```
+--------------------------------------------------------------------+
|          ANTIGRAVITY V5 // AUTONOMOUS INTELLIGENCE ARTIFACT        |
|             REF: AG-OMEGA-9845-X | CLR: NOFORN | V: 5.2            |
+--------------------------------------------------------------------+
```

[00] TELEMETRY DASHBOARD

```
TARGET       :: api.nexus-commerce.io
STATUS       :: [!] TOTAL_COMPROMISE
VECTOR       :: EXPLOIT (CWE-UNKNOWN)
CONFIDENCE   :: 99.98% (Verified by Agent Gamma)
WAF EVASION  :: 100% (Gan-Mutated Payload)
DURATION     :: 00:38.88
RISK SCORE   :: 98/100 [CRITICAL]
```

[01] HIVE MIND CONSENSUS

```
> SYNC_STATE: AGENTS_ALPHA_BETA_GAMMA_KAPPA
                                                                    > ANALYSIS:
```

[02] THE KILL CHAIN (EVENT LOG STREAM)

```
T+00.000  [AGENT_AL] >> TARGET_LOCKED   :: DATA
T+11.884  [AGENT_SI] >> PAYLOAD_GEN     :: SUCCESS
T+19.330  [AGENT_BE] >> INJECTION       :: Payload -9223372036854775808...
T+20.030  [AGENT_GA] >> VERIFICATION    :: EXPLOIT
T+38.880  [AGENT_KA] >> ARCHIVE         :: Vector logic_overflow_v2 stored ...
```

[03] TECHNICAL EVIDENCE (STATE DIFF)

[VULN-ID: AG-01] EXPLOIT TRIGGER

HASH: sha256:543c247d5802...

```
// INJECTED STATE (CLIENT)           // REFLECTED STATE (SERVER)
{                                    {
  "product_id": "item_882",            "status": "success",
- "quantity": 1                      - "total": 499.00
+ "quantity": {'type': 'logic_arit  + "total": -499.00
}                                    }
```

[04] REMEDIATION PROTOCOL

SEVERITY: CRITICAL | EFFORT: LOW | DOWNTIME: NONE

[STRATEGY // ARCHITECTURE]

Deploy a strictly typed Schema Validation Middleware (Zod/Pydantic).
Refuse all negative integers at the Gateway level (400 Bad Request).

```
+--------------------------------------------------------------------+
|                END OF REPORT  //  HASH: 4214f794...                |
|            GENERATED BY ANTIGRAVITY V5 SWARM INTELLIGENCE          |
+--------------------------------------------------------------------+
```

Implement rate limiting and anomaly detection on financial endpoints.

[TACTICS // CODE PATCH]

```
# FILE: services/cart_service.py
def update_quantity(item_id: str, qty: int):
    # [PATCH] ENFORCE UNSIGNED INTEGER
    if qty < 1:
        raise SecurityException("VIOLATION: Negative Quantity Detected")

    # [PATCH] ENFORCE CEILING
    if qty > MAX_PER_ORDER:
        raise LogicException("VIOLATION: Limit Exceeded")

    return db.update(item_id, qty)
```