Bilko Rate Limiting — Trusted Client IP Strategy (ADR-022)

Bilko Rate Limiting — Trusted Client IP Strategy (ADR-022)

Summary

This document describes the fix for the `X-Forwarded-For` spoofing vulnerability in Bilko's rate limiter (MC #99917, PR #63). The rate limiter previously read the leftmost (attacker-controlled) value from the `X-Forwarded-For` header using `firstOrNull()`, allowing trivial bypass of all three rate-limit buckets. The fix introduces a `TrustedIpExtractor` helper that reads a configurable number of trusted proxy hops from the right of the XFF chain via the `TRUSTED_PROXY_HOP_COUNT` environment variable (default: 2).

Bilko's production topology (verified 2026-05-08) consists of three hops: **Internet → GCLB EXTERNAL_MANAGED → Serverless NEG → Cloud Run GFE → Ktor**. GCLB appends the real client IP, and Cloud Run GFE appends the GCLB POP address, resulting in an XFF chain of `[attacker-supplied-values, real-client-ip, gclb-pop-ip]`. With `hopCount=2`, the extractor correctly reads `xff[size - 2]` to retrieve the real client IP.

Network Topology

Bilko's production topology (as of 2026-05-08) consists of:

```
Internet client
     |
     v
GCLB EXTERNAL_MANAGED (google_compute_global_forwarding_rule "bilko-stage-https-fwd")
     | TLS termination, Cloud Armor policy (if enabled)
     v
Serverless NEG (google_compute_region_network_endpoint_group "bilko-stage-api-neg")
     |
     v
```

```
Cloud Run GFE (internal Google Frontend — *.run.app / *.europe-north1.run.app)
     |
     v
Ktor Application (bilko-api-stage container)
```

X-Forwarded-For structure on the GCLB path:

```
X-Forwarded-For: <attacker-supplied-values>, <real-client-ip>, <gclb-pop-ip>
                 ^                           ^                  ^
                 Index 0 (NEVER trust)       Index size-2       Index size-1
                                             (real client)      (GCLB POP, appended by GFE)
```

Direct *.run.app bypass path: With `INGRESS_TRAFFIC_ALL` set (confirmed live 2026-05-08), direct `*.run.app` requests skip GCLB entirely. On that path, only one GCP hop is appended (Cloud Run GFE), so `hopCount=2` would under-extract. This is a residual risk tracked in MC #99924 (FlowForge INGRESS lockdown).

GCP Documentation Reference: https://cloud.google.com/load-balancing/docs/https#x-forwarded-for_header

TrustedIpExtractor Pattern

The `TrustedIpExtractor` utility (`apps/api/src/main/kotlin/no/alai/bilko/util/TrustedIpExtractor.kt`) implements the following algorithm:

1. Read `TRUSTED_PROXY_HOP_COUNT` from the environment variable (default: 2).
2. Split the `X-Forwarded-For` header on commas, trim each entry, and filter out empty values.
3. Return the entry at index `size - hopCount`.
4. If the XFF header is absent or shorter than `hopCount`, fall back to `call.request.local.remoteAddress`.

Code Excerpt

```
object TrustedIpExtractor {
    val hopCount: Int = run {
        val raw = System.getenv("TRUSTED_PROXY_HOP_COUNT")
        raw?.toIntOrNull()?.takeIf { it >= 1 } ?: 2
    }

    fun extractTrustedClientIp(call: ApplicationCall): String {
        val xffHeader = call.request.header("X-Forwarded-For")
        val remoteAddress = call.request.local.remoteAddress
        return extractFromParts(xffHeader, remoteAddress, hopCount)
```

```
    }

    fun extractFromParts(xffHeader: String?, remoteAddress: String, hopCount: Int = this.hopCour
        if (xffHeader.isNullOrBlank()) return remoteAddress

        val parts = xffHeader.split(",").map { it.trim() }.filter { it.isNotEmpty() }

        if (parts.size < hopCount) {
            return remoteAddress  // XFF chain shorter than expected — fall back
        }

        return parts[parts.size - hopCount]
    }
}
```

Fallback Behavior

WARNING: On Cloud Run, `call.request.local.remoteAddress` is the Cloud Run GFE internal network address — **NOT** the real client IP. Falling back here degrades the rate-limiter to per-GFE-region keying (all requests without XFF share one bucket per region). This is acceptable as a last resort; it is not a security bypass.

Environment Variable Contract

- **Name:** `TRUSTED_PROXY_HOP_COUNT`
- **Default:** 2 (correct for Bilko GCLB + Cloud Run GFE topology)
- **Valid range:** ≥ 1 (negative or zero values fall back to default)
- **Override location:** `apps/api/src/main/resources/.env.example`

Rate Limiting Bucket Strategy (Post #99917)

Bilko uses three rate-limit buckets, each with distinct keying strategies:

1. `auth` Bucket (5 requests/minute)

Applied to: Pre-authentication routes (`/register`, `/login`, `/2fa/challenge`, `/refresh`).

Key strategy: IP address via `TrustedIpExtractor` (closes XFF spoofing).

Trade-off: A shared corporate NAT means all users behind it share the same rate-limit bucket. This is acceptable for login attempts (5 attempts/min × team size is typically sufficient). JWT principal is not yet issued at this stage, so IP keying is the only viable option.

Alternative considered: Email/username keying from request body (Parisa dissent PARISA-TABRIZ-D1) — deferred as follow-up improvement.

2. `api` Bucket (100 requests/minute)

Applied to: All authenticated routes inside `authenticate("bilko-jwt")`.

Key strategy: JWT principal `organizationId` from `BilkoPrincipal`. Falls back to IP via `TrustedIpExtractor` if principal is unavailable (should not normally happen inside the authenticated block).

Benefit: Removes office-NAT lockout for paying customers. Each organization has its own rate-limit bucket.

3. `public` Bucket — REMOVED

The `public` bucket was registered in `RateLimit.kt` but never mounted in `Routing.kt` (confirmed: no route file uses the public bucket name). Dead code removed to eliminate confusion. If a public route is added in future, add the bucket registration back alongside the matching `rateLimit(...) { ... }` route wrapper.

ADR-022 — IP Trust Strategy

Status

Accepted (2026-05-08)

Context

The rate limiter in `RateLimit.kt` previously used `X-Forwarded-For.split(",").firstOrNull()` to extract the client IP, reading the leftmost (attacker-controlled) value. Any caller who could set HTTP headers could bypass all three rate-limit buckets by supplying a fresh fake IP per request.

Bilko's production topology consists of three hops: GCLB EXTERNAL_MANAGED + Serverless NEG + Cloud Run GFE. GCLB preserves the incoming XFF and appends the real client IP; Cloud Run GFE then appends the GCLB POP address. The real client IP is therefore at index `size - 2` (two trusted hops from the right).

Decision

We will use an explicit **trusted proxy hop count** via the `TRUSTED_PROXY_HOP_COUNT` environment variable to determine the correct extraction offset. The default value is **2** for Bilko's current GCLB + Cloud Run topology. This makes the offset empirically verifiable and reconfigurable without requiring a code deploy.

Consequences

- **Positive:** Topology changes (e.g., adding a Cloudflare proxy layer upstream) require only an environment variable update, not a code change. The extraction logic is testable and deterministic.
- **Negative:** Direct `*.run.app` bypass remains open (tracked in MC #99924) until `INGRESS_TRAFFIC_ALL` is locked to `INGRESS_TRAFFIC_INTERNAL_LOAD_BALANCER_ONLY`. On that path, `hopCount=2` under-extracts by one position (Cloud Run GFE appends only one hop, not two).
- **Deferred:** Distributed rate-limit state (Jedis) is tracked as a follow-up MC. In-memory per-instance counters mean `auth` bucket (limit=5) multiplies by instance count. With `max_instance_count=2`, effective limit is 10 brute-force attempts per minute. Jedis is declared in `build.gradle.kts` but not wired.
- **Parallel vulnerability:** `CallLogging.kt` lines 31-32 log the raw full XFF string without trusted-IP extraction. Fabricated IPs persist in the audit log. Tracked in MC #99925 (same fix pattern: use `TrustedIpExtractor`).

Alternatives Considered

1. **Naive `lastOrNull()`** (Kelsey dissent): Wrong for Bilko's 3-hop GCLB topology. `lastOrNull()` consumes the GCLB POP IP, not the real client IP. Correct index is `size - 2`.
2. **Cloud Armor IP rate-limiting at GFE layer** (Petter dissent): Deferred. Compute Engine API is disabled on project `tribal-sign-487920-k0`. Cloud Armor is declared in Terraform (`enable_cloud_armor = true`) but not deployed. Tracked as a follow-up MC (FlowForge infra workstream).
3. **Jedis distributed counter** (Kleppmann dissent): Deferred. In-memory per-instance counters are insufficient for the `auth` bucket at scale, but implementing Jedis-backed distributed rate-limiting is scope-creep for this MC. Tracked as a follow-up MC after Redis endpoint is provisioned.
4. **Identity-based keying (email/username) for auth bucket** (Parisa dissent): Rejected for this MC. JWT principal is not yet issued at pre-auth stage, so email/username extraction from request body is the only alternative to IP keying. This requires route-level key function (not plugin-level) and is deferred as a follow-up improvement.

Operational Notes

Override TRUSTED_PROXY_HOP_COUNT

If the network topology changes (e.g., adding a Cloudflare proxy layer), update `TRUSTED_PROXY_HOP_COUNT` in `apps/api/src/main/resources/.env.example` and redeploy. For example, adding Cloudflare upstream would require `TRUSTED_PROXY_HOP_COUNT=3`.

Live Spoof Probe (Post-Merge)

After PR #63 is merged and deployed to stage, run the following probe to verify the fix:

```
for i in {1..6}; do
  curl -H "X-Forwarded-For: 1.2.3.4" -i https://api.bilko.io/api/v1/auth/login
done
```

Expected result: First 4 requests succeed (200 OK). Requests 5 and 6 return `429 Too Many Requests` keyed on the **real client IP**, not the spoofed `1.2.3.4` value.

Known Limitation: *.run.app Direct Bypass

As of 2026-05-08, both `bilko-api` and `bilko-api-stage` use `ingress = "INGRESS_TRAFFIC_ALL"` (verified in `compute/main.tf` line 28). This means direct `*.run.app` URL access bypasses GCLB and Cloud Armor. XFF on that path is 100% attacker-controlled regardless of any IP extraction logic in `RateLimit.kt`.

Mitigation: MC #99924 (FlowForge, H priority) will lock ingress to `INGRESS_TRAFFIC_INTERNAL_LOAD_BALANCER_ONLY`. This fix (#99917) mitigates the GCLB path only.

Impact analysis: `CORS_ORIGINS` env variable (section 5 of topology probe) confirms `*.run.app` URLs are present in CORS allow-list. Audit required to determine if these are stale entries or if web frontend still calls the `*.run.app` URL directly.

Cross-References

- **PR #63:** feat/99917-trusted-ip-extractor
- **MC #99917:** This fix (rate-limit IP trust strategy)
- **MC #99924:** INGRESS lockdown (FlowForge, H priority, parallel)
- **MC #99925:** CallLogging.kt parallel vulnerability (CodeCraft, M priority, depends on #99917)
- **Forged prompt:** `~/system/prompts/forged/99917.md`
- **Topology probe:** `docs/security/rate-limit-topology-probe-2026-05-08.md`

Document prepared by: *Skillforge (MC #99917 D5)*
Validated by: *Proveo (angie-jones, MC #99917 D4)*
Date: *2026-05-08*

Status: *ADR-022 Accepted*

Revision #2
Created 2026-05-08 10:31:42 UTC by John
Updated 2026-06-14 20:02:36 UTC by John