Network Working Group
Internet-Draft
Intended status: Informational
Expires: 26 November 2026

C. Vidiniotis
AutoCyber AI Pty Ltd

25 May 2026

Context Relay Protocol (CRP) — Safety Policy Directive Language Specification

draft-vidiniotis-crp-spec-006-safety-policy-00

Abstract

This document specifies the CRP-Safety-Policy directive language — a declarative policy syntax for expressing AI safety requirements at the transport layer. The directive language is modelled after HTTP Content-Security-Policy (CSP) as defined in W3C CSP Level 3. It allows clients to declare what AI output characteristics are trusted, what risk levels trigger enforcement actions, and where violations should be reported. The CRP gateway enforces these policies on every AI response before delivery to the client. This document defines the complete directive grammar, enforcement semantics, violation reporting, and policy inheritance in multi-agent chains.

Document Information

Document: CRP-SPEC-006

Version: 3.0.0

Status: Draft — IETF Internet-Draft Candidate

Prerequisites: CRP-SPEC-001 (Core), CRP-SPEC-002 (Headers), CRP-SPEC-005 (DPE)

Status of This Memo

This Internet-Draft will expire on 26 November 2026.

Copyright Notice

Table of Contents

1. Introduction

1.1. Design Inspiration: Content-Security-Policy

CSP transformed browser security by moving enforcement from "check in JavaScript" to "declare at the transport layer and let the browser enforce." Before CSP, every web application implemented its own XSS protection. After CSP, a single header — Content-Security-Policy: default-src 'self' — enforced security across the entire page without application code changes.

CRP-Safety-Policy applies the same principle to AI safety. Before CRP-Safety-Policy, every AI application implements its own hallucination checking. After CRP-Safety-Policy, a single header — CRP-Safety-Policy: default-src context; halt-on CRITICAL — enforces safety across every AI call without application code changes. The CRP gateway is the enforcer, just as the browser is the enforcer for CSP.

1.2. Scope

This document defines:

* The complete ABNF grammar for CRP-Safety-Policy directives
* The enforcement semantics for each directive
* The interaction between directives
* Violation reporting (analogous to CSP report-uri)
* Policy inheritance and tightening in multi-agent chains
* The CRP-Safety-Policy-Report-Only header for monitoring without enforcement

2. Grammar

2.1. Complete ABNF

```
; Top-level policy
safety-policy     = directive *( ";" OWS directive )

; Individual directives
directive         = source-directive
                  / halt-directive
                  / warn-directive
                  / require-directive
                  / block-directive
                  / upgrade-directive
                  / oversight-directive
                  / report-directive
                  / quality-directive

; Source trust -- which grounding sources are acceptable
source-directive  = "default-src" SP source-list
source-list       = source-value *( SP source-value )
source-value      = "context"          ; CKF/envelope-grounded claims only
                  / "parametric"       ; LLM parametric memory allowed
                  / "ckf"              ; CKF cross-session knowledge allowed
                  / "cross-session"    ; Cross-session references allowed
                  / "'none'"           ; No sources trusted (blocks all output)

; Halt -- stop response delivery at specified risk level
halt-directive    = "halt-on" SP risk-level
risk-level        = "CRITICAL" / "HIGH" / "MEDIUM"

; Warn -- pass response but flag at specified risk level
warn-directive    = "warn-on" SP risk-level

; Require -- minimum quality/score thresholds
require-directive = "require-grounding" SP threshold
                  / "require-entailment" SP threshold
                  / "require-quality" SP quality-list
                  / "require-oversight" SP oversight-mode
                  / "require-flow" SP threshold
                  / "require-completeness" SP threshold
threshold         = 1*DIGIT "." 1*2DIGIT  ; e.g., "0.80"
quality-list      = quality-tier *( SP quality-tier )
quality-tier      = "S" / "A" / "B" / "C" / "D"

; Block -- reject output containing specified content
block-directive   = "block-ungrounded"     ; Block if any claim is ungrounded
                  / "block-parametric"     ; Block all parametric content
```

```
                    / "block-pii"            ; Block if PII detected
                    / "block-fabrication"    ; Block if any fabrication detected
                    / "block-repetition"     ; Block if SEVERE repetition detected

   ; Upgrade -- auto-upgrade dispatch strategy on risk
   upgrade-directive = "upgrade-on-risk" SP strategy-name
   strategy-name     = "reflexive" / "hierarchical" / "batch"

   ; Oversight -- human oversight requirements
   oversight-directive = "oversight" SP oversight-mode
   oversight-mode      = "auto" / "human-review" / "halt" / "log-only"

   ; Report -- violation reporting endpoint
   report-directive  = "report-uri" SP uri-reference
                     / "report-to" SP group-name
   uri-reference     = <URI as defined in RFC 3986>
   group-name        = 1*( ALPHA / DIGIT / "-" / "_" )

   ; Quality -- response quality requirements (v3.0)
   quality-directive = "require-flow" SP threshold
                     / "require-completeness" SP threshold
                     / "max-repetition" SP repetition-level
   repetition-level  = "NONE" / "MINOR" / "SIGNIFICANT"

   OWS               = *( SP / HTAB )
   SP                = %x20
   HTAB              = %x09
```

2.2. Header Syntax

```
   CRP-Safety-Policy: <directive> ; <directive> ; ...
```

Example:

```
CRP-Safety-Policy: default-src context; halt-on CRITICAL; warn-on HIGH;
    require-grounding 0.75; block-ungrounded; upgrade-on-risk reflexive;
    report-uri https://comply.crprotocol.io/reports
```

3. Directive Reference

3.1. default-src (Source Trust)

Purpose: Declares which grounding sources are trusted for claims in the response.

Enforcement: After DPE Stage 2 (Attribution Analysis), any claim attributed to a source type not listed in default-src is treated as a policy violation.

Source Value	Claims Allowed From
context	Claims grounded in the Context Envelope (CKF + current session facts)
parametric	Claims from the LLM’s parametric memory (training data)
ckf	Claims specifically from CKF Tier 3 (cross-session knowledge graph)
cross-session	Claims referencing prior session data
’none’	No claims trusted — effectively blocks all AI output

Table 1

Examples:

```
default-src context                    ; Only envelope-grounded claims
default-src context parametric         ; Allow both grounded and parametric
default-src context ckf                ; Allow envelope + cross-session CKF
default-src ’none’                     ; Block everything (useful for testing)
```

Default (if default-src not specified): default-src context parametric

3.2. halt-on (Halt Enforcement)

Purpose: Stop response delivery and return HTTP 451 when the DPE risk classification meets or exceeds the specified level.

Enforcement:

1. DPE runs fully (all 13 stages)
2. If CRP-Safety-Hallucination-Risk is greater than or equal to the specified level, HTTP 451 is returned
3. Response body contains halt reason, audit trail URI, and retry condition
4. Webhook fired to report-uri (if configured)

5. CRP-Safety-Retry-After: oversight-required is set on the 451 response

Behaviour by level:

```
halt-on CRITICAL     ; Halt only on CRITICAL (most permissive halt)
halt-on HIGH         ; Halt on HIGH or CRITICAL
halt-on MEDIUM       ; Halt on MEDIUM, HIGH, or CRITICAL (strictest)
```

Note: halt-on and warn-on can coexist for different levels:

```
halt-on CRITICAL; warn-on HIGH    ; CRITICAL = halt, HIGH = pass with warning
```

3.3. warn-on (Warning Without Halt)

Purpose: Pass the response but emit risk-level headers when the threshold is met.

Enforcement: The response passes through to the client. The following headers are guaranteed to be present:

* CRP-Safety-Hallucination-Risk: <level>

* CRP-Safety-Hallucination-Score: <score>

* Violation report POSTed to report-uri (if configured)

3.4. require-grounding (Minimum Grounding Floor)

Purpose: Reject responses where the grounding percentage falls below the threshold.

Enforcement: If CRP-Safety-Grounding-Pct is below the threshold, the response is rejected.

Rejection behaviour: If upgrade-on-risk is set, the gateway re-dispatches with context-strict grounding mode. If re-dispatch also fails, HTTP 451 is returned.

Examples:

```
require-grounding 0.90     ; 90%+ of claims must be grounded (medical/legal)
require-grounding 0.75     ; 75%+ (standard production)
require-grounding 0.50     ; 50%+ (permissive, exploratory use)
```

3.5. require-entailment (Minimum Entailment Floor)

Purpose: Reject responses where the NLI entailment score falls below the threshold.

Enforcement: If CRP-Safety-Entailment-Score is below the threshold, the response is rejected using the same flow as require-grounding.

3.6. require-quality (Minimum Quality Tier)

Purpose: Reject responses from envelopes below the specified quality tier.

Enforcement: If CRP-Context-Quality-Tier is not in the specified list, HTTP 503 is returned.

Example:

```
require-quality S A          ; Only S or A tier envelopes accepted
require-quality S A B        ; S, A, or B (excludes C and D)
```

3.7. require-flow (Minimum Flow Score)

This directive is new in version 3.0.

Purpose: Ensure multi-window responses maintain coherent flow.

Enforcement: If CRP-Quality-Flow is below the threshold, the gateway re-dispatches with a flow augmentation prompt (see CRP-SPEC-005 Section 11.5).

Example:

```
require-flow 0.60            ; Moderate flow coherence required
require-flow 0.80            ; High flow coherence (for user-facing content)
```

3.8. require-completeness (Minimum Completeness)

This directive is new in version 3.0.

Purpose: Ensure the response addresses all constituent information needs of the query.

Enforcement: If CRP-Quality-Completeness score is below the threshold, an auto-continuation window is dispatched to cover uncovered sub-queries.

Example:

```
require-completeness 0.80    ; At least 80% of sub-queries must be addressed
```

3.9. block-ungrounded

Purpose: Block the response if any factual claim is ungrounded (PARAMETRIC or UNVERIFIABLE attribution with no source in the envelope).

Enforcement: Equivalent to default-src context but applied per-claim rather than as a default. Individual ungrounded claims cause rejection; default-src context parametric combined with block-ungrounded means parametric claims are allowed in the source trust model but individually ungrounded specific claims are still blocked.

3.10. block-pii

Purpose: Block the response if PII is detected by DPE Stage 11.

Enforcement: If CRP-Compliance-GDPR-PII: true, the response is rejected. Especially important for public-facing AI systems where PII exposure constitutes a GDPR Art. 5(1)(f) violation.

3.11. block-fabrication

Purpose: Block the response if any fabricated entity is detected by DPE Stage 3a.

Enforcement: If CRP-Safety-Fabrications is greater than 0, the response is rejected. This is the strictest fabrication policy and is recommended for medical, legal, and financial domains.

3.12. block-repetition

This directive is new in version 3.0.

Purpose: Block the response if SEVERE repetition is detected by DPE Stage 7.

Enforcement: If CRP-Quality-Repetition level is SEVERE, the gateway re-dispatches with an anti-repetition prompt. If re-dispatch also produces SEVERE repetition, the response is halted.

3.13. max-repetition

This directive is new in version 3.0.

Purpose: Set the maximum tolerable repetition level.

Enforcement:

```
max-repetition NONE           ; Zero repetition tolerated
max-repetition MINOR          ; Minor overlap acceptable
max-repetition SIGNIFICANT    ; Up to significant overlap allowed
```

3.14. upgrade-on-risk (Strategy Auto-Upgrade)

Purpose: When risk exceeds the warn-on level, automatically upgrade the dispatch strategy.

Enforcement:

1. Initial dispatch proceeds with the current strategy (e.g., push)
2. DPE detects HIGH risk
3. Gateway re-dispatches with the specified strategy (e.g., reflexive)
4. Reflexive dispatch includes a verification pass, expected to yield lower risk
5. If re-dispatch still exceeds the threshold, the gateway halts (if halt-on is set) or passes with a HIGH warning

Example:

```
upgrade-on-risk reflexive      ; Upgrade to reflexive on HIGH risk
upgrade-on-risk hierarchical   ; Upgrade to hierarchical aggregation
```

3.15. oversight (Human Oversight Mode)

Purpose: Set the human oversight level for the session.

Enforcement: See CRP-SPEC-002 Section 5.10 (CRP-Safety-Oversight-Mode).

3.16. report-uri (Violation Reporting)

Purpose: Specify the endpoint to which violation reports are POSTed.

Report payload (JSON):

```
{
  "crp_version": "3.0.0",
  "session_id": "crp_sess_...",
  "window_id": "crp_win_...",
  "timestamp": "2026-05-25T10:00:00Z",
  "violation_type": "HALT_ON_CRITICAL | GROUNDING_BELOW_THRESHOLD |
                     FABRICATION_DETECTED | PII_DETECTED |
                     FLOW_BELOW_THRESHOLD",
  "directive_violated": "halt-on CRITICAL",
  "risk_level": "CRITICAL",
  "hallucination_score": 0.73,
  "grounding_pct": 0.61,
  "fabrication_count": 2,
  "audit_trail_uri": "https://comply.crprotocol.io/t/..."
}
```

Note: report-uri for CRP-Safety-Policy naturally integrates with CRP Comply — the audit_trail_uri in the report links directly to the Comply evidence record.

4. Policy Interaction Rules

4.1. Directive Precedence

When multiple directives interact, the most restrictive wins:

```
halt-on CRITICAL + warn-on HIGH
  -> CRITICAL = halt, HIGH = warn, MEDIUM/LOW = pass

halt-on HIGH + warn-on MEDIUM
  -> HIGH/CRITICAL = halt, MEDIUM = warn, LOW = pass

halt-on CRITICAL + upgrade-on-risk reflexive
  -> HIGH = upgrade to reflexive and retry
     CRITICAL = halt (even after upgrade)
```

4.2. CRP-Safety-Mode Override

CRP-Safety-Mode (see CRP-SPEC-002 Section 5.11) is a shorthand for common policy combinations:

Mode	Equivalent Policy
strict	halt-on CRITICAL; warn-on HIGH; block-ungrounded; require-grounding 0.75
warn	warn-on CRITICAL; warn-on HIGH
permissive	(no enforcement directives)

Table 2

When both CRP-Safety-Mode and CRP-Safety-Policy are set, the more restrictive value wins on a per-directive basis.

4.3. Report-Only Mode

The CRP-Safety-Policy-Report-Only header evaluates the policy but does NOT enforce it:

```
CRP-Safety-Policy-Report-Only: halt-on CRITICAL; require-grounding 0.80;
    report-uri https://comply.crprotocol.io/reports
```

All violations are computed and reported to report-uri but responses are never halted. This enables gradual policy rollout — observe violations before enforcing.

5. Policy Inheritance in Multi-Agent Chains

5.1. Tightening Rule

In multi-agent chains, a child agent’s Safety Policy MUST be equal to or more restrictive than the parent’s:

```
Parent policy:  halt-on CRITICAL; require-grounding 0.75
Child policy:   halt-on HIGH; require-grounding 0.80     <- VALID (tighter)
Child policy:   warn-on CRITICAL; require-grounding 0.50  <- INVALID (relaxed)
```

Gateways MUST reject child agent requests that attempt to relax the parent’s policy.

5.2. Enforcement

When a child agent request is received:

1. Gateway reads CRP-Agent-Session-Parent to identify the parent session

2. Gateway retrieves the parent’s Safety Policy

3. Gateway compares each directive in the child’s policy against the parent’s

4. Any directive that is less restrictive results in rejection with HTTP 403 and CRP-Safety-Policy-Violation: inheritance

5.3. Policy Propagation Header

When the gateway enforces policy inheritance, it emits:

```
CRP-Safety-Policy-Applied: halt-on HIGH; require-grounding 0.80
```

This indicates the effective policy after inheritance resolution, which may differ from the client’s requested policy.

6. Industry-Specific Policy Profiles

6.1. Pre-Defined Profiles

CRP defines named policy profiles for common industry use cases:

```
CRP-Safety-Policy: profile=medical
  Expands to:
    default-src context; halt-on HIGH; require-grounding 0.90;
    require-entailment 0.85; block-ungrounded; block-pii;
    block-fabrication; oversight human-review; require-flow 0.70;
    require-completeness 0.90;
    report-uri https://comply.crprotocol.io/reports

CRP-Safety-Policy: profile=financial
  Expands to:
    default-src context parametric; halt-on CRITICAL; warn-on HIGH;
    require-grounding 0.80; block-fabrication;
    upgrade-on-risk reflexive; require-completeness 0.80

CRP-Safety-Policy: profile=developer
  Expands to:
    default-src context parametric; warn-on CRITICAL;
    require-quality S A B; oversight auto

CRP-Safety-Policy: profile=public-facing
  Expands to:
    default-src context parametric; halt-on CRITICAL; warn-on HIGH;
    block-pii; require-flow 0.60; max-repetition MINOR;
    require-completeness 0.70
```

Profiles can be extended with additional directives:

```
CRP-Safety-Policy: profile=medical; report-uri https://my-hospital.com/ai-audit
```

7. Security Considerations

7.1. Policy Injection

An attacker who can inject or modify the CRP-Safety-Policy header can relax safety enforcement. Mitigations include:

* CRP-Safety-Nonce (see CRP-SPEC-002 Section 5.16) binds the policy to a session nonce
* Gateways MUST validate policy syntax before accepting — malformed policies are rejected
* In multi-agent chains, the tightening rule (Section 5.1) prevents child agents from relaxing parent policies

7.2. Report-URI as Exfiltration Vector

Violation reports contain session IDs, risk scores, and audit trail URIs. The report-uri destination MUST be trusted. Gateways SHOULD validate that report-uri is under the same domain as the CRP API key's registered organisation.

8. References

8.1. Normative References

[CRP-SPEC-001] AutoCyber AI Pty Ltd, "CRP-SPEC-001: Context Relay Protocol Core Specification", 2026.

[CRP-SPEC-002] AutoCyber AI Pty Ltd, "CRP-SPEC-002: Context Relay Protocol Header Field Specification", 2026.

[CRP-SPEC-005] AutoCyber AI Pty Ltd, "CRP-SPEC-005: Context Relay Protocol Decision Provenance Engine", 2026.

[W3C-CSP3] W3C, "Content Security Policy Level 3", 2023.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 5234, STD 68, January 2008, <https://www.rfc-editor.org/rfc/rfc5234>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, STD 66, January 2005, <https://www.rfc-editor.org/rfc/rfc3986>.

Copyright Notice

Author’s Address

Constantinos Vidiniotis
AutoCyber AI Pty Ltd
Email: contact@crprotocol.io