Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 6 March 2026

R. Ehlers
PastWipe S.L.

2 September 2025

Reputation Security Protocol (RepSec)
draft-ehlers-repsec-00

Abstract

The Reputation Security Protocol (RepSec) defines a lightweight, extensible, and secure method for exchanging digital reputation and security-state information across the Internet. RepSec follows the design philosophy of SMTP (simplicity) and SNMP (extensibility). Entities can register, verify, remove, and audit reputation data in an interoperable and standardized way.

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 6 March 2026.

Table of Contents

1. Introduction 2
2. Requirements Language 3
3. Protocol Overview 3
4. Terminology 3
5. Protocol Elements 3
5.1. Transport and Encoding 3
5.2. Commands 3
5.2.1. REGISTER 3
5.2.2. VERIFY 4
5.2.3. REMOVE 4
5.2.4. AUDIT 4
5.2.5. PING 4
5.3. Responses 4
5.4. Example Session 4
6. Security Considerations 5
7. Extensibility 5
8. IANA Considerations 5
9. Licensing and IPR 5
10. References 5
10.1. Normative References 5
10.2. Informative References 6
Appendix A. Acknowledgments 6
Author’s Address 6

1. Introduction

Current approaches to reputation and security metadata are fragmented and proprietary. RepSec provides a standardized protocol for secure, interoperable exchange of this information. The protocol is designed to be simple to implement, secure by default, and extensible for future needs, drawing on SMTP’s command/response model and SNMP’s extensible object framework.

Goals:

- Simple, JSON-based message exchange
- Secure by default (TLS mandatory)
- Extensible command set and object schemas
- Open and free to implement

2. 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] when, and only when, they appear in all capitals, as shown here.

3. Protocol Overview

RepSec uses a client/server request宴途esponse model.

Transport: TCP with mandatory TLS (TLS 1.2 or later; TLS 1.3 SHOULD be supported).
Port: TBD by IANA (proposed: 4655).
Encoding: UTF-8 JSON for commands and responses.
Status Codes: Numeric, modeled after SMTP.

4. Terminology

Client: entity initiating RepSec commands.
Server: entity providing RepSec services.
Entity: subject of registration (domain, IP, userID, organization, etc.).
Message: JSON-encoded RepSec command or response.
Extension: optional module that defines new commands and/or schemas.

5. Protocol Elements

5.1. Transport and Encoding

All RepSec traffic MUST be transmitted over TLS. Servers SHOULD support TLS 1.3. Messages MUST be encoded as UTF-8 JSON objects. If compression is used, it MUST be negotiated at the TLS layer; TLS record compression is NOT RECOMMENDED.

5.2. Commands

The minimum interoperable command set is shown below. Commands are JSON objects with a "command" field and command-specific parameters.

5.2.1. REGISTER

Registers an entity with RepSec.

```
{ "command": "REGISTER", "entity": "example.com", "type": "domain", "auth": "token123" }
```

5.2.2. VERIFY

Verifies authenticity of registered data.

```
{ "command": "VERIFY", "entity": "example.com" }
```

5.2.3. REMOVE

Removes registered data.

```
{ "command": "REMOVE", "entity": "example.com" }
```

5.2.4. AUDIT

Retrieves entity history (audit log).

```
{ "command": "AUDIT", "entity": "example.com" }
```

5.2.5. PING

Checks server availability and round-trip.

```
{ "command": "PING" }
```

5.3. Responses

Responses MUST include a numeric status code and a human-readable message. Additional fields are allowed.

```
200 OK                Command succeeded
400 BAD REQUEST       Syntax error or missing parameters
401 UNAUTHORIZED      Authentication required or failed
403 FORBIDDEN         Command not permitted
404 NOT FOUND         No such entity or resource
409 CONFLICT          State conflict (e.g., already registered)
429 TOO MANY REQUESTS Rate limiting in effect
500 SERVER ERROR      Internal processing error
```

Example:

```
{ "status": 200, "message": "Entity registered successfully" }
```

5.4. Example Session

```
C: { "command": "REGISTER", "entity": "example.com", "type": "domain" }
S: { "status": 200, "message": "Entity registered successfully" }

C: { "command": "VERIFY", "entity": "example.com" }
S: { "status": 200, "verified": true, "last_updated": "2025-09-02T10:00:00Z" }
```

6. Security Considerations

All RepSec sessions MUST use TLS with server authentication; mutual authentication via client certificates or token-based schemes is RECOMMENDED. Implementations MUST provide replay protection (e.g., timestamps and nonces) and SHOULD employ rate limiting and abuse detection. Privacy by design: servers MUST NOT expose unnecessary metadata and SHOULD minimize data retention.

7. Extensibility

RepSec supports extensions similar to SMTP EHLO and SNMP MIBs. Extensions MAY define new commands and/or JSON schemas (RepSec Object Definitions). Servers SHOULD advertise supported extensions during capability discovery (future work).

8. IANA Considerations

IANA is requested to assign a new TCP port for RepSec (suggested: 4655) and to create a new "RepSec Parameters" registry containing:

- Registered RepSec Commands (Specification Required)
- Registered RepSec Extensions (Specification Required)
- Status Codes (Standards Action)

9. Licensing and IPR

This specification is made available under the IETF Trust Legal Provisions Relating to IETF Documents (TLP). Implementations may be licensed under permissive or copyleft licenses (e.g., Apache-2.0, GPL/LGPL), provided interoperability with the open standard is maintained. "RepSec" may be used as a trademark to indicate interoperable implementations.

10. References

10.1. Normative References

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

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

10.2. Informative References

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008, <https://www.rfc-editor.org/rfc/rfc5321>.

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 3411, December 2002, <https://www.rfc-editor.org/rfc/rfc3411>.

Appendix A. Acknowledgments

RepSec was inspired by the simplicity of SMTP and the extensibility of SNMP, with the goal of creating an open, non-proprietary Internet standard for reputation security.

Author’s Address

Ralph Ehlers
PastWipe S.L.
Marbella
Spain
Email: info@pastwipe.com