Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 5 November 2026

K. Xu
Tsinghua University & Zhongguancun Laboratory
X. Feng
Tsinghua University
Q. Li
Tsinghua University & Zhongguancun Laboratory
Z. Li
Tsinghua University

4 May 2026

# Problem Statement for Cross-Layer Vulnerabilities due to Forged ICMP Errors

draft-xu-intarea-vulnerabilities-forged-icmp-02

## Abstract

ICMP error messages are vital for network reliability, providing feedback on issues such as unreachable hosts or fragmentation requirements. They help devices adapt dynamically, support troubleshooting, and enable essential functions like Path MTU Discovery. However, off-path attackers on the Internet may forge ICMP error messages to bypass legitimate validation mechanisms, causing the victim’s TCP/IP stack to misinterpret network conditions and exposing critical vulnerabilities. This document analyzes how such forged ICMP errors can be exploited by off-path attackers to induce cross-layer interactions within the victim’s TCP/IP stack, leading to four classes of vulnerabilities: information leakage, desynchronization of shared variables, semantic gaps, and identity deception. These ICMP-based attacks allow off-path attackers to manipulate network traffic, disrupt communication flows, and compromise both infrastructure and user privacy, without being on the direct communication path. The document concludes with proposed countermeasures and recommendations for protocol evolution.

## Status of This Memo

Table of Contents



# 1. Introduction

ICMP error messages constitute a fundamental component of the Internet control architecture[RFC792]. These messages serve as critical feedback mechanisms that inform endpoints and intermediate devices about various network conditions, including unreachable destinations, routing failures, and packet fragmentation requirements[RFC1122]. By enabling dynamic adjustments in response to network conditions, ICMP error messages contribute to maintaining the reliability, efficiency, and robustness of end-to-end communication. Without this essential feedback channel, numerous critical protocols and diagnostic tools, including Path MTU Discovery and traceroute, would be substantially impaired.

ICMP error messages present inherent validation challenges. This issue becomes particularly pronounced when the ICMP error contains a payload from a stateless protocol, such as UDP or ICMP. In such scenarios, the receiving host frequently lacks sufficient context to determine whether the error message corresponds to a legitimate packet it previously transmitted. Since UDP does not maintain per-flow state at the transport layer, the absence of session semantics renders it difficult to verify the authenticity and relevance of the ICMP error, thereby creating opportunities for potential exploitation.

The inherent difficulty in validating ICMP errors creates a significant attack vector for off-path adversaries. By forging ICMP error messages that appear to target legitimate traffic, adversaries can deceive recipients into misinterpreting network conditions. These forged messages can trigger complex cross-layer interactions within the TCP/IP stack -- particularly when they reference stateless protocols -- causing systems to react in unintended ways. Such manipulation can result in serious vulnerabilities, including information disclosure, denial of service, or traffic misdirection. Crucially, these attacks do not require the adversary to intercept or observe actual traffic, enabling covert exploitation of protocol semantics from remote Internet positions.

These ICMP-based attack vectors enable multiple exploitation scenarios:

1. Information disclosure, where adversaries observe system responses to ICMP error messages to infer internal state such as TCP sequence numbers or IP Identification (IPID) values;

2. State desynchronization, where ICMP messages create inconsistencies in shared variables such as MTU between protocol layers, resulting in fragmentation errors or packet drops;

3. Semantic validation gaps, where stateless protocols accept ICMP control messages without adequate validation mechanisms to verify the legitimacy of diverse protocol data;

4. Source authentication failures, where ICMP packets impersonate legitimate network devices without proper authentication, deceiving targets into accepting malicious routing or control information.

The effectiveness of these attacks stems from the implicit trust protocols place in cross-layer communications and the difficulty of validating control message authenticity across protocol boundaries[ACM2025TCPIP]. This document follows the problem statement methodology outlined in [RFC4336], which provides guidance for identifying and classifying protocol vulnerabilities in cross-layer interactions.

### 1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] and [RFC8174] (BCP 14) when, and only when, they appear in all capitals, as shown here.

## 2. Threat Model

This document focuses on vulnerabilities that can be exploited by off-path attackers-adversaries who are not positioned on the direct communication path between a client and a server. Unlike on-path attackers who can intercept, modify, or drop packets in transit, off-path attackers operate from external network locations and lack the ability to directly eavesdrop on or manipulate legitimate traffic flows. However, they retain the capability to inject spoofed packets into the network, making them a significant threat to protocol security.

The off-path threat model represents a realistic and prevalent attack scenario in modern networks. Off-path attackers can operate from anywhere on the Internet, including compromised hosts, botnets, or even legitimate network positions that are simply not on the target communication path. This positioning makes detection more challenging and expands the potential attack surface significantly compared to on-path attacks, which require the attacker to be strategically positioned between communicating endpoints.

Off-Path Threat model:

```
+---------+        +-------------+        +---------+
| Server  |--------+  Internet   +--------+ Client  |
+---------+        +------+------+        +---------+
                          |
                          | Spoofed ICMP
                          |
                      +----+----+
                      |Off-Path |
                      |Attacker |
                      +---------+
```

Figure 1: Off-Path Attack Model

In this threat model, the attacker leverages the ability to send spoofed IP packets-particularly ICMP messages-to trigger cross-layer vulnerabilities within the target’s network stack. By forging source IP addresses, the attacker can impersonate trusted entities such as routers, servers, or network infrastructure components. These spoofed packets appear legitimate at the network layer and can successfully traverse network paths to reach their intended targets.

The fundamental assumption underlying many protocol designs is that packets arriving from the network layer carry implicit trust regarding their origin and legitimacy. However, protocol implementations SHOULD NOT blindly trust such packets. This assumption becomes a critical vulnerability in the off-path attack model, where malicious packets can be crafted to exploit cross-layer interactions without requiring the attacker to compromise the direct communication path[RFC5927]. The resulting attacks can violate protocol semantics, disrupt ongoing communications, or manipulate system behavior while remaining difficult to detect and attribute.

## 3. Problem Statement

Four distinct classes of vulnerabilities -- Cross-Layer Information Disclosure, Cross-Layer State Desynchronization, Cross-Layer Semantic Validation Deficiencies, and Cross-Layer Source Authentication Failures -- can emerge from cross-layer interactions within the TCP/IP protocol suite. These vulnerabilities stem from fundamental architectural assumptions, shared state management deficiencies, and the absence of robust cross-layer validation mechanisms. Protocol designers SHOULD carefully evaluate cross-layer interactions to minimize such risks.

### 3.1. Cross-Layer Information Disclosure

This vulnerability arises when observable fields in one protocol layer expose information that is semantically or cryptographically bound to another layer. Specifically, when a protocol assigns values to a field based on internal or high-layer state-such as counters or identifiers-that field may serve as an unintentional side channel. If this field is externally observable, an entity without access to internal state can correlate changes in its value to infer sensitive information, undermining the isolation between protocol layers.

The underlying cause of this vulnerability lies in the lack of entropy separation across protocol layers. When the state generation logic of a lower-layer protocol is influenced by, or derived from, upper-layer state-either directly or indirectly-observable behavior at the lower layer may unintentionally reveal sensitive upper-layer information. This creates a channel through which confidential state can be inferred by external observers. Furthermore, if protocol behavior permits external stimuli (such as control-plane messages) to affect the internal assignment policies of protocol fields, the risk of information leakage is significantly amplified. Fields originally designed for operational purposes at the network layer may, under such conditions, become conduits for exposing transport-layer state, thereby undermining confidentiality and weakening protocol-layer security guarantees such as sequence number randomization[RFC4086].

A notable instance of this phenomenon is the coupling between the IP Identification (IPID) field and the TCP sequence number space. Although the IPID field was originally introduced to support IP-layer fragmentation and reassembly, certain implementations assign its value in ways that are indirectly influenced by active TCP session states. In some systems, off-path attackers can manipulate this assignment logic-e.g., by sending crafted ICMP errors-to trigger changes in the IPID generation behavior. By observing variations in IPID values, attackers can infer whether their injected TCP packets contain correct or incorrect sequence numbers, thereby learning the valid sequence number and enabling off-path TCP session hijacking [CCS2020IPID].

```
       Attacker                Victim Server              Victim Client
       --------                -------------              -------------
          |                         |                         |
          |---(1) ICMP "Frag.------->|  (Action: Clear DF      |
          |      Needed"            |   Flag, Use Hash-based  |
          |      (src=attacker)     |   IPID)                 |
          |                         |                         |
          |---(2) Probe Request---->|                         |
          |<--(3) Probe Reply-------|                         |
          |      (IPID=x)           |                         |
          |                         |                         |
          |                         |<--------(4) SYN---------|
          |                         |                         |
          |                         |---(5) SYN/ACK---------->|
          |                         |      (IPID=x+1)         |
          |                         |   (Shared IPID          |
          |                         |    increments)          |
          |                         |                         |
          |---(6) Probe Request---->|                         |
          |<--(7) Probe Reply-------|                         |
          |      (IPID=x+2)         |                         |
          | (Attacker observes      |                         |
          |  IPID jump)             |                         |
```

Figure 2: IPID-Based Connection Inference Attack

## 3.2. Cross-Layer State Desynchronization

Protocols within the same stack frequently operate on shared global variables, such as metrics related to path properties, endpoint capabilities, or buffer dimensions. When these variables are updated asynchronously or within layer-specific contexts, state inconsistencies may arise. These shared variables are often used by multiple layers to make decisions, leading to potential conflicts or discrepancies if updates are not properly synchronized. State desynchronization occurs when one layer updates a shared variable in response to a specific event, while other dependent layers are either unaware of the update or unable to reflect the change immediately.

This type of vulnerability is characterized by temporal or contextual divergence in the interpretation of shared data. Such divergence can occur when network or transport layers rely on outdated or inconsistent information due to the asynchronous nature of updates across layers. For instance, a network-layer parameter such as Maximum Transmission Unit (MTU), which is updated in response to a control-plane input or a network change, may not immediately be propagated to the transport layer. As a result, the transport layer may continue to operate under the assumption of an outdated MTU, which can lead to improper handling of data fragments or errors in packet transmission.

A concrete example of this vulnerability can be seen in the interaction between the TCP and IP layers concerning Path MTU Discovery (PMTUD). When an attacker sends a forged ICMP fragmentation-needed packet to manipulate the Path MTU (PMTU), the newly updated PMTU may not be immediately propagated to the transport layer, which continues to operate under the assumption of the previous, higher MTU value. This desynchronization can cause TCP to generate packets that exceed the updated MTU, resulting in unintended fragmentation at points where fragmentation is prohibited, or packet drops in environments where fragmentation is not supported.

Such cross-layer inconsistencies disrupt data transmission, violate TCP’s non-fragmentation assumptions, and introduce operational errors including communication delays and packet loss. Implementations SHOULD ensure that updates to shared variables are properly synchronized across protocol layers. More critically, this vulnerability enables attackers to exploit IP fragmentation mechanisms to inject malicious packets and potentially hijack TCP connections [NDSS2022MTU]. The lack of proper synchronization between layers in handling shared path properties like MTU creates significant security vulnerabilities within the protocol stack, exposing systems to both denial-of-service and session hijacking attacks.

```
      Attacker                Victim Server          Victim Client
      --------                -------------          -------------
         |                         |                          |
         |---(1) Infer TCP-------->|                          |
         |      Connection         |                          |
         |                         |                          |
         |---(2) Trigger IP------->|                          |
         |      Fragmentation      |                          |
         |                         |                          |
         |-----------(3) Forge & Inject IP Fragment--------->|
         |                         |                          |
         |                         |<--(4) Legitimate--------|
         |                         |      IP Fragments        |
         |                         |                          |
         |                         |   (5) Mis-reassemble     |
         |                         |       Packet             |
         |                         |   (Forged + Legitimate) |
         |                         |                          |
```

Figure 3: IP Fragment Injection Attack

### 3.3.  Cross-Layer Semantic Validation Deficiencies

Protocols often depend on implicit assumptions about the structure and statefulness of adjacent protocol layers.  When a protocol receives cross-layer input containing data attributed to another protocol, it may attempt validation based on limited semantic knowledge or incomplete contextual information.  If the incoming data originates from a stateless or loosely specified layer, or lacks integrity guarantees, the receiving protocol faces significant challenges in determining data legitimacy.  This fundamental limitation creates vulnerabilities when protocol validation processes prove inadequate for ensuring the authenticity and integrity of cross-layer communications.

This semantic validation gap becomes particularly exploitable when protocols rely on partial data representations, such as fixed-length headers or truncated payloads, for legitimacy assessment.  Protocols often implement basic checksums or header-based validation mechanisms without considering the full operational context or semantic meaning of the data.  This creates opportunities for attackers to craft malicious packets that conform syntactically to expected formats while semantically violating intended operational contexts.  Such carefully crafted malformed packets can trigger unintended state transitions, erroneous control decisions, or inconsistent processing behaviors that diverge from correct protocol logic.

A concrete illustration of this vulnerability emerges in the handling of forged ICMP redirect messages targeting stateless protocols such as UDP.  ICMP redirect messages are legitimate network control packets used by routers to inform hosts about more optimal routing paths.  However, stateless protocols like UDP cannot maintain session state or establish trust relationships for their connections, making direct validation of ICMP control messages impossible.  Attackers exploit this validation gap by crafting and injecting malicious ICMP redirect messages with spoofed source addresses, deceiving target hosts into redirecting their traffic through attacker-controlled gateways [USENIXSECURITY2023ICMP].  This enables sophisticated man-in-the-middle attacks where adversaries can intercept, modify, or redirect network traffic without being positioned on the original communication path.  The fundamental vulnerability stems from the absence of stateful correlation mechanisms and authenticity validation across protocol layer boundaries, allowing forged control messages to manipulate legitimate network behavior.

```
       Attacker      Neighbor Host      Victim          Destination
       --------      -------------      ------          -----------
          |                |               |                  |
          |--(1) Detects-->|               |                  |
          |    Neighbor    |               |                  |
          |    Host        |               |                  |
          |                |               |                  |
          |-------(2) Forge UDP Datagrams-------------------->|
          |                |               |                  |
          |                |               |---(3) Establish->|
          |                |               |   UDP Socket     |
          |                |               |                  |
          |-------(4) Forge ICMP Redirect-------------------->|
          |                |               |                  |
          |                |        (5) Check Passed          |
          |                |               |                  |
          |                |<-----(6) Redirected Traffic------|
          |                |               |                  |
          |                |--(7) Discard--|                  |
          |                |    Traffic X  |                  |
          |                |               |                  |
          |                |               |      (8)   No Response
          |                |               |            Received
```

Figure 4: ICMP Redirect Traffic Hijacking Attack

### 3.4.  Cross-Layer Source Authentication Failures

Cross-layer interactions often rely on implicit trust assumptions regarding the provenance and authenticity of received data, particularly when control messages or routing information traverses protocol layer boundaries.  Protocol implementations typically operate under the assumption that data originating from lower layers-including control messages, routing updates, and error notifications-carries inherent legitimacy, provided it conforms to expected syntactic formats and protocol specifications.  However, this fundamental trust assumption creates a critical vulnerability when protocols accept cross-layer input without implementing robust mechanisms to verify the authentic origin or validate the association with established communication contexts.  Under such circumstances, protocols become susceptible to identity deception attacks, where malicious entities can successfully impersonate legitimate network components or communication peers.

This vulnerability emerges from the systematic absence of cryptographic authentication frameworks and contextual validation mechanisms that would otherwise establish secure bindings between data sources and trusted communication relationships.  The lack of comprehensive identity verification enables malicious actors to exploit these authentication gaps by injecting carefully crafted ICMP packets and other spoofed control messages into legitimate communication flows, effectively masquerading as authoritative network infrastructure components such as routers, gateways, or access points.  The severity of this vulnerability is further amplified when underlying network infrastructure-including forwarding engines, hardware accelerators, and intermediate processing nodes-fails to enforce stringent access control policies or implement comprehensive provenance verification before relaying control messages to higher protocol layers.  Consequently, maliciously crafted ICMP packets can propagate through the network stack unchecked, systematically undermining the integrity and trustworthiness of the entire communication system.

The propagation of these unverified ICMP control messages can trigger cascading security failures, including unauthorized reconfiguration of forwarding behaviors, systematic disruption of established communication flows, illegitimate privilege escalation within protocol stacks, and ultimately comprehensive compromise of network reliability and security.  These attacks leverage the inherent trust protocols place in ICMP control messages, exploiting the fundamental assumption that such messages originate from legitimate network infrastructure.

A concrete manifestation of this vulnerability can be observed in sophisticated attacks against Wi-Fi networks, where adversaries deploy malicious terminals to impersonate legitimate Access Points (APs) while simultaneously injecting forged ICMP redirect messages. In this attack scenario, the malicious entity exploits the implicit trust that Wi-Fi-enabled devices place in both wireless control frames and ICMP network control messages.  By strategically crafting and transmitting spoofed ICMP redirect packets alongside fraudulent wireless association messages, attackers can systematically deceive target devices into redirecting their network traffic through attacker-controlled infrastructure.  This multi-vector approach enables sophisticated man-in-the-middle attacks that combine wireless protocol exploitation with ICMP-based traffic manipulation, allowing adversaries to intercept, modify, or redirect victim communications while maintaining the appearance of legitimate network operation [SP2023MITM].  The effectiveness of these attacks stems from the fundamental vulnerability in cross-layer trust assumptions and the absence of robust authentication mechanisms for verifying the identity of ICMP message sources.

```
       Attacker      Victim Supplicant       AP           Server
       --------      -----------------       --           ------
          |                 |                |              |
          |                 |                |              |
          |-(1.1) Ping----->|                |              |
          |  (src=Server)   |                |              |
          |                 |-(1.2) Ping---->|------------->|
          |                 |    Reply       |              |
          |                 | (Action: Create|              |
          |                 |  Routing Cache)|              |
          |                 |                |              |
          |-(1.3) UDP Port->|                |              |
          |    Probe        |                |              |
          |<-(1.4) ICMP-----|                |              |
          |    Port         |                |              |
          |    Unreachable  |                |              |
          | (Attacker finds |                |              |
          |  open port)     |                |              |
          |                 |                |              |
          |                 |                |              |
          |-(2.1) ICMP----->|                |              |
          |    Redirect     |                |              |
          | (src=AP, with   |                |              |
          |  forged UDP for |                |              |
          |  check evasion) |                |              |
          |                 |                |              |
          |         (2.2) Action: Routing Cache is Poisoned,|
          |               Attacker becomes the next-hop.    |
          |                 |                |              |
          |                 |                |              |
          |<-(3.1) Redirect-|                |              |
          |    Traffic      |                |              |
          | ("I like you,   |                |              |
          |  server Bob!")  |                |              |
          |                 |                |              |
          |-(3.2) Falsified>|------------------------------>|
          |    Traffic      |                |              |
          | ("I hate you,   |                |              |
          |  server Bob!")  |                |              |
          |                 |                |              |
```

Figure 5: Wireless Network Routing Cache Poisoning

## 4. Mitigation Directions

The mitigation directions in this section are intentionally framed as design objectives rather than a dependency on any single protocol extension. Concrete challenge-confirm mechanisms are being explored elsewhere, but those proposals are still evolving. This document therefore focuses on properties that robust future solutions should provide, regardless of the exact encoding or signaling method.

First, implementations should strengthen contextual validation before acting on ICMP-driven state changes. At a minimum, an ICMP message should be correlated with an existing communication context, and the quoted packet should be checked against the transport-layer and routing information available to the receiver. For TCP, this includes robustness improvements of the kind discussed in [RFC5927]. For Redirect processing, hosts should continue to apply the strict acceptance checks described in [RFC1122] instead of treating a syntactically valid message as sufficient proof of legitimacy.

Second, mitigation mechanisms should avoid turning forged ICMP traffic into an attacker-controlled state-allocation problem. A useful design principle is to create no per-message state for ICMP errors that do not match an active flow, and to limit any state added for matched flows to minimal or bounded metadata. Likewise, receipt of an unverified ICMP message should not by itself trigger the immediate transmission of a new packet, because that can create reflection, amplification, or state-exhaustion risks. Stateless or bounded-state validation techniques, conceptually similar to SYN-cookie-style approaches [RFC4987], are preferable to designs that cache challenge state for every unverified message.

Third, protocol stacks should reduce reliance on unauthenticated ICMP input for path adaptation when more robust confirmation techniques are available. In particular, PMTU-related adjustments should be designed so that unauthenticated Packet Too Big or fragmentation-needed messages are treated as hints that can be confirmed through Packetization Layer PMTU Discovery or Datagram PLPMTUD techniques [RFC4821] [RFC8899] before irreversible changes are made. This does not eliminate the utility of ICMP, but it reduces the impact of forged control messages on shared path state.

Finally, implementations should ensure that cross-layer reactions are synchronized, rate-limited, and scoped. Updates to shared variables such as PMTU, routing cache entries, or packetization behavior should be propagated consistently across layers, and security-sensitive transitions should be guarded so that a single unauthenticated control message cannot immediately trigger persistent or system-wide reconfiguration. The goal is not to reject all ICMP input, but to ensure that any accepted signal is both contextually plausible and operationally bounded.

## 5. IANA Considerations

This memo includes no request to IANA.

## 6. Security Considerations

This document identifies security vulnerabilities in cross-layer interactions within the TCP/IP protocol suite. The vulnerabilities described-cross-layer information disclosure, cross-layer state desynchronization, cross-layer semantic validation deficiencies, and cross-layer source authentication failures-represent significant threats to network security that require careful consideration in protocol design and implementation.

The security implications of these vulnerabilities extend beyond individual protocol layers to affect the overall integrity and trustworthiness of network communications. In particular, defenses should be evaluated not only for their ability to reject forged control messages, but also for whether they avoid creating new reflection, amplification, or state-exhaustion risks while doing so. Implementers and protocol designers should consider the mitigation directions outlined in this document when developing new protocols or updating existing ones.

## 7. References

### 7.1. Normative References

[RFC792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/rfc/rfc792>.

[RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, DOI 10.17487/RFC5927, July 2010, <https://www.rfc-editor.org/rfc/rfc5927>.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <https://www.rfc-editor.org/rfc/rfc1122>.

[RFC4336] Floyd, S., Handley, M., and E. Kohler, "Problem Statement for the Datagram Congestion Control Protocol (DCCP)", RFC 4336, DOI 10.17487/RFC4336, March 2006, <https://www.rfc-editor.org/rfc/rfc4336>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

## 7.2. Informative References

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/rfc/rfc4086>.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <https://www.rfc-editor.org/rfc/rfc4821>.

[RFC4987] Eddy, W., "TCP SYN Flooding Attacks and Common Mitigations", RFC 4987, DOI 10.17487/RFC4987, August 2007, <https://www.rfc-editor.org/rfc/rfc4987>.

[RFC8899] Fairhurst, G., Jones, T., T端xen, M., R端ngeler, I., and T. V旦lker, "Packetization Layer Path MTU Discovery for Datagram Transports", RFC 8899, DOI 10.17487/RFC8899, September 2020, <https://www.rfc-editor.org/rfc/rfc8899>.

[ACM2025TCPIP] Feng, X., Li, Q., Sun, K., Xu, K., and J. Wu, "Exploiting Cross-Layer Vulnerabilities: Off-Path Attacks on the TCP/IP Protocol Suite", February 2025, <https://cacm.acm.org/research/exploiting-cross-layer-vulnerabilities-off-path-attacks-on-the-tcp-ip-protocol-suite/>.

[CCS2020IPID]
Feng, X., Fu, C., Li, Q., Sun, K., and K. Xu, "Off-path TCP exploits of the mixed IPID assignment", October 2020, <https://dl.acm.org/doi/abs/10.1145/3372297.3417884>.

[NDSS2022MTU]
Feng, X., Li, Q., Sun, K., Xu, K., Liu, B., Zheng, X., Yang, Q., Duan, H., and Z. Qian, "PMTUD is not Panacea: Revisiting IP Fragmentation Attacks against TCP", April 2022, <https://www.ndss-symposium.org/ndss-paper/auto-draft-185/>.

[SP2023MITM]
Feng, X., Li, Q., Sun, K., Yang, Y., and K. Xu, "Man-in-the-middle attacks without rogue AP: when WPAs meet ICMP redirects", May 2023, <https://ieeexplore.ieee.org/document/10179441>.

[USENIXSECURITY2023ICMP]
Feng, X., Li, Q., Sun, K., Qian, Z., Zhao, G., Kuang, X., Fu, C., and K. Xu, "Off-Path Network Traffic Manipulation via Revitalized ICMP Redirect Attacks", August 2022, <https://www.usenix.org/conference/usenixsecurity22/presentation/feng>.

## Acknowledgements

The authors would like to thank the IETF community, particularly members of the ICMP and Security Working Groups, for their valuable feedback and insights during the development of this proposal. Special thanks to the contributors who provided research findings that form the foundation of this analysis.

## Authors’ Addresses

Ke Xu
Tsinghua University & Zhongguancun Laboratory
Email: xuke@tsinghua.edu.cn

Xuewei Feng
Tsinghua University
Email: fengxw06@126.com

Qi Li
Tsinghua University & Zhongguancun Laboratory
Email: qli01@tsinghua.edu.cn

Zhaoxi Li
Tsinghua University
Email: li-zx24@mails.tsinghua.edu.cn