network censorship filters at global scale." in *Proceedings of the Network & Distributed System Security Symposium (NDSS)*, 2020.

[20] "Google ip address ranges." [Online]. Available: https://www.gstatic.com/ipranges/cloud.json

[21] J. Postel, "Internet control message protocol," Internet Requests for Comments, Internet Engineering Task Force, document RFC 792, Sep. 1981. [Online]. Available: https://www.rfc-editor.org/rfc/rfc792.txt

[22] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast internet-wide scanning and its security applications," in *22th USENIX Security Symposium (USENIX Security)*, 2013.

[23] K. Ruth, D. Kumar, B. Wang, L. Valenta, and Z. Durumeric, "Toppling top lists: Evaluating the accuracy of popular website lists," in *Proceedings of the 22nd ACM Internet Measurement Conference*, 2022, pp. 374–387.

[24] "The majestic million." [Online]. Available: https://majestic.com/reports/majestic-million

[25] V. L. Pochat, T. van Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," in *26th Annual Network and Distributed System Security Symposium, (NDSS)*, 2019.

[26] R. T. Morris, "A weakness in the 4.2 bsd unix tcp/ip software," 1985.

[27] T. Shimomura and J. Markoff, *Takedown: the pursuit and capture of Kevin Mitnick, America's most wanted computer outlaws-by the man who did it*. Hyperion Press, 1995.

[28] R. R. Stewart, M. Dalal, and A. Ramaiah, "Improving tcp's robustness to blind in-window attacks," Internet Requests for Comments, Internet Engineering Task Force, document RFC 5961, Aug. 2010. [Online]. Available: https://www.rfc-editor.org/rfc/rfc5961.txt

[29] Y. Gilad and A. Herzberg, "Off-path tcp injection attacks," *ACM Transactions on Information and System Security (TISSEC)*, vol. 16, no. 4, pp. 1–32, 2014.

[30] Y. Yang, X. Feng, Q. Li, K. Sun, Z. Wang, and K. Xu, "Exploiting sequence number leakage: Tcp hijacking in nat-enabled wi-fi networks," in *Network and Distributed System Security (NDSS) Symposium*, 2024.

[31] L. Pan, J. Yang, L. He, Z. Wang, L. Nie, G. Song, and Y. Liu, "Your router is my prober: Measuring ipv6 networks via icmp rate limiting side channels," in *Proceedings of the Network & Distributed System Security Symposium (NDSS)*, 2023.