ClearSkies™

CORE FUNCTION BRIEF

Threat Hunting

Proactive, hypothesis-driven discovery inside the TDIR core

One of the twelve core functions of the ClearSkies iISOC platform

Find the threat that never tripped an alert, then make sure it never goes unnoticed again.

This document explains how proactive, human-led threat hunting works inside ClearSkies TDIR, how a hypothesis becomes a validated finding and then durable detection the platform runs by itself, and what that is worth to a security team and to a managed service provider. It covers the hunt lifecycle, the capabilities behind it, how a finding is judged and recorded, what hunting contributes to the rest of the platform, how coverage is measured, and the questions a buying team asks about it.

Contents

SECTION 1

Executive Summary

THE ONE-SENTENCE IDEA.

Automated detection finds the threats it already recognizes. Threat Hunting is how ClearSkies iISOC finds the ones it does not, turning a human hypothesis into a validated finding and then into ATT&CK-mapped detection the platform runs automatically from that point on.

The most damaging intrusions are the quiet ones. Attackers work below alert thresholds, use valid credentials rather than malware, and spread activity across systems and weeks so that no single event looks alarming. Automated detection reacts to what it already recognizes. Hunting goes looking for what it does not, by testing how the environment could be compromised right now.

Threat Hunting is not a separate product bolted onto the platform. It is a function of the ClearSkies TDIR core: the same engine that collects, correlates and investigates is the engine a hunter queries directly. In a conventional security operations center, hunting means exporting data into a separate analytics tool and pivoting between the SIEM, the endpoint console and the threat-intelligence portal. Here there is no export, no integration gap and no second console, so every query runs against the full normalized picture rather than a stripped-down copy of it.

What makes the output trustworthy is that a person decides what a pattern means. The platform speeds up the search and supplies the correlations; a hunter judges whether the activity is an early-stage compromise, a detection gap, a policy violation or normal behavior, and that judgment is recorded before anything becomes detection. Proactive discovery therefore never means unproven noise.

And because every validated finding is written back into the engine as a new rule, a tuned baseline and an ATT&CK tag, hunting compounds. A behavior found once by a person is caught automatically afterward, for every tenant. That is what separates hunting as a capability from hunting as an occasional exercise: the coverage it produces stays.

SECTION 2

What Automated Detection Leaves Behind

Detection content is written against behavior somebody has already seen. That is its strength and its limit. Four gaps follow from it, and each is a gap a person has to go looking for.

▸**Activity below the threshold.** Scoring and thresholds exist to keep the queue usable, which means anything deliberately kept small stays under them. A beacon that calls home once every few hours is dismissed as noise at every individual occurrence.

▸**Legitimate tools used illegitimately.** An attacker using valid credentials, a signed system binary and a scheduled task is doing things the environment does every day. Nothing is malicious in isolation; only the sequence is.

▸**The technique nobody wrote a rule for.** Coverage maps show which ATT&CK techniques have detection behind them. The blank spaces are not theoretical: they are the techniques an attacker can use in this environment without anything firing.

▸**Suppressed signal.** A signal scored low-confidence and closed is not the same as a signal that was never there. Without someone testing the suppression, the difference is invisible.

The operational problem is that the team best placed to close these gaps is the team with the least time. Analysts spend their day on the queue, so hunting becomes the thing that happens when nothing else is on fire, which is to say rarely. And when it does happen in a conventional environment, the work is spent assembling data rather than reasoning about it: exporting from one tool, normalizing by hand, pivoting between consoles that disagree about what a user is. The hunt that produces a finding then produces nothing durable, because the result lives in a notebook rather than in the detection engine.

> **THE REFRAME.**
>
> Hunting is worth doing only if it is cheap to start and its output is permanent. Both are architectural properties rather than matters of discipline: cheap to start because the hunter queries the live correlated picture with nothing to export, and permanent because a validated finding is written back into the engine as detection rather than written up as a report.

SECTION 3

How a Hunt Works

A hunt runs in four steps, and the loop closes: what one hunt proves becomes the coverage the next hunt starts from.

> *Figure 1. The hunt lifecycle. A hypothesis becomes a saved query against the live correlated picture, a person validates what it returns, and the finding is written back into the engine as durable detection.*

3.1 The capabilities behind it

CAPABILITY	WHAT IT DOES
Elastic queries in the dashboard	Hunters query normalized telemetry directly from the Threat Hunting dashboard, in free-form and structured searches across identity, endpoint, network, cloud and OT. A saved query runs in the background at a set interval, so a hypothesis becomes a standing hunt that re-evaluates on every new batch of data rather than a one-off search.
Hypothesis-driven method	Each hunt begins with an explicit, testable statement about adversary behavior, drawn from an ATT&CK coverage gap, a risk-scoring trend or threat intelligence. Generative AI helps shape the hypothesis and refine the query, so effort goes where risk exists but alerts do not.
Cross-source exploration	One view over normalized telemetry from every source, with filtering by time, asset, user and ATT&CK technique, and visual correlation across long time ranges to surface what looks ordinary in isolation.
Human interpretation	The hunter reads weak signals and behavioral nuance while the platform supplies context, correlations and summaries. A person decides what the pattern means, which is what keeps judgment, accountability and governance with the team.
ATT&CK mapping of findings	A validated finding is tagged to an ATT&CK tactic and placed under the right technique from inside the dashboard, so it is mapped the moment it is confirmed and slots straight into the platform's coverage map.
Write-back into the engine	Every validated hunt returns to TDIR as new use cases, Sigma rules, tuned machine-learning baselines, risk-score adjustments and ATT&CK coverage, so the behavior is detected automatically thereafter.

3.2 How a finding is judged

The step that makes hunting trustworthy is validation. Each hunt resolves its hypothesis into one of four dispositions, and every disposition is recorded with its reasoning, including the ones that found nothing. A hunt that confirms normal behavior is not a wasted hunt: it sharpens the baseline the next one runs against.

DISPOSITION	WHAT IT MEANS, AND WHAT HAPPENS NEXT
Early-stage compromise	A validated threat. A detection and a response are created in TDIR and the finding is escalated for containment.
Detection gap	Nothing was watching for this. A Sigma rule, a use case or a machine-learning baseline is added so the behavior is caught automatically from now on.

DISPOSITION	WHAT IT MEANS, AND WHAT HAPPENS NEXT
Policy violation	Not an attack, but not permitted. Documented and routed to the owning team, and it may inform hardening.
Benign behavior	Expected activity. Documented to refine baselines and reduce future false positives, so the next hunt starts sharper.

> **DISCOVERY WITHOUT VALIDATION IS ITSELF A RISK.**
>
> Every finding is interpreted by a person, tagged to ATT&CK and written to an immutable record before it becomes detection. That is what keeps escalations trustworthy and detection gaps real, and it lets a team promote a confirmed technique to automated coverage while holding an ambiguous signal for review.

SECTION 4

Threat Hunting in the Platform

The platform brief describes ClearSkies TDIR as the orchestration core through which all detection, investigation and response flows. Threat Hunting is the exploration stage of that same workflow. The correlated, ATT&CK-mapped picture is the input; the hunt is where human reasoning takes over.

Placement is the whole argument. Because hunting lives inside the correlation core rather than beside it, every query runs against the full normalized picture that the iCollector produced at collection time and the ***ClearSkies iISOC Centric-AI Fabric*** resolved to shared entities: user, host, identity, domain, asset and session. The hunter is not querying a copy, a subset or an export. And the finding does not have to be carried anywhere to become useful, because the engine it is written back into is the one already running detection. See the companion documents ***ClearSkies iISOC TDIR*** and ***ClearSkies iISOC iCollector***.

4.1 What a hunt draws on, and what it writes back

▸ **What it draws on.** Normalized telemetry from every source the platform collects, the entity resolution the Centric-AI Fabric performs, the ATT&CK coverage map, risk-scoring trends, and the threat intelligence the platform curates. Where the native add-ons are licensed, their signal is in the same picture: exposure context from ClearSkies ASM, resolution verdicts from ClearSkies DNS Shield, identity context from ClearSkies ITP, and process and user context from ClearSkies ETMR.

▸ **What it writes back.** New use cases and Sigma rules into the Detection Factory, tuned baselines into the machine-learning models, risk-score adjustments, and a technique tag onto the coverage map. Nothing is handed to another component directly: the write-back goes into the engine, which is what makes the result available to every tenant and every later detection at once.

▸ **What it does not do.** Hunting does not act. It produces a validated finding and durable detection; containment is the work of governed playbooks driven by the engine, under the oversight model the platform brief sets out. Keeping discovery and action separate is what keeps the audit trail readable.

4.2 A worked hunt

One hunt, followed end to end. It is chosen because the signal it starts from had already been seen and dismissed, which is the case automated detection cannot solve on its own.

STAGE	WHAT HAPPENS
Hypothesis	A hunter suspects valid credentials are being used below the alert line, prompted by a risk-score trend on a group of privileged accounts rather than by any alert.
Query	The hypothesis becomes an elastic query over identity and endpoint telemetry, saved to re-run every fifteen minutes so it keeps testing itself against new data.
Return	Overnight it flags a privileged sign-in at an unusual hour that the engine had scored low-confidence and closed. On its own it is unremarkable.
Validation	The hunter pulls the surrounding context the engine already holds: identity signal showing the account authenticating from an unfamiliar location, and process history on the destination host showing a discovery command run minutes later. A person reads the sequence and confirms early-stage lateral movement.
Conversion	The finding is tagged to the Lateral Movement tactic under the right technique and written back as a detection, and the response is escalated for containment.
What changed	The same behavior now escalates automatically, in this tenant and every other. A signal the platform had suppressed became mapped coverage, and the suppression itself was corrected.

The pattern holds whatever the starting point. A blank technique on the coverage map becomes a hypothesis about how that technique would look here. A low-and-slow beaconing suspicion becomes a query that lines up sparse outbound connections across weeks that automation dismissed one at a time. In each case the hunt begins where the evidence is thin and ends with coverage that is not.

4.3 Hunting across many tenants

A finding in one customer is a lead. The same behavior across several is a campaign, and no single customer has the vantage point to see it. A provider hunter runs one hypothesis across the whole book of business from the cross-tenant dashboard, saved to re-run on an interval. In isolation each tenant shows a faint trace; together the pattern is one actor using the same tradecraft against several customers. The hunter validates it once, maps it once, and promotes the result to a managed detection applied to every tenant at once, with per-tenant scope separation preserved throughout. That is the difference between hunting that scales with headcount and hunting that scales with the book. See the companion document ***ClearSkies iISOC MSSP***.

SECTION 5

ATT&CK Alignment and Supported Frameworks

Two audiences read a hunt differently. A SOC manager needs to know which techniques now have detection behind them and which do not. An auditor or a board committee needs to know that the programme is finding things deliberately rather than by luck. Both come out of the same record, because every validated finding is tagged the moment it is confirmed.

WHAT ATT&CK IS USED FOR	HOW IT WORKS IN THREAT HUNTING
Choosing the hypothesis	The coverage map is a source of hunts in its own right. A technique with no detection behind it is a hypothesis waiting to be written, and hunting is how a blank space on the map gets closed.
Tagging the finding	A validated finding is converted to a tactic and placed under the right technique from inside the dashboard, rather than mapped afterward in a spreadsheet.
Reporting coverage	Coverage is reported as techniques with detection behind them, and the hunts that put it there are traceable. A technique covered because a hunt proved it is a stronger claim than a technique covered because a rule was written for it.
Setting the next cycle	Gaps that remain after a hunt are the input to detection engineering, so the map drives the work rather than describing it afterward.

The same record serves the governance frameworks a programme is assessed against. NIST CSF asks for continuous monitoring and improvement under Detect and Identify, and a documented hunt cycle with recorded dispositions is direct evidence of both. ISO/IEC 27001 asks for technical vulnerability and threat management performed and reviewed, which the disposition record supplies. NIS2 and DORA

both ask for demonstrable detection capability rather than a stated one, and the coverage map with hunts behind it is what demonstrates it. The platform produces what an auditor asks for; it does not make an organization compliant.

SECTION 6

Business Impact and Delivery

Threat Hunting turns expertise into coverage that stays. The change is measurable, and it is measurable in the customer's own environment rather than as a platform-level claim.

METRIC	DEFINITION
Hunts run per period	Hypotheses tested in the period, split between one-off queries and saved queries running on an interval.
Dispositions recorded	Findings by disposition: early-stage compromise, detection gap, policy violation and benign behavior. The mix matters more than the total.
Detections created from hunts	Validated findings written back as a rule, a use case or a tuned baseline, as a share of hunts run.
ATT&CK techniques newly covered	Techniques that gained detection in the period because a hunt proved them, reported against the coverage map.
Time from hypothesis to detection	Elapsed time from a hypothesis being written to the resulting detection being live in the engine.

6.1 What changes, and for whom

ROLE	THE MEASURE THAT CHANGES
CISO	Time to discover a threat that never raised an alert. Hunting is the only mechanism that addresses it, and because every finding becomes detection, the same threat is not discovered twice.
SOC manager	Coverage per unit of analyst effort. A hunt costs a person's time once and returns detection that runs by itself, so the team's capacity compounds rather than resets each week.

ROLE	THE MEASURE THAT CHANGES
Detection engineering	Where the next rule comes from. Hypotheses are drawn from the coverage map and the risk trends, so the backlog is driven by evidence rather than by the last incident.
Risk and compliance	Demonstrable rather than stated detection capability. Every hunt, its disposition and its reasoning are recorded, which is the evidence a regulator or an auditor asks for.
Service provider	Margin per tenant served. One validated hunt becomes detection across the whole book of business, so hunting is delivered as a differentiated service without headcount rising in step with customers.

6.2 What sets it apart

▸ **Hunting inside the engine, not beside it.** The hunter queries the live correlated picture with nothing exported and no second console. Where hunting means extracting data into a separate analytics tool, the hunt is limited to what was extracted, and it is stale the moment it is.

▸ **Every finding becomes permanent coverage.** A validated hunt is written back as a rule, a baseline and an ATT&CK tag. Where the output of a hunt is a report, the knowledge leaves with the person who wrote it.

▸ **A person decides, and the decision is recorded.** Four dispositions, each with its reasoning, held in an immutable record. That is what makes an escalation trustworthy and a detection gap real, and it is what an auditor can read.

▸ **Standing hunts, not occasional exercises.** A saved query re-runs on an interval against new data, so a hypothesis keeps testing itself. Hunting stops being the thing that happens when the queue is quiet.

All four hold for a single organization. For a provider, one further property applies: because the write-back goes into the shared engine, a hunt validated once is detection for every tenant, which is what makes hunting a service rather than an engagement.

6.3 Objection handling

▸ **Do we need hunters, or does the platform do it?** The platform does the searching and the correlation; a person forms the hypothesis and judges what the result means. Threat Hunting is designed around that division rather than around removing it.

▸ **Our team has no time for hunting.** That is the usual state, and it is why the hunt is a saved query rather than a session. A hypothesis written once keeps running in the background, and the cost of starting is a query rather than a data export project.

▸ **How is this different from running searches in a SIEM?** The search runs against the correlated, entity-resolved picture rather than raw logs, and the finding is written back into the engine as detection rather than pasted into a report (Section 4, Threat Hunting in the Platform).

▸ **How do we know a hunt found something real?** Every finding carries a disposition decided by a person, its reasoning, and an ATT&CK tag, recorded before it becomes detection (Section 3.2, How a finding is judged).

SECTION 7

Summary

Threat Hunting is how the platform looks for what it has not been told to find. It runs inside the TDIR core rather than beside it, so a hypothesis becomes a query against the live correlated picture with nothing exported. A person judges what comes back and records the judgment. The finding is tagged to ATT&CK and written into the engine as detection. And the next hunt starts from the coverage the last one created.

▸ **Proactive.** It tests how the environment could be compromised now, rather than waiting for behavior somebody has already written a rule against.

▸ **Native.** The engine that detects and correlates is the engine the hunter queries, so there is no export, no integration gap and no second console.

▸**Human-validated.** Four dispositions, each decided by a person and recorded with its reasoning, so proactive discovery never means unproven noise.

▸ **Compounding.** Every validated finding becomes a rule, a baseline and a technique tag, so a behavior found once is detected automatically thereafter.

▸ **Multi-tenant.** One hunt validated once becomes detection across every tenant, which is what lets a provider deliver hunting as a service.