

Internet Area Working Group
Internet-Draft
Intended status: Informational
Expires: 29 October 2026

K. Xu
Tsinghua University & Zhongguancun Laboratory
X. Feng
Y. Yang
Tsinghua University
Q. Li
Tsinghua University & Zhongguancun Laboratory
27 April 2026

Enhancing ICMP Error Message Authentication Using Challenge-Confirm
Mechanism
draft-xu-intarea-challenge-icmpv4-03

Abstract

The Internet Control Message Protocol (ICMP) is essential for network diagnostics but is vulnerable to off-path spoofing attacks, especially when error messages relate to stateless transport protocols like UDP. An attacker can forge these messages to degrade performance or enable Man-in-the-Middle attacks.

This document proposes a robust, stateless challenge-response mechanism to authenticate ICMP error messages. Traditional stateful challenge mechanisms are vulnerable to state-exhaustion Denial-of-Service (DoS) attacks. To avoid this, the proposed solution is inspired by TCP SYN-Cookies, eliminating the need to store per-challenge state by using cryptographic computation. It limits state management to minimal flags on existing sockets or a bounded probabilistic data structure. This approach effectively authenticates ICMP error messages while inherently resisting both off-path spoofing and state-exhaustion DoS attacks, thus improving the robustness of ICMP.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 29 October 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Terminology	3
3. Problem Statement	4
3.1. Source-Based Blocking Ineffectiveness	4
3.2. Authentication Evasion based on Embedded Packet State . .	4
3.2.1. Stateful Embedded Packets (e.g., TCP)	4
3.2.2. Stateless Embedded Packets (e.g., UDP, ICMP)	4
4. Stateless Challenge-Confirm Mechanism	5
4.1. Core Principle: Eliminating State with Cryptographic Computation	5
4.2. Challenge-Confirm Procedure	5
4.3. Protocol-Specific State Management	7
4.4. Challenge-Confirm Option	8
5. Exception Handling and Edge Cases	9
5.1. Packet Loss	9
5.2. Multi-Path Routing Scenarios	10
6. Security Considerations	10
7. IANA Considerations	11
8. References	12
8.1. Normative References	12
8.2. Informative References	12
Acknowledgments	13
Authors' Addresses	13

1. Introduction

The Internet Control Message Protocol (ICMP) [RFC792] is an integral part of network operations, providing essential feedback on transmission issues. However, the legitimate verification of ICMP error messages is inherently vulnerable by design. To enable senders to correlate error reports with the original packets for effective network diagnostics, ICMP error messages, as specified in [RFC792], MUST include the header information and a portion of the payload of the original message that triggered the error. When the original message originates from stateless protocols like UDP or ICMP, the embedded original message header lacks contextual information (e.g., sequence numbers, acknowledgement numbers, and ports in stateful protocols like TCP). This makes it difficult for the receiver to effectively verify the legitimacy of the error messages. Consequently, attackers can forge ICMP error messages embedded with stateless protocol payloads to bypass the legitimate verification of the receiver, tricking the receiver into erroneously accepting and responding to the message, which can lead to malicious activities.

For example, off-path attackers can forge ICMP "Fragmentation Needed" messages to degrade throughput and harm latency-sensitive applications. This can also induce TCP segment fragmentation [NDSS2022MTU] and enable IP ID-based TCP session hijacking [CCS2020IPID]. Moreover, forged ICMP Redirect messages embedded with stateless protocol data can be used to trick victims into modifying their routing, facilitating off-path traffic interception, modification, and credential theft [USENIXSECURITY2023ICMP], [SP2023MITM]. These diverse attack vectors starkly underscore the critical and urgent necessity for robust authentication mechanisms in ICMP for error message processing.

This document proposes a stateless challenge-confirm mechanism that authenticates these ICMP error messages. The mechanism is designed to prove that the source of an error is on the path of the associated data flow, thwarting off-path attackers without introducing new Denial-of-Service vulnerabilities.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. TCP terminology should be interpreted as described in [RFC9293].

3. Problem Statement

Current ICMP specifications have inherent limitations that allow off-path attackers to forge ICMP error messages, undermining network security and reliability. The primary issues are:

3.1. Source-Based Blocking Ineffectiveness

Certain ICMP error messages, such as "Fragmentation Needed" messages, can originate from any intermediate router along the packet's path. This ubiquity makes source-based blocking ineffective, as legitimate messages can come from multiple sources.

3.2. Authentication Evasion based on Embedded Packet State

Although [RFC792] and [RFC1122] stipulate that error messages should include as much of the original (offending) packet as possible, the inherent characteristics of the embedded packet protocol directly influence the difficulty of authenticating ICMP error messages and their overall security strength.

3.2.1. Stateful Embedded Packets (e.g., TCP)

When attackers embed stateful protocol packets, such as TCP segments, in forged ICMP error messages, receivers can theoretically utilize the inherent state information of the TCP protocol for a certain degree of verification. The TCP protocol establishes and maintains state between communicating parties through sequence numbers, acknowledgment numbers, and ports. These connection-based TCP state information are difficult for attackers to accurately guess. Receivers can attempt to verify whether these connection-specific details in the embedded TCP header match their maintained TCP connection state, thereby judging the authenticity of the ICMP error message.

3.2.2. Stateless Embedded Packets (e.g., UDP, ICMP)

In contrast to stateful TCP, when attackers embed stateless protocol packets, such as UDP or ICMP messages, in forged ICMP error messages, receivers lose the ability to perform effective state verification. UDP and ICMP protocols are inherently designed as stateless protocols. The UDP or ICMP messages embedded in ICMP error messages contain almost no state information that can be used for context verification. In addition to performing some basic protocol format checks, receivers have virtually no way to determine the authenticity of ICMP error messages based on the embedded stateless packet header. This lack of state verification greatly reduces the authentication strength of ICMP error messages, making it easier for attackers to

implement authentication evasion and use forged error messages for malicious attacks.

4. Stateless Challenge-Confirm Mechanism

A simple stateful challenge-response mechanism, where a host stores a nonce while waiting for a confirmation, would introduce a critical state-exhaustion Denial-of-Service (DoS) vulnerability. An attacker could flood a target with forged error messages, forcing it to allocate state for each one. To solve this, the mechanism proposed here is stateless and inspired by TCP SYN-Cookies [RFC4987], where state is not stored but is instead encoded cryptographically and later re-computed for validation.

4.1. Core Principle: Eliminating State with Cryptographic Computation

Instead of generating and storing a random nonce, the host computes a deterministic nonce on demand. This nonce is a cryptographic hash of information that defines the flow, combined with a secret key known only to the host.

Challenge Nonce = F(secret_key, src_IP, dest_IP, [other_flow_info])

- * secret_key: A high-entropy secret value held by the host's operating system. This key MUST be rotated periodically (e.g., every few minutes) to limit the impact of any potential key compromise and to mitigate replay attacks.
- * F: A keyed-hash function, such as HMAC-SHA256, truncated to the size of the nonce field.

With this approach, a nonce can be generated when needed (for an outgoing challenge) and verified later (on a returning confirmation) by simply re-computing it. There is no need to store it in a cache.

4.2. Challenge-Confirm Procedure

The stateless process is as follows:

- * Receive and Validate Error: Host A receives an ICMP error message. It first validates the embedded header's 4-tuple against its list of active sockets/connections. If no matching socket exists, the message is silently discarded. No state is created.
- * Mark Flow for Challenge: If a matching socket is found, Host A does not create new state. Instead, it sets a simple flag on the existing socket control block, marking it as "requires challenge". The initial ICMP error is then discarded.

- * **Issue Computed Challenge:** The next time the application sends a packet on this marked socket, the networking stack intercepts it. It computes the challenge nonce using the secret key and the packet's flow information. This nonce is placed in a Challenge-Confirm IP Option, and the packet is sent.
- * **Receive and Verify Confirmation:** If a legitimate on-path node returns a new ICMP error, it will contain the challenge packet. Host A receives this new error, extracts the embedded nonce, and recomputes the expected nonce using the same secret key and flow information.
- * **Process or Discard:** If the received nonce matches the re-computed one, the error is authentic, and Host A can act on it. If it does not match, the message is a forgery or is stale, and it is discarded.

This flow achieves the anti-spoofing goal without creating state for unverified messages, thus defeating potential DoS attacks. Figure 1 illustrates the complete interaction.

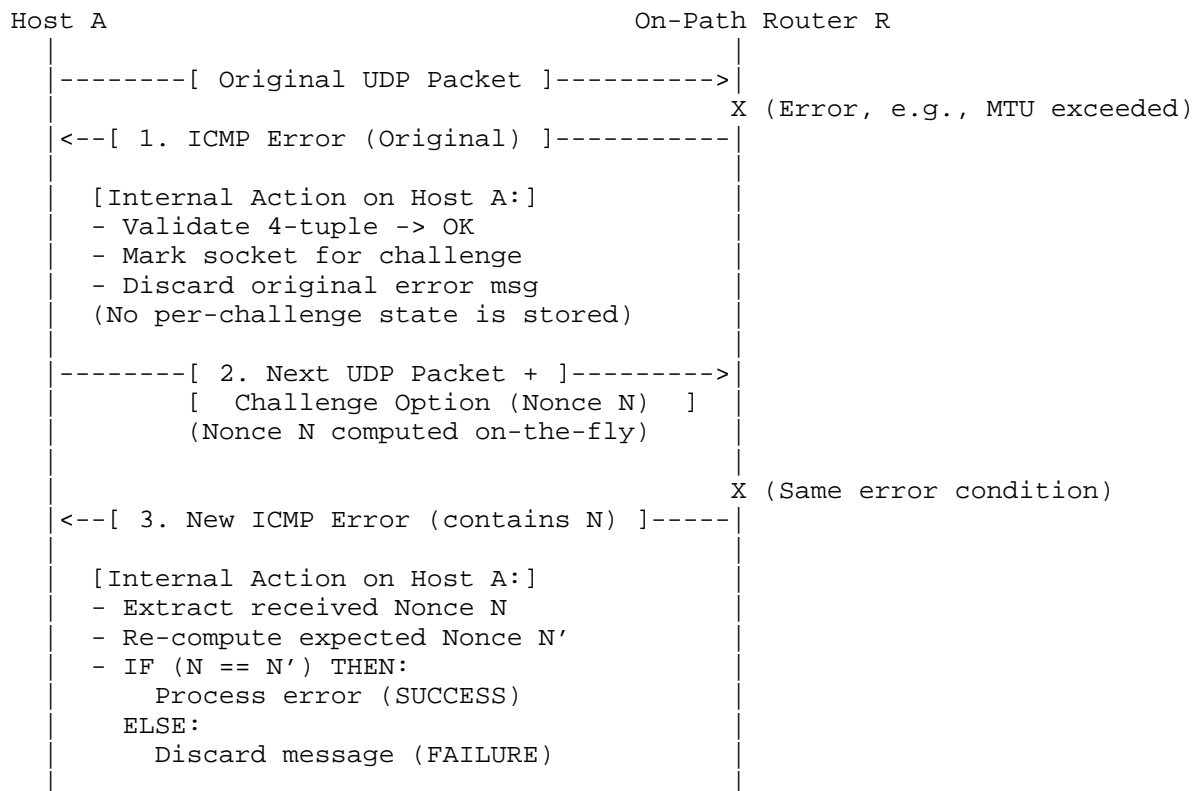


Figure 1: Challenge-Confirm Mechanism

4.3. Protocol-Specific State Management

The mechanism for "marking a flow" is lightweight and transport-specific.

- * ***UDP***: Upon receiving a validatable ICMP error, the host sets a flag on the corresponding UDP socket's control block.
- * ***TCP***: While TCP has its own protections, this mechanism can supplement it. A flag can be set on the TCB.
- * ***ICMP***: For connectionless protocols like ICMP Echo, which lack a socket state, a probabilistic, fixed-size data structure like a Sketch or Bloom Filter SHOULD be used.
 - On Error Reception: The host hashes a flow identifier (e.g., source IP, destination IP, ICMP Identifier) and increments the corresponding counter(s) in the sketch.

- On Packet Transmission: When sending a new ICMP packet, the host queries the sketch. If the query indicates this flow has likely received a recent error, it attaches the computed challenge. This probabilistic approach ensures that state remains bounded, preventing DoS attacks against ICMP-based applications.

4.4. Challenge-Confirm Option

To support the Challenge-Confirm mechanism, this document defines a new Challenge-Confirm IP Option. The challenge packet for a received ICMP error message containing a stateless protocol payload includes this option in the IP header (as shown in Figure 2). Similarly, the ICMP error message triggered in response to this challenge packet should also include the same option in the header of the embedded IP challenge packet (as shown in Figure 3).

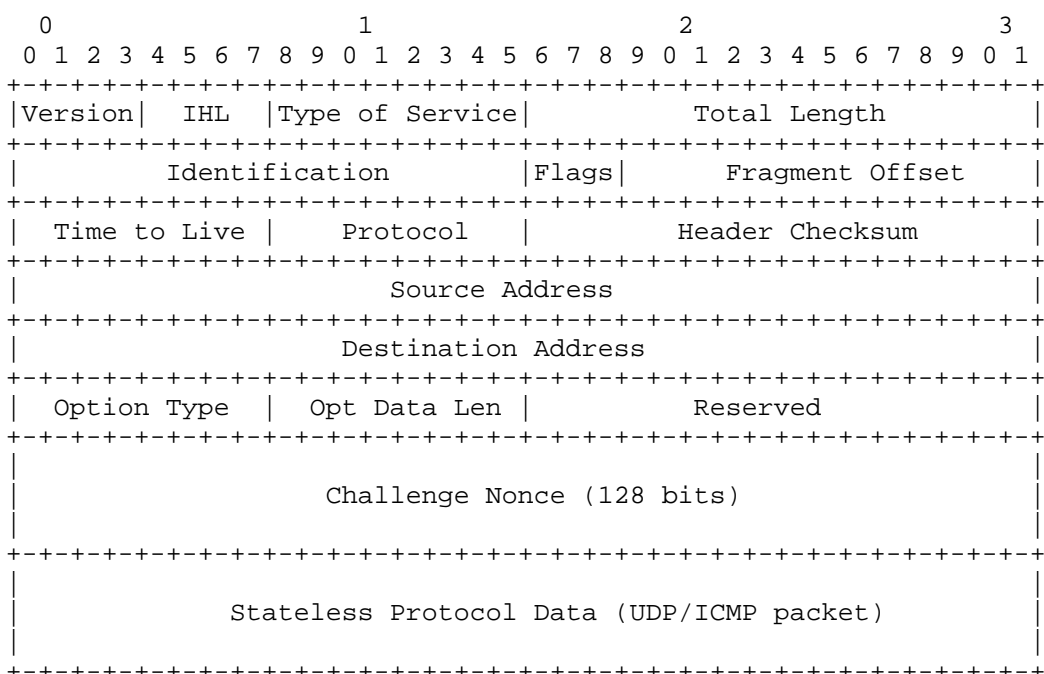


Figure 2: The IPv4 Challenge Packet with Challenge-Confirm Option

The fields in the Challenge-Confirm Option are defined as follows:

- * *Option Type*: 8-bit identifier for the challenge-confirm option. The final value requires IANA assignment.

- * *Opt Data Len*: 8-bit unsigned integer specifying the length of the option data field in bytes.
- * *Reserved*: 16-bit field reserved for future use. MUST be set to zero on transmission and ignored on reception.
- * *Challenge Nonce*: 128-bit number computed as described in Section 4.1.

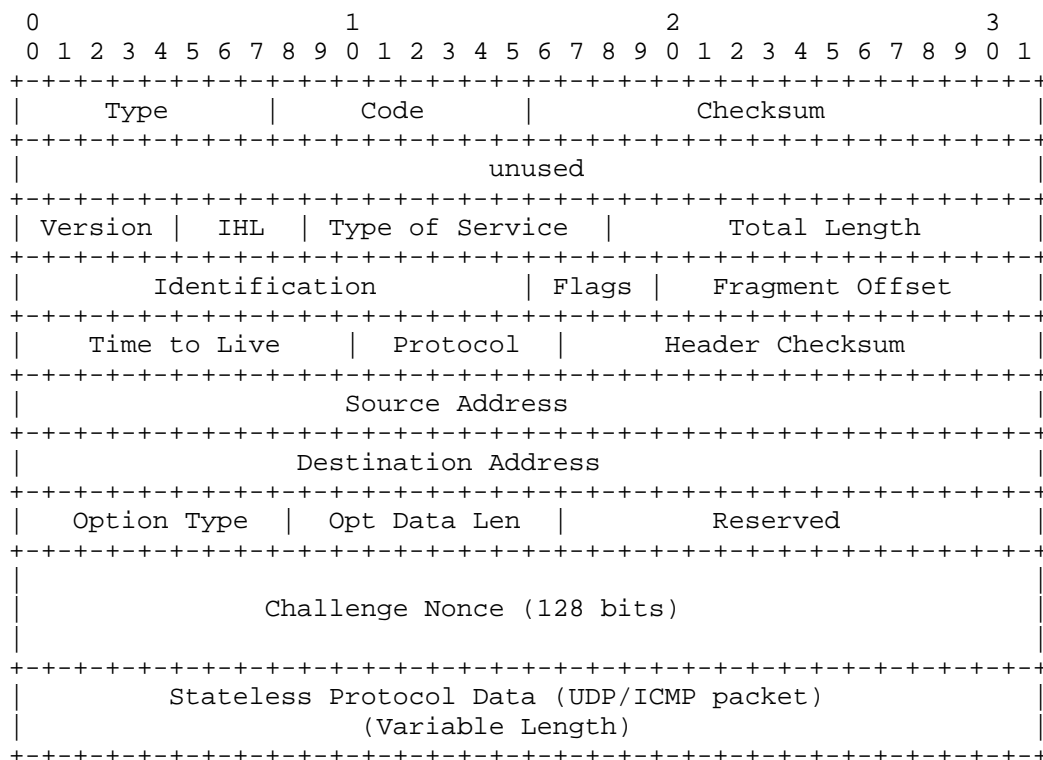


Figure 3: New ICMP Error Responding to the Challenge Packet

5. Exception Handling and Edge Cases

5.1. Packet Loss

The proposed mechanism is inherently resilient to packet loss due to its stateless design. It does not maintain timers or retransmission states for the challenge-confirm exchange itself. The requires challenge flag is cleared as soon as the challenge packet is transmitted, meaning the host does not enter a state of "waiting for a confirmation".

Whether the outgoing challenge packet or the returning ICMP confirmation is lost in transit, the outcome is the same: the host that issued the challenge does not receive a confirmation and takes no special action. The exchange silently fails.

Recovery is not driven by a timer, but by the persistence of the underlying network issue. If the condition that caused the initial ICMP error persists, a subsequent data packet from the application will likely trigger a new, initial ICMP error, naturally restarting the challenge process. This "fire-and-forget" approach avoids adding stateful complexity for the challenge itself.

5.2. Multi-Path Routing Scenarios

The mechanism's performance, but not its security, can be affected in networks that employ per-packet load balancing across multiple paths. Consider a scenario where a flow's packets alternate between a "bad" path that triggers an ICMP error and a "good" path that does not.

A recurring cycle could emerge: 1. A data packet is routed to the "bad" path, triggering an initial ICMP error and causing the host to set the requires challenge flag. 2. The next packet (now a challenge packet) is routed to the "good" path and reaches its destination successfully. No ICMP confirmation is returned. 3. The host, having sent its challenge, clears the flag. The next data packet is a normal packet, which is again routed to the "bad" path, restarting the cycle.

This cycle does not compromise the security of the mechanism. The host never acts on an unvalidated ICMP error, so spoofing attacks remain ineffective. However, it creates a performance degradation. In this specific scenario, the effective throughput for the flow could be halved. This is a performance cost in certain network topologies, not a security vulnerability.

6. Security Considerations

The proposed enhancements aim to bolster ICMP security by addressing specific vulnerabilities related to message authentication. Key security aspects include:

- * ***Authentication Strength***: The security of the authentication depends on the unguessability of the computed nonce, which is guaranteed by the use of a strong keyed-hash function and a secret key with sufficient entropy [RFC4086].

- * ***Denial of Service (DoS) Resistance***: This is the principal security advantage over stateful designs. The mechanism is resilient to state-exhaustion attacks because:
 1. It creates no state for ICMP errors that do not correspond to an existing, active transport-layer socket.
 2. For valid flows, the state added is minimal (a flag) or probabilistically bounded (a sketch), preventing uncontrolled resource consumption.
- * ***Replay Attack Mitigation***: The periodic rotation of the `secret_key` provides the primary defense against replay attacks. A captured nonce-confirmation pair will become invalid after the key is changed. The rotation interval presents a trade-off between security and the maximum legitimate round-trip time for a challenge-confirm exchange.
- * ***Reflection and Amplification Attacks***: The mechanism is designed to be immune to reflection and amplification attacks. An attacker cannot use this protocol to turn a victim into a traffic amplifier. The critical design choice preventing this is that the receipt of an initial, unverified ICMP error message does NOT trigger the immediate transmission of a new packet. Instead, the host's response is limited to two low-cost internal actions: silently discarding the incoming message and setting a lightweight flag on an existing socket's control block. The challenge packet itself is not a new, separately generated packet; it is the `_next application packet_` for that flow, modified on-the-fly to include the Challenge-Confirm option. Therefore, an attacker sending a flood of forged ICMP messages cannot compel the target to generate any network traffic beyond what its applications would have sent anyway. The victim does not become a reflector.
- * ***Backward Compatibility***: The mechanism is fully backward-compatible. Hosts not implementing this specification will ignore the IP Option as per standard IP header processing rules [RFC1122]. Intermediate routers are unaffected. Only end hosts wishing to enhance their security need to implement the changes.

7. IANA Considerations

The Challenge-Confirm Option Type should be assigned in IANA's "IP Option Numbers" registry [RFC2780].

This draft requests the following IP Option Type assignments from the IP Option Numbers registry in the Internet Protocol (IP) Parameters registry group (<https://www.iana.org/assignments/ip-parameters>).

Copy	Class	Number	Value	Name	Reference
TBD	TBD	TBD	TBD	Challenge-Confirm	This draft

8. References

8.1. Normative References

- [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>>.
- [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>>.
- [RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, DOI 10.17487/RFC2780, March 2000, <<https://www.rfc-editor.org/rfc/rfc2780>>.
- [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>>.
- [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>>.
- [RFC792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <<https://www.rfc-editor.org/rfc/rfc792>>.
- [RFC9293] Eddy, W., Ed., "Transmission Control Protocol (TCP)", STD 7, RFC 9293, DOI 10.17487/RFC9293, August 2022, <<https://www.rfc-editor.org/rfc/rfc9293>>.

8.2. Informative References

- [CCS2020IPID] Feng, X., Fu, C., Li, Q., Sun, K., and K. Xu, "Off-path TCP exploits of the mixed IPID assignment", ACM Conference on Computer and Communications Security (CCS) , 2020.

[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", Network and Distributed System Security Symposium (NDSS) , 2022.

[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", IEEE Symposium on Security and Privacy (SP) , 2023.

[USENIXSECURITY2023ICMP]

Feng, X., Li, Q., Sun, K., Qian, Z., Fu, C., Zhao, G., Kuang, X., and K. Xu, "Off-Path Network Traffic Manipulation via Revitalized ICMP Redirect Attacks", USENIX Security Symposium (Security) , 2023.

Acknowledgments

The authors would like to thank the IETF community, particularly members of the INT-AREA working groups, for their valuable feedback and insights during the development of this proposal.

Authors' Addresses

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

Xuwei Feng
Tsinghua University
Beijing
China
Email: fengxw06@126.com

Yuxiang Yang
Tsinghua University
Beijing
China
Email: yangyx22@mails.tsinghua.edu.cn

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