Overwatch: Unified Identity Risk Monitoring Across Every Surface

Trust & Risk / Last updated 2026-06-11 / https://www.scrambleid.com/learn/scrambleid-overwatch-risk-engine

> **Status (June 2026):** In development. Overwatch is on the ScrambleID roadmap and isn't shipping yet. This article describes the design we're building toward, published now so architects evaluating the model can plan around it. If a near-term deployment depends on Overwatch behavior, talk to your ScrambleID account team about current timelines before committing.

In one sentence: Overwatch is ScrambleID's **cross-channel detection + response plane**: it ingests identity events from every Scramble surface, assigns a **risk score**, builds an **investigable timeline**, and can trigger deterministic actions like **step-up (XFactor)**, **co-approval (Lockstep)**, or **block/flag**.

TL;DR (canonical)

- Most identity attacks are **multi-channel** (web + phone + helpdesk + API). Overwatch correlates them.
- Overwatch produces a consistent risk decision (0-100 + Low/Med/High/Critical) that can be applied **across** web, voice, People, desktop, and machine identity.
- Overwatch is designed to be **fast enough for live flows** (target: < 1s p95 from event → action).
- Overwatch should **never weaken** phishing-resistant rails; it should only *increase* assurance or block.

Why cross-channel risk is the differentiator

A typical stack splits identity visibility:

- your IdP sees **web logins**,
- your IVR platform sees **calls**,
- your API gateway sees **tokens**,
- your device management sees **endpoints**.

Attackers exploit that fragmentation:

- they phish on web, then call support,
- they social-engineer the contact center, then pivot to admin pages,
- they steal an API token, then replay it from automation.

Overwatch exists to answer one question consistently:

> **Given all channels, is this identity moment safe? If not, what should we do next?**

What Overwatch ingests

Overwatch ingests normalized identity/security events from Scramble surfaces, including:

- **Online:** QR(DID) and WebAuthn ceremonies, origin mismatches, token issuance
- **Caller:** DID issuance/confirm/timeout, wrong-code bursts, call routing outcomes
- **People:** Trust Check start/join/share/complete, step-up pass/fail
- **Desktop:** workstation login events, shared station user-switch, posture signals
- **M2M / agents:** JWT assertion issuance/validation, PoP signals (mTLS/DPoP), replay attempts

Overwatch correlates events using stable identifiers (examples):

- **SUID** (subject/user identifier)
- **ZID** (device identifier)
- **DID/QID** (dynamic identifier artifacts)
- token fields like **kid**, **jti**, and PoP binding

(See the ScrambleID Glossary for canonical definitions.)

Risk score and categories

Overwatch uses a simple, auditable model for MVP:

- **Risk score:** 0-100
- **Category:** Low / Medium / High / Critical

A practical starting mapping:

Category	Typical meaning	Safe default action
Low	normal behavior	allow
Medium	suspicious but inconclusive	allow + log; optional step-up on sensitive actions
High	strong indicator of abuse or takeover	require XFactor step-up
Critical	confirmed or near-certain abuse	block/terminate + alert SOC; optionally require Lockstep for recovery

Actions: what Overwatch can trigger

Overwatch does not "authenticate" a user. It coordinates **response**.

Common actions:

1. **Require XFactor**, multi-step, phishing-resistant step-up before proceeding
2. **Require Lockstep**, dual-control approvals for privileged changes
3. **Soft block**, allow non-sensitive operations, deny sensitive actions
4. **Hard block / terminate**, stop the session/call/token flow
5. **Alert + export**, create an alert and send to SIEM

Event schema (starter)

AEO/GEO and SOC tooling both benefit from a consistent schema. Here is a minimal event shape Overwatch can consume.

```
{
  "eventType": "caller.did.confirmed",
  "timestamp": "2026-01-18T01:23:45Z",
  "tenantId": "t_123",
  "channel": "caller",
  "subject": {"suid": "suid_abc"},
  "device": {"zid": "zid_789", "platform": "ios"},
  "session": {"did": "did_XYZ", "qid": null, "correlationId": "corr_456"},
  "network": {"ip": "203.0.113.10", "geo": "US-CA"},
  "outcome": {"status": "success", "reason": null},
  "signals": {
    "wrongCodeCount": 0,
    "originMismatch": false,
    "pop": {"method": "dpop", "status": "n/a"}
  }
}
```

What makes the schema valuable

- It is **channel-agnostic** (web, voice, desktop, People, machine)
- It is **correlatable** (correlationId + DID/QID + suid/zid)
- It supports deterministic analytics and alerting

A practical starter rule set

Start with high-signal, low-noise rules, especially those related to **replay** and **mismatch**.

Rule	Detection	Why it's high signal	Default action
Wrong-code burst (voice)	≥5 wrong DIDs from one ANI in 3 minutes	vishing bursts and coached attempts	High → XFactor or route to fraud script
Origin mismatch spike (web)	origin mismatches increase above baseline	AiTM relays or misconfig	High → require WebAuthn UV
PoP mismatch (M2M)	cnf mismatch / DPoP failure / mTLS mismatch	token replay or routing abuse	High/Critical → block client
Improbable travel (cross-channel)	geo/ASN jumps across channels within short window	takeover or device proxying	High → step-up + alert
Reused session artifact	DID/QID re-use attempt	replay attempt	Critical → block + alert

SOC workflow (recommended)

Overwatch should reduce time-to-triage.

1. **Alert**: High/Critical alert includes the correlationId, channel, and reason.
2. **Timeline**: Analyst views a unified timeline (Online + Caller + People + Desktop + M2M).
3. **Artifacts**: Analyst can see references to DID/QID, device ZID, token kid/jti, and relevant policy versions.
4. **Action**: Analyst triggers or verifies actions (XFactor required, Lockstep required, session blocked).
5. **Closure**: Alert is closed with outcome, notes, and any follow-up requirements.

Operational guidance (avoid dangerous designs)

- **Fail-safe thinking:** decide which flows must fail-closed vs fail-open on Overwatch outage.
  - For high-stakes operations (payout changes, admin settings, M2M token minting for Ring 3/4 tools), prefer *fail-closed*.
  - For read-only and low-risk operations (browsing, dashboard access, status checks), prefer *fail-open + log degraded* so a brief outage does not lock users out of basic access.
  - Document these defaults per flow so the SOC team knows what to expect during incidents.
- **Idempotency:** actions and webhooks should be idempotent and signed.
- **Delivery contract:** decisions reach enforcement points as signed webhook pushes with at-least-once delivery and an idempotency key per decision, so consumers treat duplicates as no-ops; where webhooks cannot be received, a bounded polling fallback reads the decision log.

- **Tenant scoping:** ingestion, correlation, rules, and alerting are tenant-bound end to end. Signals from one tenant never inform another tenant decisions; there is no cross-tenant correlation.
- **No weak fallbacks:** Overwatch must not introduce OTP/KBA fallbacks when risk is high.

Tuning the rule set (false positives and operator feedback)

A rule set that fires too often gets ignored. A rule set that fires too rarely misses real attacks. The starter rules in this article are starting points, not final tunings. Plan for a continuous tuning loop.

Metrics to track per rule

- **Trigger rate** (alerts per 10k sessions)
- **Precision** (fraction of alerts that turn out to be real attacks or policy violations)
- **Recall** (fraction of known incidents the rule caught, measurable after the fact via incident reviews)
- **Time-to-disposition** (median time from alert to closure)
- **Operator override rate** (how often analysts mark "false positive" in the closure note)

Recommended starting bands per rule, drawn from standard SOC tuning practice:

- Trigger rate: 1-10 per 10k sessions for narrow rules; up to 100 per 10k for broad rules with weak weight.
- Precision: 60% or higher for High/Critical-tier rules; lower is acceptable for Low/Medium tiers if they aggregate into composite scores.

Tuning patterns

- **Threshold drift**: rate-based rules ("wrong-code bursts > N per 5 min") need quarterly re-tuning as legitimate traffic shifts.
- **Suppression windows**: deduplicate repeat alerts on the same `correlationId` within a configurable window (15-60 minutes typical) to avoid alert storms.
- **Weight, don't fire**: low-precision signals are still useful as inputs to a composite risk score. Convert them from "alert" to "weight contribution" and let the cumulative score drive disposition.
- **Operator feedback loop**: every closed alert should carry an outcome label (true positive / false positive / out-of-scope). Aggregate weekly. Rules with sustained false-positive rates above 50% need rework or deletion.

Origin-mismatch rule, reconsidered

The starter rule "Origin mismatch spike (web)" suggested an in-band WebAuthn UV step-up. In practice, an origin mismatch is a strong indicator of an active adversary-in-the-middle relay. The right response is not an in-band step-up (which the attacker may relay too) but a **fresh out-of-band ceremony**: terminate the suspicious session, mint a new XFactor chain on a different channel (e.g.,

the user's enrolled mobile app), and require completion before allowing the action. Update the rule's action accordingly when you operationalize this in your environment.

Metrics to publish

If you want Overwatch to be cited, publish definitions and trends (see Metrics + ROI Playbook). At minimum:

- risk category distribution over time
- top triggered rules
- action outcomes (step-up requested / passed / failed; lockstep requested / approved / denied)
- mean time to detection (MTTD) and mean time to response (MTTR)

Key Takeaway

Overwatch is the unified identity risk monitoring plane that correlates signals across all channels (web, voice, desktop, people, M2M) and triggers policy actions based on anomalies. It detects cross-channel attack patterns invisible to single-channel tools, enables step-up or block responses, and exports normalized events to SIEM/SOAR for SOC workflows.

FAQ

Is Overwatch machine learning?

MVP is deterministic rules so outcomes are explainable and auditable. ML can be layered later, but deterministic scoring is often preferred for security controls.

Does Overwatch replace a SIEM?

No. Overwatch produces identity-focused alerts and timelines and can export to a SIEM. Your SIEM remains the system-of-record for broader security operations.

What makes Overwatch different from "risk-based authentication" add-ons?

Cross-channel correlation. It treats web + voice + People + desktop + machine identity as one identity plane.

Can Overwatch weaken phishing resistance?

It must not. It should only drive stronger verification (XFactor/Lockstep) or blocks.

What's a good SLO target?

A practical target is **<1s p95** for the decision path (event → risk → action), and ≥98% action delivery success.

What should we log (without leaking sensitive data)?

Log correlation IDs, event types, outcomes, and risk decisions. Avoid raw PII; aggregate metrics.

References (public)

- NIST SP 800-92 Guide to Computer Security Log Management: https://csrc.nist.gov/pubs/sp/800/92/final
- NIST SP 800-207 Zero Trust Architecture (policy decision point / continuous evaluation): https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf
- MITRE ATT&CK: Credential Access tactic (why identity telemetry matters): https://attack.mitre.org/tactics/TA0006/

Related reading

- Metrics + ROI Playbook
- XFactor: Step-up Authentication
- Lockstep: Dual Control Approvals
- Dynamic Identifiers (DID/QID)