AI Workflows in the SOC

Speeding up Defense with Automation & LLM Orchestration

John Van Lowe - ATX Javascript - AI Workflows - July 2026

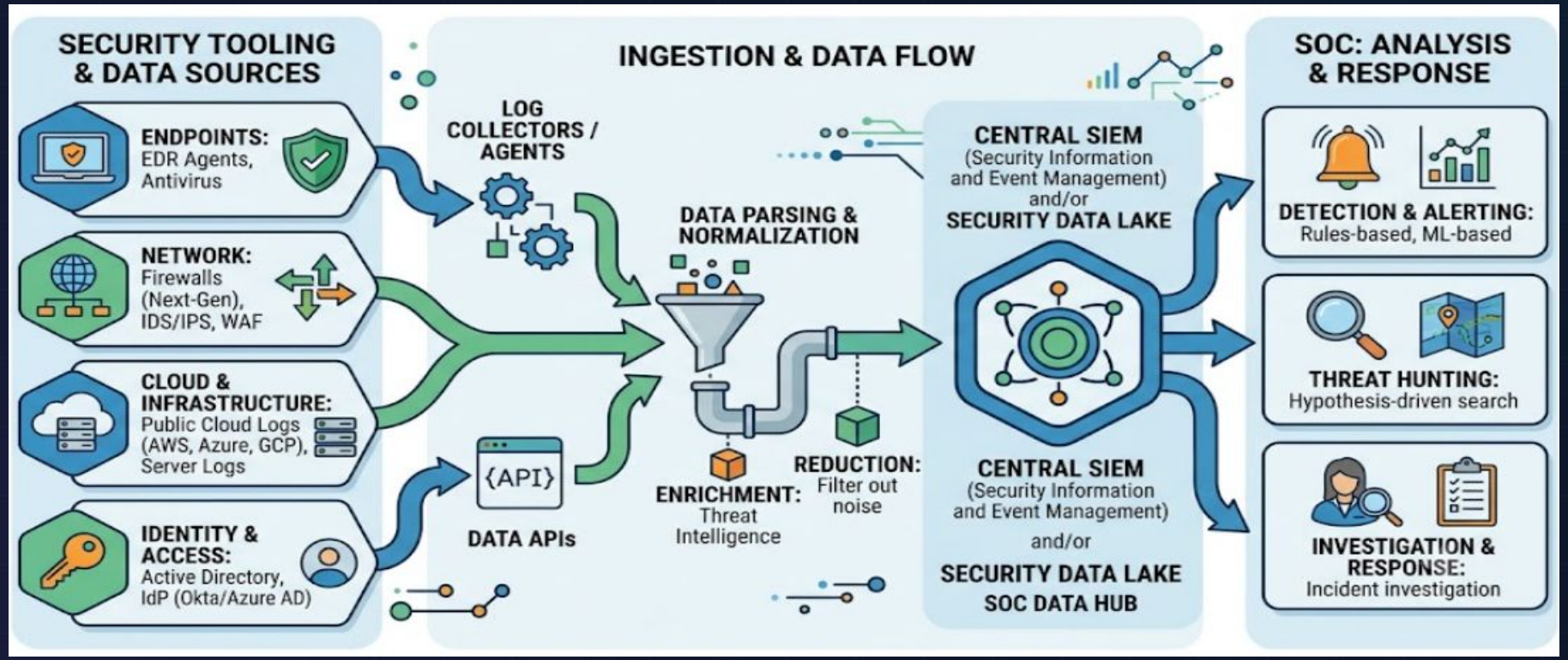
The Problem: Modern SOC Challenges

Alert Fatigue & Volume

SOC analysts process thousands of daily telemetry alerts across SIEM, EDR, and Cloud platforms. Up to 70% are false positives, leading to critical burnout and missed threats.

Context Switching Friction

Investigating a single alert requires toggling between 8+ disparate tools for IP lookups, log correlation, and threat intelligence. MTTD and MTTR spike as context is fragmented.



The Intent: Core Pillars of AI SOC Workflows

1. Ingestion & Triage

Automated parsing and prioritization of high-volume SIEM alerts using lightweight microservices and natural language classification.

2. Agent Enrichment

Autonomous lookup of VirusTotal, WHOIS, and endpoint state to produce clear, structured incident summaries for Tier 1 analysts.

3. Governed Action

Enforce human-IN/ON-the-loop approvals for sensitive remediation steps like host isolation or user account suspension.

85%

MTTR REDUCTION

Goal!? Accelerating SOC Outcomes

Automating Tier 1 triage and context gathering slashes Mean Time to Respond from hours to minutes. Analysts are liberated to focus on proactive threat hunting and continuous defense engineering.

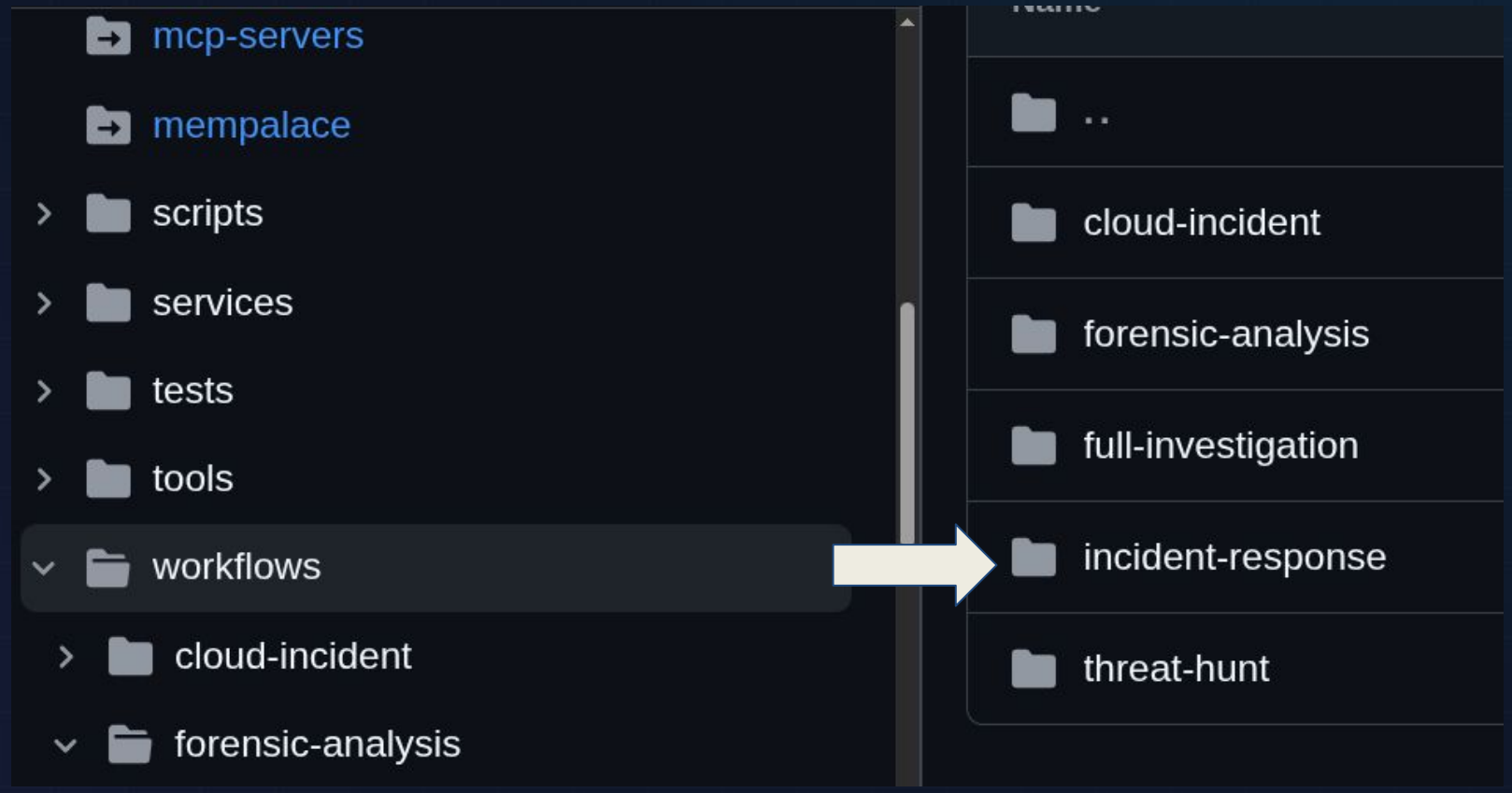
How It Is Done: Enter Vigil SOC Workflows

Beyond Static Playbooks

Traditional playbooks rely on hardcoded conditional rules that fail when attack patterns evolve.

AI workflows combine deterministic code execution with dynamic LLM agent reasoning. It allows us to evaluate incoming telemetry in context before taking action.

Distributed in Markdown, revision controlled within Git.



https://github.com/Vigil-SOC/vigil/blob/main/workflows/incident-response/WORKFLOW.md

AI Workflow: IR Workflow In Depth

Incident Response Workflow

Multi-agent incident response workflow following the NIST Incident Response framework. Sequences four specialized agents to rapidly triage, deeply investigate, contain threats, and produce audit-ready documentation.

USE CASES:

Security alert has fired requiring immediate review
Finding is flagged as critical or secure
An active threat (malware, C2, data exfiltration) is occuring
You need end to end triaging for a report

name	incident-response
description	Respond to active security incidents with rapid triage, deep investigation, containment, and documentation. Follows NIS
agents	triage, investigator, responder, reporter
tools-used	get_finding, list_findings, nearest_neighbors, search_detections, create_approval_action, update_case
use-case	Active incident response -- an alert fires and the SOC needs to triage, investigate, contain, and document.
trigger-examples	Run incident response on finding f-20260215-abc123, We have an active incident -- ransomware detected on HOST

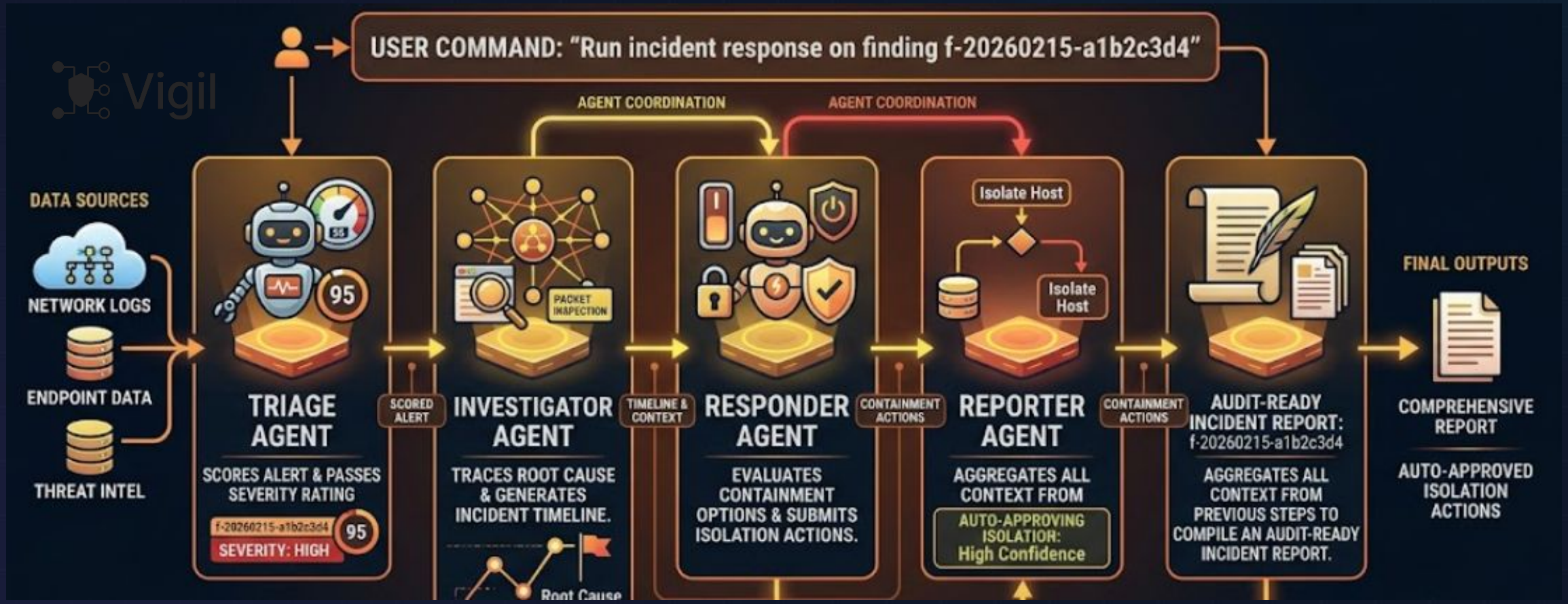
Example Workflow: Vigil SOC Incident Response

1. Ingest
Stream alerts via Webhooks to a Node.js Express/Fastify listener.

2. Enrich
LLM Agent queries threat intelligence APIs in parallel.

3. Evaluate
Synthesize risk score and draft executive incident summary.

4. Respond
Trigger Slack approval prompt before execution.



Path to Full Autonomy: Incident Response

Phase 1: Triage & Classify (Triage Agent)

Purpose: Rapid assessment of the alert -- severity scoring, false-positive check, categorization.

Tools: `get_finding`, `list_findings`

Steps:

1. Fetch the finding via `get_finding` to retrieve full details
2. Assess severity based on anomaly score, data source, and MITRE techniques
3. Categorize the alert: malware, intrusion, policy violation, reconnaissance, exfiltration, or false positive
4. Assign priority: Critical (immediate action), High (within 1hr), Medium (queue), Low (monitor), False Positive (dismiss)
5. Make escalation decision with reasoning

Output: Severity score, alert category, priority level, escalation decision

Short-circuit: If classified as False Positive, skip to Phase 4 for a brief dismissal report.

Lifecycle Stage	Automation Level	Key Automated Tasks
Ingestion & Triage	🔴 High	Deduplication, alert grouping, initial severity scoring
Enrichment	🔴 High	Threat intel queries, user/asset context lookups, Slack/Teams verification
Investigation	🟡 Partial	Sandbox detonation, forensic artifact collection, automated timeline generation
Containment	🟢 Medium/High	Host isolation, session revocation, IP/domain blocking (often needs approval)
Eradication	🟡 Partial	Email purge, file deletion, remote process termination
Recovery	🔵 Low	Backup restoration triggers, continuous system monitoring
Reporting	🟡 Partial	Report template generation, metric logging, ticket status synchronization

AI Workflow: IR Agent Phases

Phase 1: Triage & Classify (Triage Agent)

Purpose: Rapid assessment of the alert -- severity scoring, false-positive check, categorization.
Tools: `get_finding`, `list_findings`
Output: Severity score, alert category, priority level, escalation decision
Short-circuit: If classified as False Positive, skip to Phase 4 for a brief dismissal report.

Phase 2: Deep Investigation (Investigator Agent)

Purpose: Root cause analysis, evidence collection, timeline reconstruction, cross-source correlation.
Tools: `get_finding`, `list_findings`, `nearest_neighbors`, `search_detections`
Output: Root cause, attack vector, affected entities (IPs/hosts/users), evidence chain, related findings, timeline

Phase 3: Contain & Respond (Responder Agent)

Purpose: NIST IR containment -- isolate hosts, block IPs, revoke credentials, plan remediation.
Tools: `create_approval_action`, `get_finding`, `update_case`
Output: Containment actions with confidence scores, approval requests, remediation checklist, blast radius assessment

Phase 4: Document & Report (Reporter Agent)

Purpose: Generate audience-tailored incident report with executive summary, technical details, and lessons learned.
Tools: `get_case`, `list_findings`, `create_attack_layer`
Output: Complete incident report, MITRE ATT&CK layer, event timeline, recommendations

Human-in-the-Loop & HOTL Safety

Deterministic Guardrails & Confidence Scores

AI can make confidence gated high-impact non-destructive actions autonomously. As models and tooling mature it is becoming more common to allow autonomy with potentially destructive actions. Implement deterministic validation filters around LLM outputs to guarantee zero hallucinated security commands.

$$C_{\text{total}} = \left(\frac{\sum_{i=1}^{n} w_i \cdot c_i}{\sum_{i=1}^{n} w_i} \right) \times (1 - P_{\text{fp}})$$

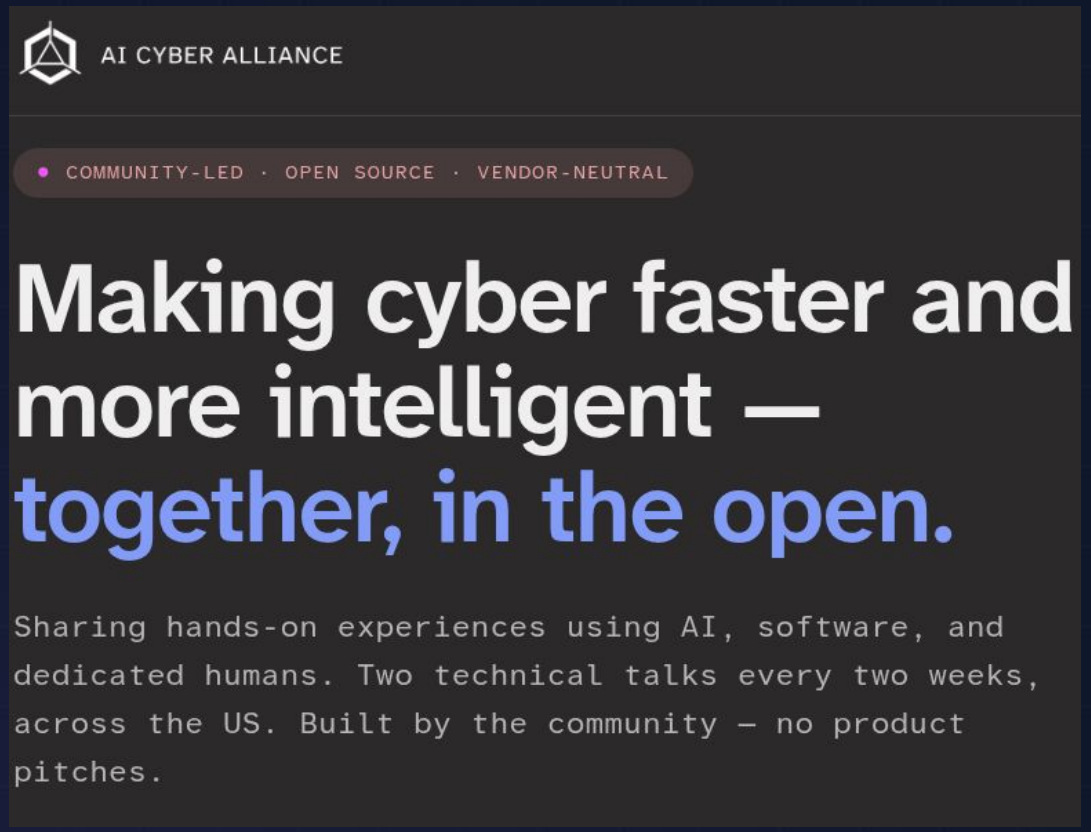
Where:

- $c_i \in [0.0, 1.0]$: Confidence score of evidence factor i (e.g., LLM analysis confidence, threat intel indicator match score, telemetry match score).
- w_i: Assigned weight for evidence component i.
- $P_{\text{fp}} \in [0.0, 1.0]$: Penalty factor based on historical false-positive rates for the rule/detector.

What is the remediation?
Can it be quickly reverted?
Can a case wait?
What are the self-demotion events?

Questions?

Thank you for listening! Connect and find more:



aicyberalliance.org

retroencabulation.com

vigilsoc.org

socbench.org

linkedin.com/in/johnvl