DOI:10.1145/3689819

Attackers can use forged ICMP error messages to exploit vulnerabilities in the TCP/IP stack.

BY XUEWEI FENG, QI LI, KUN SUN, KE XU, AND JIANPING WU

Exploiting Cross-Layer Vulnerabilities: Off-Path Attacks on the TCP/IP Protocol Suite

THE TCP/IP PROTOCOL suite is a set of communication protocols underpinning the Internet. Protocols at different layers of the suite—for example, Wi-Fi, IP, TCP, and HTTP (Figure 1)—form the essential framework for data transmission on the Internet. But given the paramount significance of the TCP/IP protocol suite, it is also a pivotal target for myriad forms of attacks.[4,9,12,16,18,22,28] Vulnerabilities in the suite can have extensive repercussions, posing a fundamental threat to Internet security and presenting significant incentives to attackers. As a result, both industry and academia have dedicated substantial efforts[6,8,16,20,25,36,37] toward combating the diverse spectrum of network attacks. What has received limited attention, however, are vulnerabilities arising from cross-layer interactions among various protocols within the TCP/IP protocol suite, caused by forged Internet Control Message Protocol (ICMP) error messages. These vulnerabilities can be exploited by off-path attackers, posing risks to Internet security.

In the process of network data processing, protocols within the suite must interact and coordinate across layers. This cross-layer interaction ensures the smooth generation, transmission, reception, and storage of data. For example, when delivering an HTTP message, protocols such as DNS, TCP, IP, ARP, and Wi-Fi may need to be invoked to process and encapsulate the message. Although each protocol within the stack may individually possess sufficient robustness, combining these protocols and engaging in cross-layer interaction through function calls can introduce security issues or anomalies. Specifically, the proper execution of one layer's specific functionality can be compromised by the normal execution

key insights

- The TCP/IP protocol suite serves as the backbone of the Internet. Despite more than 40 years of development, its security remains a critical concern.
- By exploring the security implications of cross-layer interactions within the TCP/IP protocol suite, particularly those triggered by ICMP errors, we identified several significant vulnerabilities in modern TCP/IP implementations.
- There remains a continuous need to uncover subtle semantic vulnerabilities within the TCP/IP protocol suite, particularly through techniques that minimize manual effort, such as program analysis and AI-driven approaches.

IMAGE BY ANDRIJ BORYS ASSOCIATES, USING SHUTTERSTOCK.COM

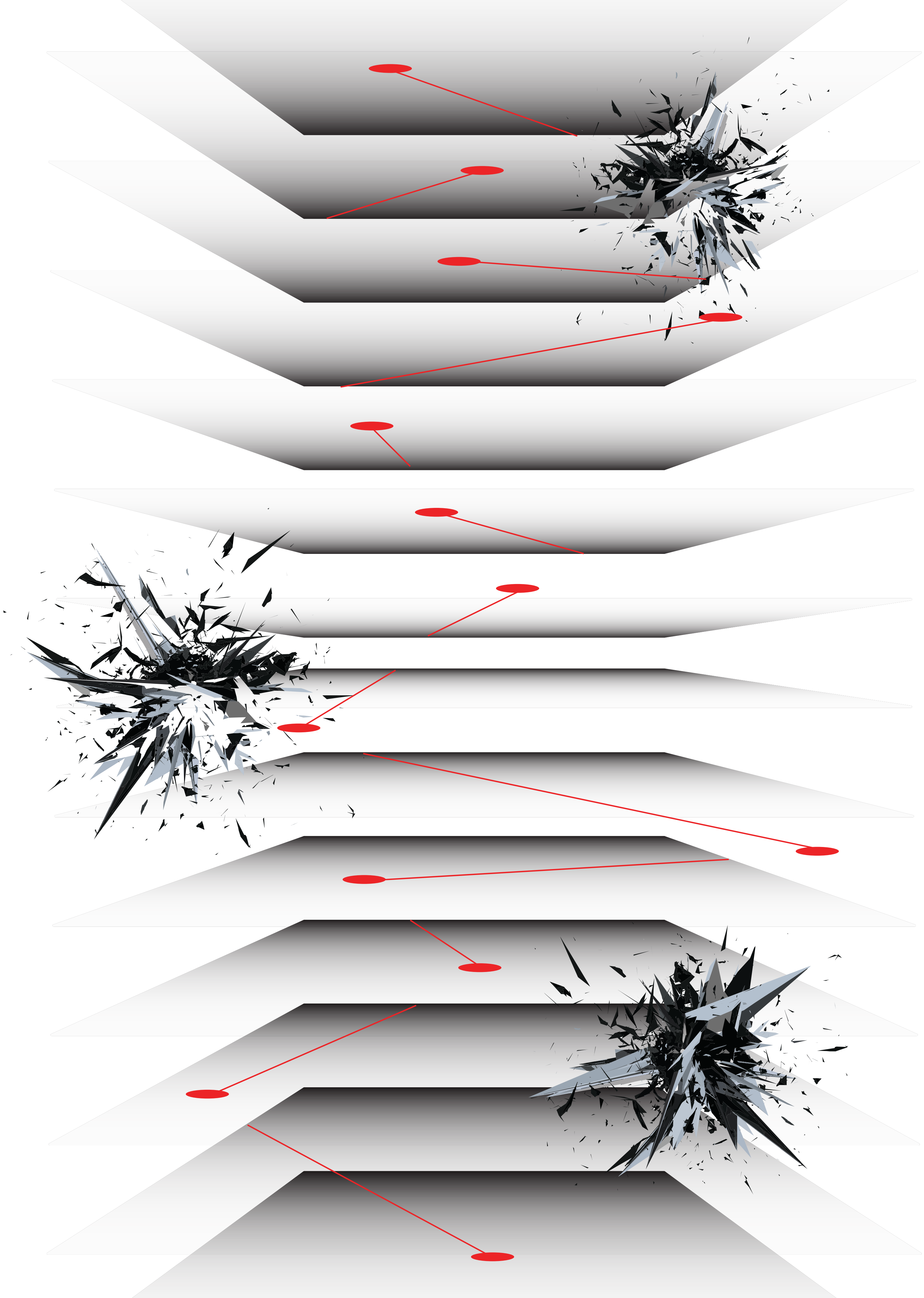


Figure 1. The TCP/IP protocol suite serves as the essential framework for data transmission on the Internet.

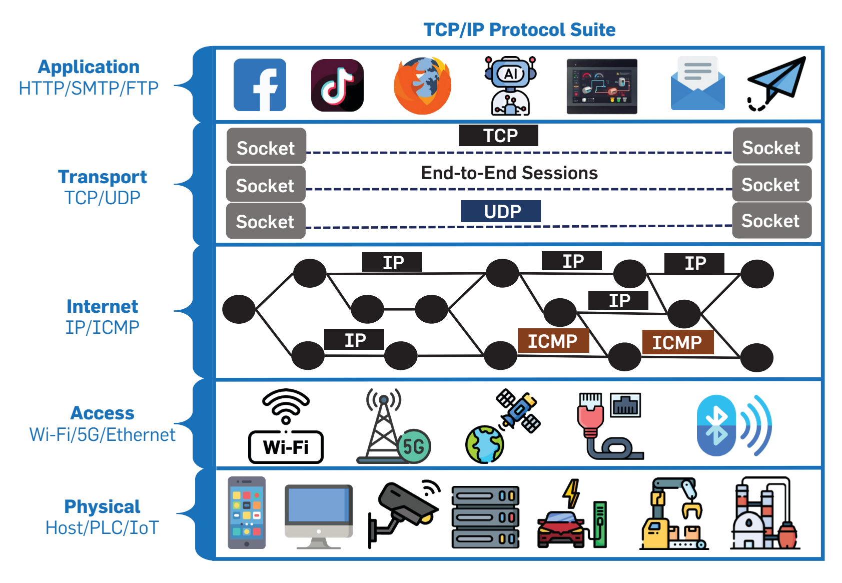
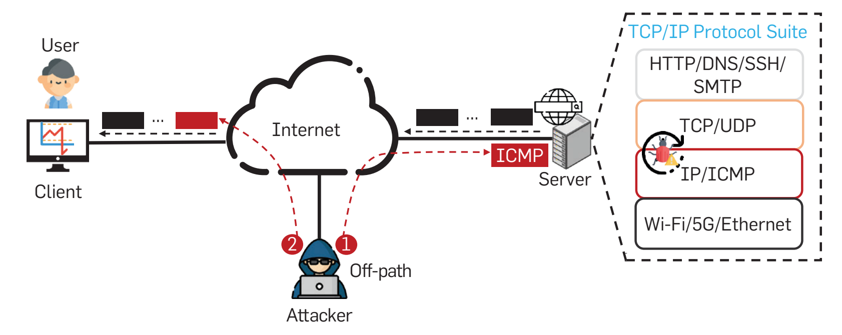


Figure 2. Threat model of off-path attacks on the TCP/IP protocol suite via forged ICMP error messages.



of other layers. For instance, the loss of frames in wireless networks commonly occurs due to inevitable communication-noise interference; however, at the TCP layer, if TCP segments are not promptly acknowledged due to the loss of wireless frames, it can mistakenly trigger the detection of network congestion, leading to inefficient execution of the TCP congestion-control algorithm.[34]

ICMP, recognized as a fundamental component of the TCP/IP protocol suite, frequently drives cross-layer interactions that transcend traditional network layer boundaries to report network conditions or errors. By operating directly on top of IP, ICMP error messages embedded with various payloads can influence the behavior of higher layers such as TCP and UDP, and can even be exploited by off-path attackers to compromise higher-layer protocols. Here, we undertake a comprehensive study to investigate the cross-layer interactions within the TCP/IP protocol suite caused by forged ICMP errors. In doing so, we uncover multiple vulnerabilities, including information leakage, desynchronization, semantic gaps, and identity spoofing. We discuss each of these in turn, but first we will provide some background on ICMP error messages and their associated threat model.

Basics of ICMP error messages. ICMP error messages are specific types of ICMP messages generated in response to network issues. They play a crucial role in identifying and diagnosing problems within a network.[3,5,35] These messages include Destination Unreachable (indicating network failures or host unavailability), Time Exceeded (resulting from packet time-to-live expiration), Parameter Problem (addressing IP header parameter issues), and Redirect Messages (for optimizing routing). Source Quench messages, historically significant for congestion control, are now deprecated. Aiming to report network issues to the receiver, ICMP error messages inevitably induce cross-layer interactions within the TCP/IP protocol stack and prompt the receiver to adjust its behavior based on the received ICMP error messages. According to the ICMP specifications,[3,5,35] ICMP error messages should contain at least the first 28 octets of the original packet that triggered the error message (that is, 20 octets of the IP header plus at least the first eight octets). When an ICMP error message is received, the receiver can use the embedded payload in the message to match it to the corresponding process. This enables the process to adapt and respond effectively. For example, when an ICMP Destination Unreachable message with the code "Packet too big" is received, it facilitates cross-layer interactions by enabling the receiver's TCP to reduce its maximum segment size (MSS), thereby avoiding IP fragmentation on the intermediate routes that issued the ICMP error message.

Unfortunately, in practice, it is easy for attackers on the Internet to forge ICMP error messages to manipulate the receiver's behavior, for a couple of reasons. First, because ICMP error messages can be generated by any intermediate router along the network path, it is difficult for the receiver to authenticate their source. This is particularly challenging because attackers can use IP address spoofing techniques to forge the source IP address. Second, although ICMP specifications require that error messages include at least the first 28 octets of the original packet, enabling the receiver to match the message and perform a legitimacy check, attackers can easily forge a 28-octet payload to bypass this check. In the context of TCP communication, the first 28 octets of the original packet contain a random sequence number, which is hard for attackers to guess. However, in UDP or ICMP scenarios, since these protocols are stateless and lack randomized sequence numbers, attackers can easily forge a 28-octet payload to include in the falsified ICMP error message. This allows them to evade the receiver's legitimacy check and deceive the receiver into responding to the message, leading to unintended protocol interactions that pose security risks.

Threat model of off-path attacks. Figure 2 shows the threat model of off-path attacks on the TCP/IP protocol suite via forged ICMP error messages. The off-path attacker is positioned outside the direct communication path between the server and the client. Consequently, the attacker cannot intercept or directly modify packets in transit between the server and the client. Instead, the attacker can forge and send packets with arbitrary source IP addresses.[2,a] Specifically, by leveraging forged ICMP error messages, the attacker exploits weaknesses and forces exceptional behaviors during cross-layer interactions among multiple protocols within the server's TCP/IP protocol suite. Once these vulnerabilities are triggered, network traffic from the server to the client will be affected. Furthermore, the off-path attacker can impersonate the server and inject crafted packets into the client to manipulate the target network traffic. The following four sections delve into work of ours that identified vulnerabilities (information leakage,[9,10] desynchronization,[12] semantic gap,[11] and identity spoofing[13]) caused by forged ICMP error messages, enabling off-path attackers to launch impactful attacks.

By operating directly on top of IP, ICMP error messages embedded with various payloads can influence the behavior of higher layers such as TCP and UDP, and can even be exploited by off-path attackers to compromise higher-layer protocols.

Information Leakage

TCP plays a fundamental role within the TCP/IP protocol suite and is an important part of the Internet, ensuring data packets reach their intended destinations accurately and in the correct sequence. A key security measure within the TCP protocol is 32-bit randomization of sequence and acknowledgment numbers. This strengthens the protocol's resilience against out-of-band malicious TCP packet injections. However, despite the extensive randomized sequence and acknowledgment number space, which significantly increases the time needed for brute-force attacks, TCP protocol operations involve interactions with other protocols in the TCP/IP protocol suite. During these interactions, certain fields of other layer protocols, such as the Identification field of the IP protocol (IPID), can be exploited to infer the TCP protocol's sequence and acknowledgment numbers. In particular, we discovered that the IPID field, even with the most advanced IPID assignment policy currently available in Linux systems, can be manipulated by a forged ICMP error message issued by off-path attackers. This manipulation allows the attacker to indirectly infer confidential information (that is, the sequence and acknowledgment numbers of TCP) by observing the IPID field, ultimately leading to information leakage during protocol cross-layer interactions. This can enable the off-path attacker to inject malicious TCP packets into the target connection, thereby jeopardizing the integrity of the associated TCP stream.[9]

IPID assignment. The IPID field is used to enable defragmentation. After abandoning previous vulnerable IPID assignment methods (for example, global IPID assignment and per-destination IPID assignment), modern operating systems typically employ advanced methods to assign IPIDs for IP packets. For instance, Linux systems use a per-socket-based IPID assignment policy for TCP packets and 2,048 globally shared hash counters for non-TCP packets.[1] Though this IPID assignment method aims to safeguard TCP protocols against information leakage stemming from IPID values, we demonstrated a vulnerability that can be exploited by off-path attackers to deduce the upper-layer sequence and acknowledgment numbers of a victim TCP connection.

Inference of randomized numbers. In this situation, the off-path attacker pretends to be a router and issues a crafted ICMP error message (an ICMP

[a] Prior studies show that about a quarter of ASes on the Internet do not filter packets with spoofed source addresses leaving their networks, and it is trivial to rent such a machine from a bulletproof hosting node.[26–28]

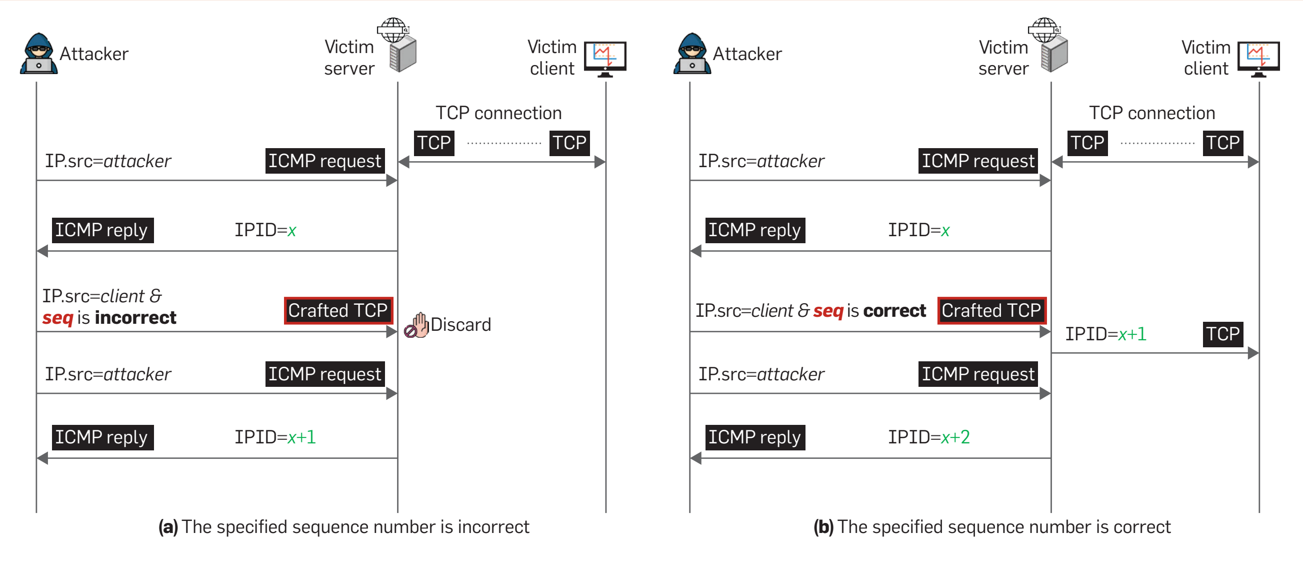
Destination Unreachable message with the code "Packet too big") embedded with a 28-octet payload of a fake ICMP echo reply packet to a Linux server. This crafted ICMP error message evades the server's legitimacy check and deceives it into downgrading its IPID assignment policy for TCP packets. The policy transitions from a per-socket-based policy to the utilization of 2,048 globally shared hash counters. Given the limited size of the hash counter pool (2,048), the attacker can change its IP address to successfully provoke a hash collision with a victim TCP client of the server.[b] This occurs because Linux servers select one of the 2,048 hash IPID counters based on the destination IP address of outgoing packets. Consequently, the server can be tricked into selecting the same IPID counter for both the attacker's IP address and the victim client's IP address. This situation allows the attacker to deduce the specific IPID counter being used by the Linux server for the victim TCP connection, thus creating a side channel to leak connection information.

Once the shared IPID counter is identified, the attacker proceeds to send crafted TCP packets to the victim server. The shared IPID counter will exhibit varying behaviors under different circumstances, enabling the attacker to discern whether the specified values in the forged TCP packets are correct. As shown in Figure 3, the attacker initiates the process by sending an ICMP echo request packet to the server and monitors the current value of the shared IPID counter when the server responds with a reply packet. Subsequently, the attacker impersonates the identity of the victim client by IP spoofing and crafts a TCP packet destined for the server. This crafted packet includes the specified sequence number (*seq*). In this scenario, the server's behavior varies based on the sequence number specified within the crafted packet. As illustrated in Figure 3a, when the specified sequence number is incorrect (that is, not within the server's receive window), the server simply discards the packet. When the attacker later observes the current value of the shared IPID counter once more, it will notice that the IPID counter's values remain consecutive.

If the specified sequence number is correct (as shown in Figure 3b), the server will generate a reply packet destined for the client, even though the client will ultimately discard this reply. This reply packet consumes a value from the shared IPID counter. When the attacker subsequently observes the current value of the shared IPID counter once more, it will notice that the IPID counter's values are no longer consecutive. By making this comparison, the attacker can accurately deduce sensitive information, such as the sequence and acknowledgment numbers of the target TCP connection. For off-path TCP injection attacks (as described in Cao et al.[6] and Qian and Mao[36]), once off-path attackers identify the randomized sequence and acknowledgment numbers of the target TCP connection, they can craft an out-of-band TCP packet specified with these identified numbers in the TCP header. When injected into the target connection, this packet will pass verification and be accepted by the receiver, potentially terminating or poisoning the connection.

Experimental results. Through real-world evaluations, we have demonstrated that the information leakage caused by cross-layer interactions can have severe real-world consequences, enabling attackers to infer and disrupt a large number of TCP connections. We found that more than 20% of the Alexa top 100k websites are vulnerable. To test this, we first establish a TCP connection from our client to each of the websites in the Alexa top 100k list. Then, an attack machine on the Internet issues a forged ICMP "Packet too big" message to the website to manipulate its IPID assignment for our client. Our results show that 20% of the websites can be tricked into downgrading the IPID assignment from the per-socket-

Figure 3. The attacker determines the accuracy of the specified sequence number by observing the shared IPID counter.[9]

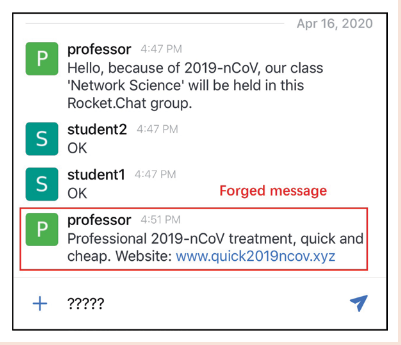


(a) The specified sequence number is incorrect

(b) The specified sequence number is correct

[b] Since kernel version 5.12.4, Linux has used a dynamic hash counter pool proportional to physical RAM size to mitigate IPID-based firewall attacks.[24]

Figure 4. Snapshot of Web application poisoning.[9]



Apr 16, 2020

P professor 4:47 PM
Hello, because of 2019-nCoV, our class 'Network Science' will be held in this Rocket.Chat group.

S student2 4:47 PM
OK

S student1 4:47 PM
OK

Forged message

P professor 4:51 PM
Professional 2019-nCoV treatment, quick and cheap. Website: www.quick2019ncov.xyz

+ ?????

based policy to the hash-based policy for their TCP packets after receiving forged ICMP error messages. We have implemented a prototype to perform case studies on a wide range of applications—for example, HTTP, SSH and BGP—to validate the effectiveness of the identified off-path TCP hijacking attack due to cross-layer information leakage. We've shown that an off-path attacker can infer the sequence number of a target TCP connection on port 22 within 155 seconds, thus crafting an out-of-band TCP RST packet to tear down the victim SSH session to cause a denial-of-service (DoS) attack. In addition, the attacker can infer the sequence and acknowledgment numbers of a target TCP connection within 215 seconds, thus crafting a TCP data packet to poison Web applications or BGP routing tables.[9] Figure 4 is a snapshot of our attack against a Web application, in which an attacker identifies a TCP connection and proceeds to inject a fake message.

Desynchronization

Desynchronization within the TCP/IP protocol suite caused by crafted ICMP errors refers to a situation where multiple protocols simultaneously work with the same variable or data unit. Factors such as network delays or conditional competition introduced by a crafted ICMP error message can cause these protocols to lose synchronization, leading to ambiguity around the value of that variable or data unit. This disruption can degrade the network's original functionality or semantics, creating opportunities for attackers to exploit and compromise network systems.

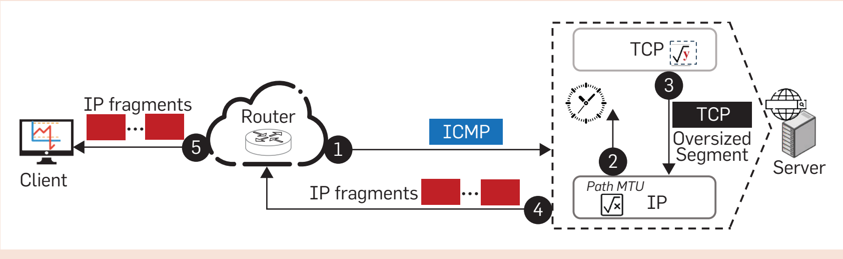
Consider the path MTU value, which is a global variable maintained within the host's IP layer. This value defines the maximum IP packet size for the path from the host to a specific destination IP address. Operations on the path MTU value extend beyond the IP protocol and involve various other protocols, such as TCP and UDP. Ideally, using the path MTU value to determine TCP segment size should eliminate the need for IP fragmentation. However, we have demonstrated that, in practice, simultaneous updates on this global variable by various protocols, tricked by a crafted ICMP error message, can lead to desynchronization issues.[12] This can result in discrepancies between the path MTU value at the IP layer and the MTU value read by the TCP layer, potentially causing the TCP layer to transmit oversized segments, leading to abnormal IP fragmentation. Consequently, off-path attackers can inject manipulated IP fragments into the target TCP connection, causing the mis-reassembling of IP fragments and disrupting the target TCP traffic without needing to infer random sequence numbers.

Forcing IP fragmentation on TCP. It is a widespread belief that TCP is immune to IP fragmentation because TCP enables path MTU discovery (PMTUD) by default. This mechanism detects the maximum allowed packet size along the path and enables TCP to adjust the maximum segment size (MSS) accordingly, thus avoiding IP fragmentation on TCP.[29,31] In practice, the detected path MTU value is a global variable maintained at the IP layer. Consequently, when multiple protocols, such as IP, TCP, UDP, and others, simultaneously interact with it, unexpected synchronization issues may occur, resulting in unintended IP fragmentation on TCP segments.

As shown in Figure 5, a router on the Internet may generate an ICMP error message (an ICMP Destination Unreachable message with the code "Packet too big") directed at the server. This ICMP error message can be triggered by various protocol sessions from the server, such as UDP or ICMP echo. Upon reaching the server, this message updates the global variable of path MTU in the IP layer based on its contents. However, as this message lacks specific TCP connection information, the update to the path MTU value is not immediately synchronized with the TCP layer. Instead, the IP layer defers feedback until it passively detects the TCP connection by receiving oversized TCP segments, which it then fragments and sends out. Once the IP layer acknowledges the TCP connection, it updates the TCP layer with the new path MTU value, allowing TCP to adjust the MSS of subsequent segments to avoid IP fragmentation.

This desynchronization issue concerning the path MTU value between TCP and IP undermines the primary purpose of the path MTU discovery mechanism and causes unintended IP fragmentation on TCP segments. In particular, we find that off-path attackers on the Internet can impersonate a router and forge such an ICMP error message to trick the server into fragmenting its TCP segments. This manipulation exploits the inherent challenge in verifying the source and transmission path of ICMP error messages within the current Internet infrastructure. For example, we can forge the ICMP error message to include an embedded ICMP echo reply packet, effectively tricking the server into fragmenting TCP segments and introducing a new attack vector.

Figure 5. IP fragmentation on TCP segments due to desynchronization of the path MTU value between IP and TCP.



Poisoning TCP traffic via IP fragmentation. Once TCP packets experience IP fragmentation due to the desynchronization issue, an off-path attacker may exploit this vulnerability to launch IP fragmentation injection attacks against TCP traffic. As shown in Figure 6, at first the off-path attacker may employ various techniques, such as social engineering or network side channels, to detect the existence of a TCP connection between a victim server and a client. Then, the attacker forges an ICMP error message and sends it to the server, triggering the desynchronization vulnerability on path MTU in the server's TCP/IP protocol suite. This manipulation causes IP fragmentation on the TCP packets sent from the server to the client. Following this, the attacker impersonates the server via IP spoofing and sends crafted IP fragments to the victim client. As a result, legitimate fragments from the server will be incorrectly reassembled with the malicious ones introduced by the attacker. Ultimately, this leads to the replacement of the original data within the TCP packets, initiating a poisoning attack on the targeted TCP stream. According to RFC 791, the minimum IP fragments on the Internet is 68 octets; thus, the random sequence and acknowledgment numbers are always carried in the first benign fragment from the server. Consequently, IP-fragmentation-based poisoning attacks against TCP can be performed without the need to infer the random sequence and acknowledgment numbers.[c]

Experimental results. We demonstrated that off-path attackers can manipulate HTTP traffic via our attack. A malicious JavaScript installed at the victim client via spam aids the attacker in synchronizing timing and aligning data to poison the local Web cache.[d] The connection to the target HTTP server is established by the puppet, and the connection and segments from the HTTP server are known to the attacker. Consequently, leveraging our method, the attacker can craft subtle IP fragments to force the incorrect reassembly of both legitimate and malicious fragments, thereby poisoning the client's Web cache, leading to regular users encountering poisoned local cache data when accessing the HTTP server later.

Furthermore, we showed that an off-path attacker can manipulate BGP routing tables via our attack. The attacker first probes periodically advertised BGP messages in advance.[15] Then, it manipulates BGP routers into fragmenting TCP segments by sending forged ICMP error messages. Finally, the attacker injects forged fragments into the BGP messages to poison the routing tables. Figure 7 illustrates the altered routing information received by a victim BGP router within our test-bed environment, which differs from the original routing information advertised by its peer BGP router. In this scenario, the attacker has replaced the legitimate routing information of 10.2.2.0/24 with a counterfeit entry of 12.2.0.0/24 by injecting meticulously crafted IP fragments into the victim BGP router. Our experimental findings indicate that these attacks can pose a significant threat to Internet infrastructure. It is worth noting that a session encryption mechanism (for example, TLS) will mitigate the identified IP fragmentation attacks against TCP, since the mis-reassembled TCP segment cannot pass the up-layer verification and will be discarded. However, the discarding of the mis-reassembled TCP segment will incur a performance loss, since benign fragments are also discarded together.

Protocols may inherently fall short in comprehensively addressing the wide spectrum of data types and exceptional scenarios when processing packets carrying cross-layer data.

Semantic Gap

Protocols may inherently fall short in comprehensively addressing the wide spectrum of data types and exceptional scenarios when processing packets carrying cross-layer data, giving rise to gaps in understanding that hinder the proper response to such packets. To maintain network functionality, protocols may resort

[c] It is worth noting that the handling of overlapped IP fragments is an implementation decision. Popular operating systems (for example Linux, OpenBSD, Windows) handle overlapped IP fragments on a first-come, first-served basis,[12] which allows attackers to send crafted IP fragments to the victim client in advance, facilitating the construction of our attack.

[d] The malicious JavaScript is sandboxed by the client's browser, having limited privileges, and cannot access any information within the TCP/IP protocol suite.[12]

to employing default and imprecise processing methods when responding to these packets, thereby introducing the possibility of semantic mismatches—semantic gaps—that attackers can exploit to compromise the system's security. Specifically, we uncovered that, due to such a semantic-gap vulnerability in the legitimacy checks against ICMP error messages, an off-path attacker on the Internet can craft an ICMP redirect message to evade the receiver's (for example, a public server) checks. This tricks the receiver into modifying its routing table incorrectly and forwarding its IP traffic to black holes, thereby conducting a stealthy DoS attack against public servers on the Internet.[11]

DoS via semantic gap of ICMP error checks. Figure 8 illustrates our design for constructing a DoS attack against a victim server, redirecting its traffic intended for the victim client into a black hole hosted by a neighboring host of the server that is unable to forward network traffic. Initially, the server can successfully forward its traffic to the client. An off-path attacker on the Internet identifies a neighboring host near the target server through actions like ICMP echo requests (for example, using the ping tool). This host will later serve as a routing black hole. The attacker then impersonates the victim client by IP spoofing and sends a crafted UDP request to the server. Deceived by the request, the server responds with a UDP reply to the victim client, which the victim client eventually discards.

The attacker then embeds the predictable UDP reply packet into a crafted ICMP redirect message and sends it to the victim server. According to the ICMP specifications,[3,5,35] the server will check the first 28 octets of the embedded UDP reply packet to validate the legitimacy of the received ICMP redirect message, thereby verifying the existence of the corresponding UDP socket, even though it cannot check any further information due to UDP's stateless nature. Since the attacker previously tricked the server into establishing the UDP socket for the victim client, this crafted ICMP redirect message will pass the server's legitimacy check. Consequently, the server will mistakenly accept and respond to the message, redirecting its traffic for the victim client to the neighboring host as specified by the forged ICMP redirect message. However, the neighboring host lacks routing and forwarding capabilities and will discard the server's traffic. This results in a cross-layer DoS attack on all network sessions above the IP layer of the server.

Experimental results. We conducted large-scale measurements on the Internet, revealing that this DoS attack, due to the semantic-gap vulnerability in the ICMP error message's legitimacy check mechanism, can be exploited to pose a significant threat to the Internet. In our ethical measurement studies, we first initiate a session between our controlled client and the target server (for example, an HTTP session from our Web client to a public HTTP server). Then, using the identified ICMP redirect DoS at-

Figure 6. Poisoning TCP traffic via IP fragmentation.[12]

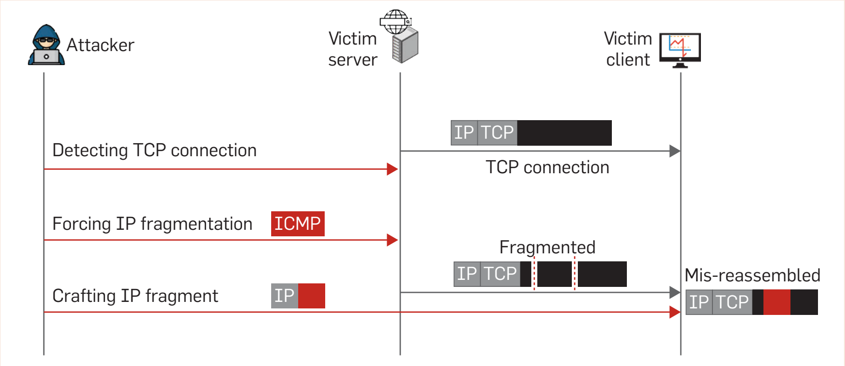
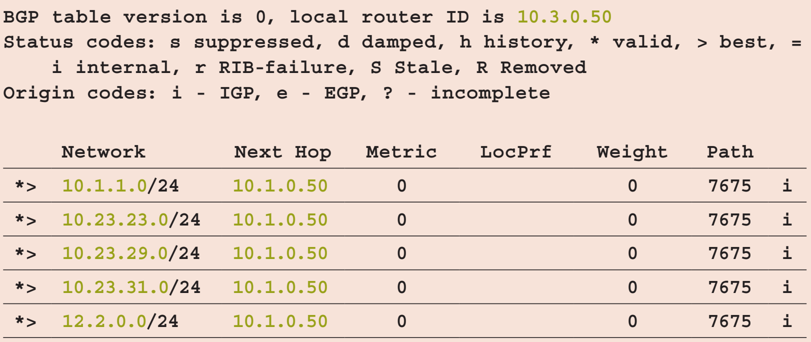


Figure 7. Fake routing due to IP fragments injection.[12]



```
BGP table version is 0, local router ID is 10.3.0.50
Status codes: s suppressed, d damped, h history, * valid, > best, =
    i internal, r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

     Network        Next Hop    Metric    LocPrf    Weight    Path
 *>  10.1.1.0/24    10.1.0.50   0                   0         7675   i
 *>  10.23.23.0/24  10.1.0.50   0                   0         7675   i
 *>  10.23.29.0/24  10.1.0.50   0                   0         7675   i
 *>  10.23.31.0/24  10.1.0.50   0                   0         7675   i
 *>  12.2.0.0/24    10.1.0.50   0                   0         7675   i
```

Figure 8. DoS via semantic gaps of ICMP error checks.[11]

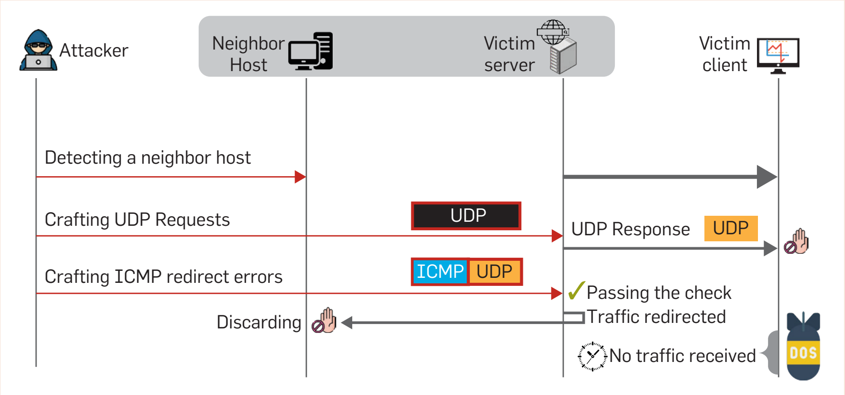
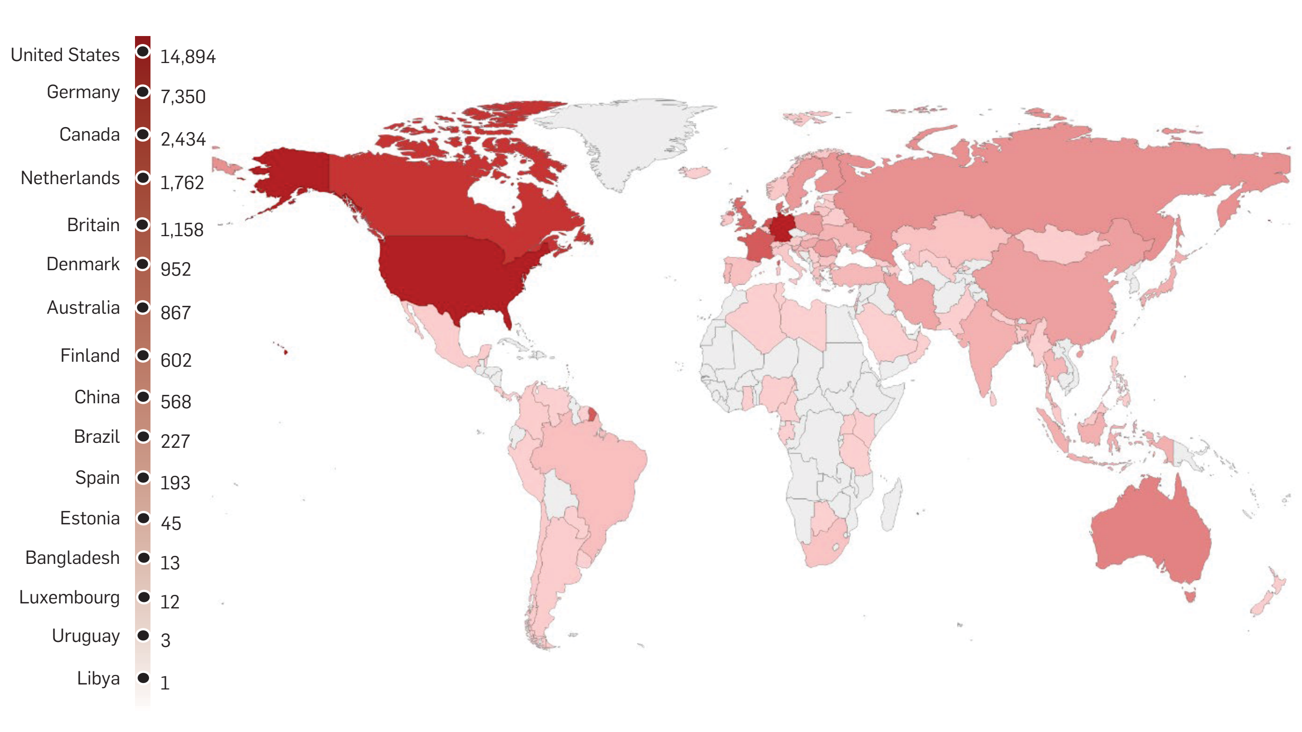


Figure 9. Distribution of websites with the semantic-gap vulnerability.[11]



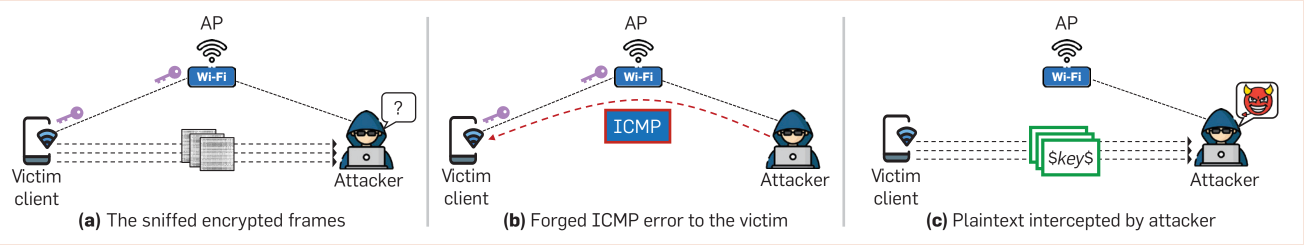
tack, we redirect the "server-to-our-client" traffic to a black hole. This causes subsequent requests from our controlled client to the server to fail, demonstrating that the server is vulnerable to our attack while not affecting the server's regular users. Our experimental results show that the identified DoS attack can target not only individual users, preventing them from visiting a Web server, but also server-to-server communication, such as shutting down a DNS resolver from contacting a particular authoritative name server (under our control in the experiments due to ethical considerations) to resolve domain names. It is even possible to disrupt the entire operation of a service such as Tor by breaking down the communication between a Tor relay node and a next hop. Our one-month empirical study on the Internet revealed that 43,081 popular websites, 54,470 open DNS resolvers, and 186 Tor relay nodes, spanning 5,184 autonomous systems (ASes) across 185 countries, are vulnerable to the semantic gap vulnerability and susceptible to the identified remote DoS attack. Figure 9 shows the geographical distribution of the vulnerable websites we detected.

Identity Deception

The problem of identity deception stems from the lack of security auditing for data sources during cross-layer interactions among multiple protocols within the TCP/IP protocol suite, a particular source of ICMP errors. This allows attackers to craft specific control protocol data, disrupting the normal operation of the network. We show that in specific network scenarios, such as Wi-Fi networks, identity deception can be particularly severe, presenting one of the most common challenges.[13] In public Wi-Fi networks such as those found in airports, coffee shops, campuses, and hotels, an attacker (a malicious client) may connect to the network and employ source-IP-address spoofing techniques to impersonate the access point (AP) gateway. The attacker can then send forged ICMP routing update control messages (that is, ICMP redirect messages designed for AP gateways only in Wi-Fi networks) to other clients. These forged ICMP routing update control messages will pass through the AP gateway; if the AP gateway fails to block the forged messages that finally arrive at other clients, these clients will be tricked into following the messages' instructions and setting the attacker as their new AP router, granting the attacker the ability to intercept traffic within the Wi-Fi network. What is more concerning is that this vulnerability enables the attacker to evade Wi-Fi protocol security measures such as WPA3, granting access to plaintext traffic.

Wi-Fi traffic hijacking. Figure 10 shows an overview of how to intercept plaintext traffic in Wi-Fi networks by leveraging the identity-deception vulnerability. In Wi-Fi networks, due to the shared nature of wireless channels, a malicious client may eavesdrop on wireless frames belonging to other clients. These frames, however, are usually encrypted by security mechanisms at the link layer, such as WPA2 or WPA3. As a result, it is difficult for the attacker to directly access plaintext information. We discovered a security vulnerability within the network processing unit (NPU) employed in AP routers. Driven by the quest for high-speed packet forwarding, these NPU chips within AP routers directly forward received ICMP messages (including forged ICMP errors from an attack)

Figure 10. Wi-Fi traffic interception via identity deception.



(a) The sniffed encrypted frames **(b)** Forged ICMP error to the victim **(c)** Plaintext intercepted by attacker

at the hardware level, thus failing access control list (ACL) rules defined at the higher layers to verify and block forged messages.

As shown in Figure 10b, this vulnerability allows the attacker to impersonate the AP router and craft an ICMP redirect message to manipulate the IP routing of the victim client. Even though such a message is meant to originate exclusively from the AP router itself and exhibits obvious illegitimate characteristics (for example, its source being the AP router's IP address), due to the NPU's direct forwarding of the message, this message passes through the AP router and remains unblocked. Ultimately, the message reaches the victim client, which is deceived into believing it originated from the AP router. As a result, the victim client updates its IP routing, designating the attacker as the next-hop gateway. This causes all subsequent traffic from the victim client to be rerouted through the attacker.

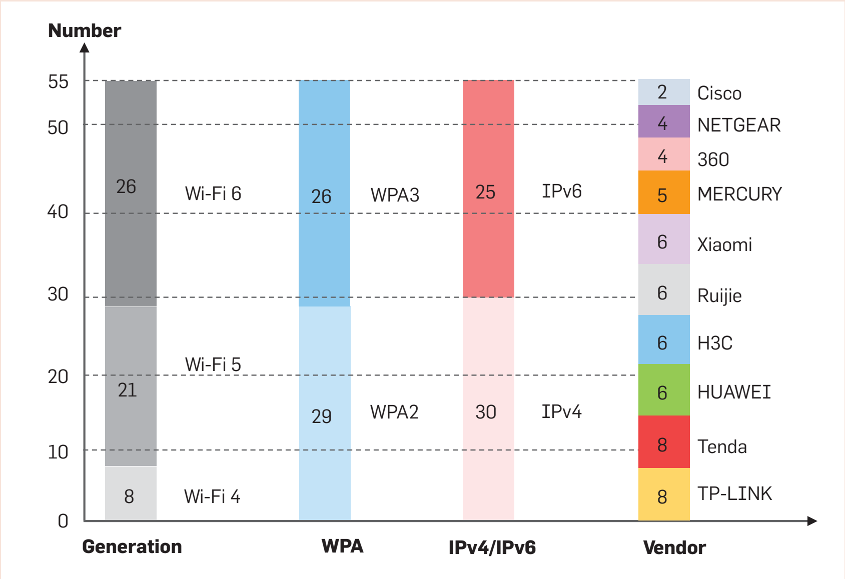
Note that this attack can bypass the link-layer encryption-protection mechanisms employed in Wi-Fi networks (for example, WPA2 and WPA3). WPA2 and WPA3 provide per-hop encryption at the link layer using a session key shared between the AP and each attached client. Due to the crafted ICMP redirect message, however, the victim client sets the attacker as the next hop in the IP layer. Therefore, when the AP receives the encrypted link-layer frames from the victim client, it needs to perform multi-hop relaying at the link layer to complete forwarding the frames to the next hop (that is, the attacker). Consequently, the AP first decrypts the encrypted frames using the shared secret key with the victim client. Next, according to the Destination Address (which has been poisoned by the attacker) in the frame header, the AP encrypts the frames using the secret key shared with the attacker and sends them to the attacker. Finally, after decrypting the frames, the attacker can intercept the victim client's plaintext traffic, and the link-layer per-hop encryption in Wi-Fi networks is successfully evaded.[13]

Experimental results. We conducted real-world evaluations to assess the impact of our attack. Initially, we investigated whether popular AP routers could effectively block forged ICMP redirect messages sent from an attacker to a victim client. Our assessment covered 55 popular wireless routers spanning 10 vendors (as shown in Figure 11). Our findings revealed that none of these routers block forged ICMP redirects from passing through. The root cause of this identity deception vulnerability, stemming from the flawed design of NPUs, has been officially recognized by Qualcomm (CVE-2022-2566) and HiSilicon (HWSA21-085272813). HUAWEI, H3C, Ruijie, MERCURY, NETGEAR, and Tenda have also confirmed the presence of this vulnerability in their AP routers due to the NPU. Furthermore, we evaluated 122 real-world Wi-Fi networks across various locations, including coffee shops, hotels, libraries, cinemas, and campuses, finding that 109 of these networks were vulnerable.

Countermeasures

We responsibly disclosed the identified vulnerabilities to the affected organizations. We reported the IPID assignment policy vulnerability, triggered by a forged ICMP "Packet too big" message, to the Linux community. They acknowledged it (CVE-2020-36516) and improved the IPID design starting from kernel version 5.16. The desynchronization and semantic-gap vulnerabilities, which are exploitable

Figure 11. Distribution of 55 vulnerable AP routers.[13]



for IP fragmentation and remote DoS attacks, were reported to Linux and FreeBSD. Both confirmed receipt, and we are awaiting updates. Qualcomm acknowledged and fixed the Wi-Fi identity deception vulnerability caused by crafted ICMP redirects in their Snapdragon chipsets (CVE-2022-2566); other affected vendors are still working on fixes. We also reported the vulnerabilities in the legitimacy-check mechanism of ICMP errors to the Internet Engineering Task Force (IETF) and are discussing our countermeasures with them.

Enhancing ICMP error authentications. The root cause of the four off-path attacks presented in this article is that an off-path attacker can forge ICMP error messages to bypass the receiver's legitimacy checks, leading to unintended protocol interactions and vulnerabilities. The most straightforward prevention measure is to strengthen the authentication of received ICMP error messages. However, as discussed earlier, verifying the legitimacy of ICMP errors is challenging due to two inherent limitations in the current ICMP specifications. First, certain ICMP errors (for example, ICMP Destination Unreachable messages with the code "Packet too big" exploited to trigger the information leakage and desynchronization vulnerabilities) can originate from any intermediate router, rendering source-based blocking ineffective. Second, although ICMP specifications mandate including at least the first 28 octets of the original packet, off-path attackers can evade this by embedding a crafted UDP or ICMP payload into the forged ICMP error messages, thwarting authentication due to the statelessness and lack of memory in the UDP and ICMP protocols.

Inspired by RFC 5961's challenge ACK mechanism[37] for defending against out-of-band TCP packet injection, we propose enhancing ICMP error authentication by introducing a new *challenge-and-confirm* mechanism. In particular, when a receiver gets an ICMP error message embedded with a stateless protocol payload (like UDP/ICMP), verifying its authenticity can be difficult. To address this, the receiver can send another (UDP/ICMP) packet on the established network session to the destination, embedding a hash value in the IP options field. If the prior ICMP error message was legitimate, this new packet will trigger another ICMP error message containing the hash value. This allows the receiver to verify authenticity and respond correctly. This challenge-and-confirm mechanism effectively defends against off-path forged ICMP error messages with minimal changes to the TCP/IP protocol suite. It only requires updates to the ICMP error message verification code on end hosts, without modifying intermediate routing devices, and it is backward compatible. We are discussing this mechanism with the IETF.

Our one-month empirical study on the Internet revealed that 43,081 popular websites, 54,470 open DNS resolvers, and 186 Tor relay nodes, spanning 5,184 autonomous systems across 185 countries, are vulnerable to the semantic gap vulnerability.

Securing sessions via cryptography. Another mitigation method is to use cryptography to secure network sessions as much as possible, such as with TLS,[38] QUIC,[21] and TCP-MD5/TCP-AO.[40] This way, even if an off-path attacker exploits forged ICMP error messages to trigger vulnerabilities in the TCP/IP stack, it is difficult for the attacker to cause real harm to applications. For instance, even if an attacker manipulates the server's IPID with ICMP error messages to create a side channel and guesses the sequence number of a target TCP connection, the injected TCP packet will fail TCP-MD5/TCP-AO or TLS validation and be discarded. Similarly, if an off-path attacker intercepts a victim client's packets in a Wi-Fi network as a man in the middle and evades link-layer encryption like WPA3, the end-to-end encryption provided by protocols such as TLS or QUIC makes it challenging for the attacker to access plaintext application data, thereby limiting the attack's impact.

Conclusion and Future Work

Off-path attacks on the TCP/IP protocol suite present a significant challenge to Internet security, as they do not constrain the attacker's network topology and require minimal resources. Previous research has demonstrated that off-path attackers can exploit vulnerabilities in the TCP/IP protocol suite to launch various attacks, such as TCP hijacking,[6,33,36] routing manipulation,[32] and Web and DNS cache poisoning.[16,17,19,23] However, off-path attacks facilitated by forged ICMP errors have received limited attention.[22,28]

In our study, we systematically revealed four security issues caused by forged ICMP errors: information leakage, desynchronization, semantic gaps, and identity deception. These issues can be exploited by attackers to pose severe security threats to the Internet. Essentially, these security issues arise from the disruption of the protocol's intended communication processes and semantic integrity by forged ICMP error messages, leading to unexpected behaviors that attackers can exploit. We call these *vulnerabilities,* as they are protocol interaction semantic vulnerabilities caused by forged ICMP errors, distinguishing them from memory corruptions caused by unsafe programming practices. Given that ICMP (including ICMPv6) is widely implemented and crucial across various TCP/IP protocol stacks, the semantic vulnerabilities caused by forged ICMP errors may extend beyond the four we have identified.

A critical area of focus for future research is automated identification of these semantic vulnerabilities, for example, by leveraging techniques from program analysis[7,14] and AI models.[30,39] In program analysis, data-flow analysis can be employed to trace the movement of packets through the protocol stack, helping to detect vulnerabilities such as desynchronization issues during packet data processing. AI models trained on network traffic patterns can identify anomalies, enabling the early detection of potential vulnerabilities. By integrating these approaches, it may be possible to develop more proactive and automated methods for identifying and mitigating security risks within network protocols.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3102301, the National Science Foundation for Distinguished Young Scholars of China under No. 62425201, the Science Fund for Creative Research Groups of the National Natural Science Foundation of China under No. 62221003, and the Key Program of the National Natural Science Foundation of China under No. 62132011 and No. 61932016.

References

1. Alexander, G., Espinoza, A.M., and Crandall, J.R. Detecting TCP/IP Connections via IPID hash collisions. *Proceedings on Privacy Enhancing Technologies 4* (2019), 311–328.
2. Ali, F. IP Spoofing. *The Internet Protocol J. 10*, 4 (2007), 1–9.
3. Baker, F. *Requirements for IP Version 4 Routers*. RFC 1812. Internet Engineering Task Force, (1995); http://www.rfc-editor.org/rfc/rfc1812.txt
4. Bellovin, S.M. A look back at "security problems in the TCP/IP protocol suite". In *20th Annual Computer Security Applications Conf.* IEEE, (2004), 229–249.
5. Braden, R. *Requirements for Internet Hosts Communication Layers*. RFC 1122. Internet Engineering Task Force, (1989); 10.17487/RFC1122
6. Cao, Y. et al. Off-path TCP exploits: Global rate limit considered dangerous. In *25th USENIX Security Symp. (USENIX Security 16)*. USENIX, (2016), 209–225.
7. Cao, Y. et al. Principled unearthing of TCP side channel vulnerabilities. In *Proceedings of the 2019 ACM SIGSAC Conf. on Computer and Communications Security*. ACM, (2019), 211–224.
8. Duke, M. et al. *A Roadmap for Transmission Control Protocol (TCP) Specification Documents*. RFC 7414. Internet Engineering Task Force, (2015); http://www.rfc-editor.org/rfc/rfc7414.txt
9. Feng, X. et al. Off-path TCP exploits of the mixed IPID assignment. In *Proceedings of the 2020 ACM SIGSAC Conf. on Computer and Communications Security*. ACM, (2020), 1323–1335.
10. Feng, X. et al. Off-path TCP hijacking attacks via the side channel of downgraded IPID. *IEEE/ACM Transactions on Networking 30*, 1 (2021), 409–422.
11. Feng, X. et al. Off-path network traffic manipulation via revitalized ICMP redirect attacks. In *31st USENIX Security Symp. (USENIX Security 22)*. USENIX, (2022), 2619–2636.
12. Feng, X. et al. PMTUD is not panacea: Revisiting IP fragmentation attacks against TCP. In *Network and Distributed System Security Symp. (NDSS)*. Internet Society, (2022).
13. Feng, X. et al. Man-in-the-middle attacks without rogue AP: When WPAs meet ICMP redirects. In *2023 IEEE Symp. on Security and Privacy (S&P)*. IEEE, (2022), 694–709.
14. Fiterau-Brostean, P. et al. Automata-based automated detection of state machine bugs in protocol implementations. In *Network and Distributed System Security Symp. (NDSS)*. Internet Society, (2023).
15. Flavel, A. et al. BGP route prediction within ISPs. *Computer Communications 33*, 10 (2010), 1180–1190.
16. Gilad, Y. and Herzberg, A. Fragmentation considered vulnerable: Blindly intercepting and discarding fragments. In *Proceedings of the 5th USENIX Conf. on Offensive Technologies*. USENIX, (2011).
17. Gilad, Y. and Herzberg, A. Off-path attacking the web. In *Proceedings of the 6th USENIX Conf. on Offensive Technologies*. USENIX, (2012), 41–52.
18. Gilad, Y. and Herzberg, A. Fragmentation considered vulnerable. *ACM Trans. on Information and System Security 15*, 4 (2013), 16.
19. Gilad, Y., Herzberg, A., and Shulman, H. Off-path hacking: The illusion of challenge-response authentication. *IEEE Security & Privacy 12*, 5 (2013), 68–77.
20. Gont, F. *ICMP Attacks against TCP*. RFC 5927. Internet Engineering Task Force, (2010); http://www.rfc-editor.org/rfc/rfc5927.txt
21. Iyengar, J. and Thomson, M. *QUIC: A UDP-Based Multiplexed and Secure Transport*. RFC 9000. Internet Engineering Task Force, (2021); http://www.rfc-editor.org/rfc/rfc9000.txt
22. Keyu, M., Zhou, X., and Qian, Z. DNS cache poisoning attack: resurrections with side channels. In *Proceedings of the 2021 ACM SIGSAC Conf. on Computer and Communications Security*. ACM, (2021), 3400–3414.
23. Klein, A. Cross layer attacks and how to use them (for DNS cache poisoning, device tracking and more). In *2021 IEEE Symp. on Security and Privacy (S&P)*. IEEE, (2021), 1179–1196.
24. Klein, A. Subverting stateful firewalls with protocol states. In *Network and Distributed System Security Symp. (NDSS)*. Internet Society, (2022).
25. Larsen, M.V. and Gont, F. *Recommendations for TransportProtocol Port Randomization*. RFC 6056. Internet Engineering Task Force, (2011); http://www.rfc-editor.org/rfc/rfc6056.txt
26. Lichtblau, F. et al. Detection, classification, and analysis of inter-domain traffic with spoofed source IP addresses. In *Proceedings of the 2017 Internet Measurement Conf.* ACM, (2017), 86–99.
27. Luckie, M. et al. Network hygiene, incentives, and regulation: Deployment of source address validation in the Internet. In *Proceedings of the 2019 ACM SIGSAC Conf. on Computer and Communications Security*. ACM, (2019), 465–480.
28. Man, K. et al. DNS cache poisoning attack reloaded: Revolutions with side channels. In *Proceedings of the 2020 ACM SIGSAC Conf. on Computer and Communications Security*. ACM, (2020), 1337–1350.
29. McCann, J. et al. *Path MTU Discovery for IP Version 6*. RFC 8201. Internet Engineering Task Force, (2017); http://www.rfc-editor.org/rfc/rfc8201.txt
30. Mirsky, Y. et al. VulChecker: Graph-based vulnerability localization in source code. In *32nd USENIX Security Symp.* USENIX, (2023).
31. Mogul, J. and Deering, S. *Path MTU Discovery*. RFC 1191. Internet Engineering Task Force, (1990); http://www.rfc-editor.org/rfc/rfc1191.txt
32. Nakibly, G. et al. Persistent OSPF attacks. In *Network and Distributed System Security Symp. (NDSS)*. Internet Society, (2012).
33. Pan, Y. and Rossow, C. TCP spoofing: Reliable payload transmission past the spoofed TCP handshake. In *2024 IEEE Symp. on Security and Privacy*. IEEE, (2024), 179–179.
34. Pokhrel, S.R. et al. TCP Performance over Wi-Fi: Joint Impact of Buffer and Channel Losses. *IEEE Trans. on Mobile Computing 15*, 5 (2016), 1279–1291; 10.1109/TMC.2015.2456883
35. Postel, J. *Internet Control Message Protocol*. RFC 792. Internet Engineering Task Force, (1981); 10.17487/RFC792
36. Qian, Z. and Mao, Z.M. Off-path TCP sequence number inference attack-how firewall middleboxes reduce security. In *2012 IEEE Symp. on Security and Privacy*. IEEE, (2012), 347–361.
37. Ramaiah, A., Stewart, R., and Dalal, M. *Improving TCP's Robustness to Blind In-Window Attacks*. RFC 5961. Internet Engineering Task Force, (2010); http://www.rfc-editor.org/rfc/rfc5961.txt
38. Rescorla, E. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. Internet Engineering Task Force, (2018); http://www.rfc-editor.org/rfc/rfc8446.txt
39. Thapa, C. et al. Transformer-based language models for software vulnerability detection. In *Proceedings of the 38th Annual Computer Security Applications Conf.* ACM, (2022), 481–496.
40. Touch, J., Mankin, A., and Bonica, R.P. *The TCP Authentication Option*. RFC 5925. Internet Engineering Task Force, (2010); http://www.rfc-editor.org/rfc/rfc5925.txt

Xuewei Feng is a research scientist at Tsinghua University, China. His research interests include network security and software vulnerability detection.

Qi Li is an associate professor with the Institute for Network Sciences and Cyberspace, Tsinghua University, China. His research interests include Internet and cloud security, IoT security, and AI security.

Kun Sun is a full professor at George Mason University. He serves as the director of the Sun Security Laboratory (SunLab) and the associate director of the Center for Secure Information Systems (CSIS).

Ke Xu (xuke@tsinghua.edu.cn) is a full professor at Tsinghua University, China. He has published over 200 technical papers in the research areas of next-generation Internet and network security.

Jianping Wu is a full professor at Tsinghua University, China. He has authored over 200 technical articles on the network architecture, high-performance routing and switching, protocol testing, and network security.

Watch the authors discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/off-path-attacks