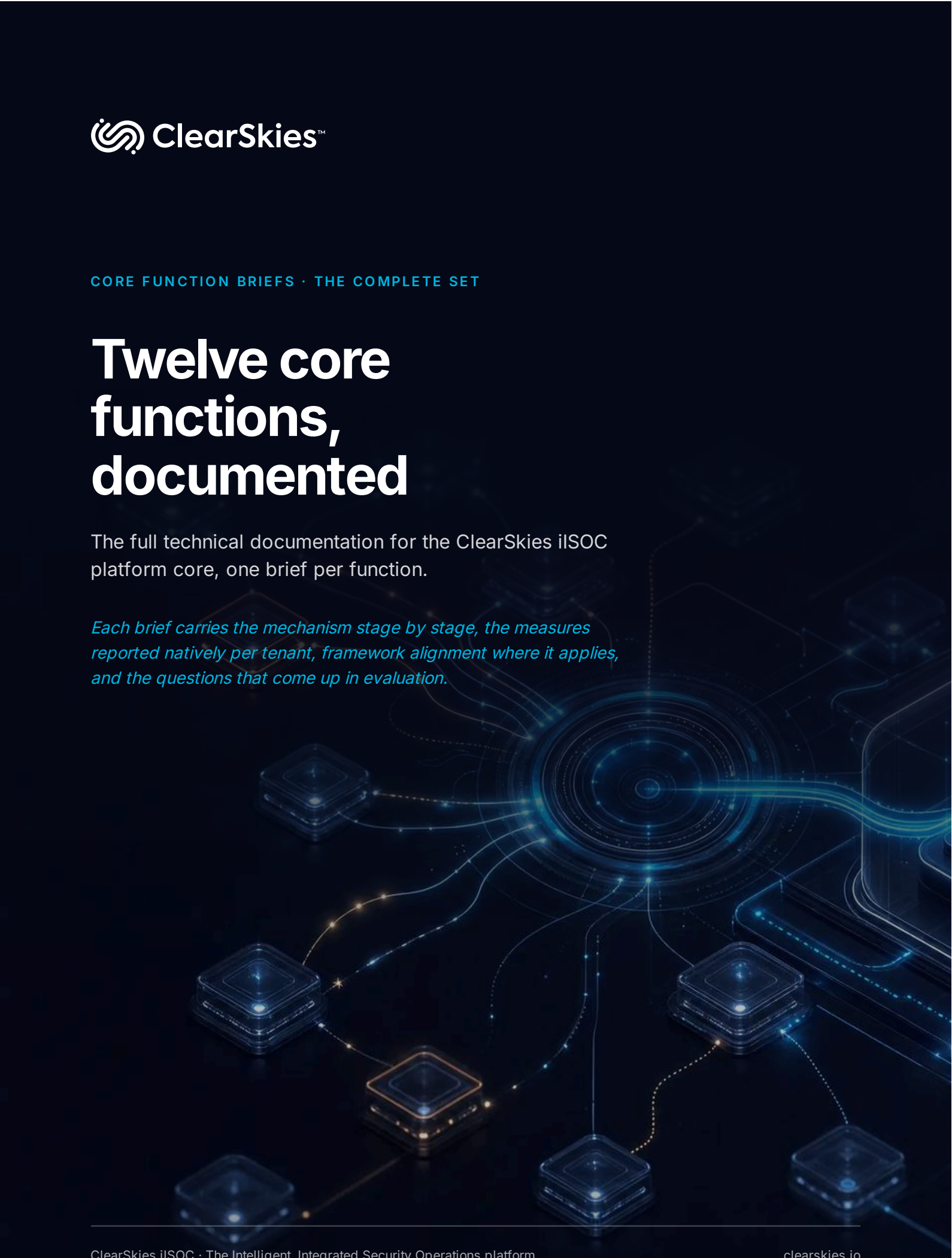
ClearSkies™

CORE FUNCTION BRIEFS · THE COMPLETE SET

Twelve core functions, documented

The full technical documentation for the ClearSkies iISOC platform core, one brief per function.

Each brief carries the mechanism stage by stage, the measures reported natively per tenant, framework alignment where it applies, and the questions that come up in evaluation.



What is in this pack

The twelve functions are the platform core. Each owns exactly one decision in the security lifecycle and hands off to the next, which is what allows an incident to be replayed end to end from a single system. The briefs are ordered by that lifecycle rather than alphabetically, so the pack reads as one argument if you read it straight through.

SEE WHAT IS HAPPENING

Threat Intelligence	Version 3.2 · 11 pages
UEBA	Version 1.1 · 14 pages
Threat Hunting	Version 2.0 · 11 pages

DECIDE WHAT MATTERS, AND WHO WORKS IT

Detection Factory	Version 3.0 · 11 pages
Risk Scoring Formula	Version 3.1.1 · 12 pages
Intelligent Alert Routing	Version 3.2 · 12 pages

ACT ON IT, UNDER GOVERNED AUTHORITY

Generative and Agentic AI	Version 3.1 · 12 pages
SOAR	Version 3.2 · 11 pages
Playbooks	Version 3.2 · 11 pages

PROVE IT, AND EXTEND IT

KPIs and SLAs	Version 3.2 · 11 pages
Regulatory Frameworks	Version 3.2 · 12 pages
Third-Party Add-ons Marketplace	Version 3.2 · 12 pages

> **Scope of this pack.** A core function is part of the platform, so no brief in this pack carries licensing, tier or packaging material. For editions and pricing, and for the six native add-ons licensed separately, see the ClearSkies iISOC Platform Concept and Architecture brief.

ClearSkies™

CORE FUNCTION BRIEF

Threat Intelligence

A core function of the ClearSkies iISOC platform

Indicators and named threats, tagged, rated and republished as the evidence changes.

Contents

SECTION 1

Executive Summary

IN ONE SENTENCE.

Threat Intelligence is the core function that collects, catalogs, assesses and alerts on indicators and named threats from global sources, publishing each with a confidence and severity rating that is republished as the underlying evidence changes rather than set once and left.

A raw indicator on an external feed rarely says what it means. An address or a hash on its own does not say what campaign it belongs to, how fresh the sighting is, or whether another source corroborates it, so an analyst has to research each one by hand before deciding whether to act. Different feeds tag and rate the same indicator differently, a global surge in one attack type is not obviously relevant to a specific tenant until someone checks, and a severity assigned once is rarely revisited as the picture around it changes.

Threat Intelligence answers this with a five-stage curation mechanism. Indicators are tagged by type and category the moment they are collected from probes and third-party feeds, cataloged by feed, recency and geographic pattern, assessed against a defined severity taxonomy, and issued as an alert carrying impact, affected systems and recommendations. The rating attached to a published indicator or named threat updates continuously, republished as new corroboration or exploitability information arrives.

Global visibility only becomes actionable once it is checked against a specific tenant's own indicator history, so a worldwide pattern is surfaced as locally relevant rather than left as an undifferentiated feed dump.

THE FUNCTION IS PRECISELY SCOPED.

Threat Intelligence collects, curates and rates indicators and named threats. It does not author or validate detection logic, which is Detection Factory. It does not compute the risk score, which is the Risk Scoring Formula. And it does not decide or execute a response, which is SOAR and Playbooks. Each of those is a separate core function with its own brief.

Five stages
Collect, catalog, assess, alert, publish

Global and local
Worldwide patterns checked against tenant exposure

Never frozen
Confidence and severity are re-rated

SECTION 2

Why Unmanaged Feeds Fall Short

Threat intelligence consumption is underinvested relative to the rest of the security operation, and the consequences surface as missed context rather than as a single visible failure. Four failure modes explain why, and each is answered by a stage of the mechanism in section 3.

2.1 Indicators without context

A raw indicator on a feed does not say what campaign or vulnerability it belongs to, how fresh the sighting is, or what to do about it, so an analyst has to research each one by hand before deciding whether it matters.

2.2 Severity that is not consistently applied

Without a shared level taxonomy, one analyst's critical is another's medium, so the same class of threat is rated inconsistently across an alert history depending on who assessed it.

2.3 Global patterns invisible at the tenant level

A worldwide surge in a specific attack type is only actionable once a tenant can see whether their own indicators show the same pattern, and a raw feed dump does not distinguish between trending everywhere and specifically probing this environment.

2.4 Advisories that go stale

A severity assigned once, to an indicator or to a named vulnerability, is rarely revisited as exploitability and affected-system information change, so an old critical rating can outrank a genuinely current threat simply because nobody downgraded it.

2.5 The cost of the workaround

The usual response is an analyst cross-referencing several external feeds and vendor advisories by hand before triaging anything, a task that scales with the number of feeds subscribed to and does not get easier as feed volume grows. The underlying problem, indicators and advisories that are not curated, rated and kept current from one place, is untouched.

Industry research on security operations consistently finds analyst time lost to cross-referencing threat feeds by hand, and a meaningful share of indicators reaching a security operation uncorroborated by a second source, though the specific figures are not yet sourced for this brief.

SECTION 3

How Threat Intelligence Works

Threat Intelligence is one continuously operating mechanism native to the TDIR core, not a feed subscription a person monitors. Raw indicators and vulnerability disclosures enter at one end, and a curated, rated, continuously refreshed advisory leaves at the other.

> *Figure 1. The five-stage curation mechanism.*

3.1 The five stages

Each stage is named once here and referenced by name throughout the rest of this brief, paired with the failure mode from section 2 that it answers.

STAGE	WHAT HAPPENS	THE FAILURE MODE IT ANSWERS
Collect	Indicators of attack are ingested in real time from global probe telemetry and third-party feeds, tagged by type and category at the point of intake.	Indicators without context
Catalog	Indicators are organized by feed source, recency and multi-source activity, and compared across local and global scope, so a worldwide pattern can be checked against a tenant's own indicator history.	Global patterns invisible at the tenant level
Assess	A severity is set against a defined level taxonomy. Where the threat is a named vulnerability, exploitability, classification and affected systems are attached alongside its severity scoring.	Severity that is not consistently applied
Alert	The assessed threat is issued as an advisory carrying its title, description, impact, affected systems and recommendations, and is tracked in the alert history.	Indicators without context
Publish	The curated indicator or named threat, with its rating, is published into the shared model, and is republished as new corroboration or exploitability information arrives.	Advisories that go stale

The stages run in sequence, but Publish is never final: a rating set at Assess is revisited the moment Collect or Catalog surfaces new evidence bearing on the same indicator or threat.

3.2 How a rating is judged

Quality is measured rather than assumed, on definitions a provider can report and a customer can audit. The measures below are reported natively and per tenant. They are defined here because the same metric name is measured differently by different vendors, and target values are agreed per engagement rather than asserted.

MEASURE	THE DEFINITION USED	WHAT A POOR VALUE TRIGGERS
Indicator freshness	Published indicators reconfirmed within a defined recency window, as a share of the catalog.	Prioritized re-collection, or retirement of the stale indicator.
Multi-source corroboration	Published indicators confirmed by more than one feed, as a share of the catalog.	A confidence downgrade for single-source indicators, pending corroboration.
Severity consistency	Alerts whose assigned severity matches the defined criteria for that classification, as a share of alerts reviewed.	A review of the rating against the defined taxonomy.
Time to publish	Elapsed time from an indicator's first collection to being published with confidence and severity attached.	Investigation of the stage where the latency is introduced.
Local relevance rate	Published threats confirmed to touch a specific tenant's own indicator history, as a share of all published threats.	A review of the matching criteria against that tenant's indicator history.

A downgraded rating is the mechanism working, not a mistake to explain away. ***An indicator whose confidence falls as corroboration fails to materialize, or a severity that drops as exploitability information is updated, is Publish doing exactly what it is built to do. Treating every downgrade as an error to investigate misreads a rating that is supposed to move.***

SECTION 4

Threat Intelligence in the Platform

4.1 What it draws on, what it writes back, and what it does not do

What it draws on. Real-time probe telemetry and third-party threat-intelligence feeds as external sources, and the normalized entity model the Centric-AI Fabric carries, used to check whether a global pattern touches a specific tenant's own infrastructure. Each is an input to a specific stage rather than a general capability the function generates itself.

What it writes back. The curated indicator or named threat, with its confidence and severity rating, published into the shared model. The Detection Factory's source stage consumes it as raw material for new detection content, and the Risk Scoring Formula's threat-intelligence dimension consumes the rating directly. The relationship is one way: nothing returned to Threat Intelligence changes a rating already published, and a later re-rating comes from new evidence arriving at Collect or Catalog.

What it does not do. It does not author or validate detection logic, compute a risk score, or decide or execute a response. Threat Hunting may draw on the same curated indicators when forming a hypothesis to test, but Threat Intelligence does not itself investigate a tenant's environment. It holds no tenant telemetry beyond what is needed to check local relevance against the Centric-AI Fabric's model.

4.2 A worked example

The chain below is illustrative and technically representative rather than drawn from a named engagement. A vulnerability affecting a widely used VPN appliance is disclosed externally. Collect ingests the first exploitation indicators tied to it, tagged to the disclosure. Catalog links the pattern to the tenant's own indicator history and confirms probes from a known campaign region reaching the tenant's edge, not only a global trend. Assess assigns severity using the published CVSS metrics and the defined level taxonomy, noting the affected systems and versions. Alert issues the advisory with impact and recommendations. Publish sends the confidence-scored, named threat into the shared model, where the Detection Factory drafts a detection for the exploitation pattern and the Risk Scoring Formula's threat-intelligence dimension reflects the new campaign.

4.3 Delivery across many tenants

A threat curated once from global sources is checked against every tenant's own indicator history independently, so the same advisory can be locally relevant to one tenant and merely informational to another. What travels between tenants is the curated indicator and its global assessment; what never travels is one tenant's own indicator history or infrastructure exposure.

Because the function is switched on with the platform rather than deployed separately, time to value is a matter of feed coverage rather than an onboarding project. In the first week, global probe telemetry and the feeds already connected begin populating the catalog, and a tenant's own indicator history starts accumulating the moment the platform is live. Through the first month, local-relevance matching sharpens as that history deepens. At steady state, indicator freshness and multi-source corroboration are tracked as feed-health measures in their own right, reviewed whenever a source is added or retired.

SECTION 5

ATT&CK and Intelligence-Exchange Alignment

Threat Intelligence does not map detection content and does not compute technique relevance itself. Where an indicator or named threat is later translated into detection logic, the Detection Factory attaches the ATT&CK technique identifier at that point. This function's own contribution is the indicator, its confidence and its named-threat context, not the technique mapping.

FRAMEWORK OR STANDARD	ITS ROLE IN THREAT INTELLIGENCE
STIX and TAXII	Structured intelligence exchange for indicator and named-threat intake.
CVE and CVSS	Vulnerability identification and severity scoring for named threats in the vulnerability-oriented advisory catalog.
MITRE ATT&CK, Enterprise, Cloud and Identity	Carried forward once the Detection Factory translates a curated indicator into detection content. It is not mapped or scored by this function.
NIST Cybersecurity Framework 2.0	Outcome language used where platform reporting discusses threat-landscape posture to an executive or board audience.

SECTION 6

Questions Raised in Evaluation

The questions below recur in buyer evaluations, each answered directly against the mechanism described in section 3.

QUESTION, AS A BUYER ASKS IT	ANSWER
Does Threat Intelligence compute the risk score or author detections?	No. Those are the Risk Scoring Formula and the Detection Factory. Threat Intelligence supplies the indicator, its confidence and the named-threat context those functions consume.
How is severity decided?	Against the defined levels the platform's Threat Level Manager maintains, with CVSS metrics attached where the threat is a named vulnerability.

QUESTION, AS A BUYER ASKS IT	ANSWER
What stops one feed's noisy indicator from being treated as high confidence?	Multi-source corroboration is tracked. An indicator confirmed by only one feed is rated accordingly.
Does a global threat automatically apply to every tenant?	No. Global visibility is checked against a tenant's own indicator history before anything is surfaced as locally relevant.
How does an outdated severity rating get corrected?	Publish is continuous. A rating updates as new corroboration or exploitability information arrives, rather than staying frozen at first assessment.
Can a tenant or provider add its own threat entries?	Yes. An authorized user can add a named threat or adjust a severity level, which then flows through the same Assess and Alert stages as an externally sourced one.

SECTION 7

The Benefits

Curated intelligence is where a feed subscription becomes an operational input, and the outcomes are countable: how much analyst time goes to judgment rather than cross-referencing, whether a rating can be trusted to be current, and whether a global pattern can be shown to touch this particular environment.

7.1 At a glance

BENEFIT	WHAT PRODUCES IT
An advisory arrives ready to act on rather than ready to research	Tagging at Collect and the impact, affected systems and recommendations attached at Alert.
The same class of threat is rated the same way every time	One severity taxonomy applied at Assess, rather than an individual analyst's judgment.

BENEFIT	WHAT PRODUCES IT
A rating can be trusted to be current	Publish is continuous, so confidence and severity move as corroboration and exploitability information arrive.
A worldwide pattern is separated from a local one	Catalog checks the global picture against each tenant's own indicator history before anything is surfaced.
Feed quality becomes visible rather than assumed	Indicator freshness and multi-source corroboration, defined in section 3.2 and reported per tenant.
One curation effort serves every tenant	A threat curated once from global sources is matched for local relevance independently, per tenant.

7.2 What changes, and for whom

Curated, continuously current intelligence changes what each role spends time on, and what each role can trust.

ROLE	WHAT CHANGES
SOC analyst	Advisories arrive with context, severity and recommendations already attached, rather than as a raw feed entry to research from scratch.
SOC manager	Indicator freshness and corroboration become visible, manageable numbers rather than a general sense of whether the feeds are trustworthy.
CISO and risk	Every severity rating traces to a defined, consistent taxonomy, which supports board and audit reporting on threat-landscape posture.
Service provider practice lead	A global threat curated once is checked for local relevance across every tenant independently, without a separate curation effort per customer.

7.3 What sets it apart

▸ **Tagged at the point of collection, not after.** Type and category are attached when an indicator first arrives, rather than reconstructed later by an analyst.

▸ **Global visibility, tenant-scoped relevance.** A worldwide pattern is surfaced as actionable only once it is checked against a tenant's own indicator history.

▸ **Continuously republished, never frozen.** Confidence and severity update as new corroboration or exploitability information arrives, which is what keeps an old critical rating from outranking a current threat.

THE VALUE IN ONE LINE.

An unmanaged feed hands a security operation a list of indicators and leaves the meaning to an analyst. A curated one hands over a rated, named, locally checked threat, and keeps the rating honest as the evidence moves.

SECTION 8

Summary

Threat Intelligence is how the platform turns raw external indicators and vulnerability disclosures into curated, rated, continuously current advisories. It is one core function among twelve, scoped to a five-stage curation mechanism that runs continuously inside the TDIR core rather than as a feed subscription someone monitors by hand.

▸ Collect, catalog, assess, alert and publish replace unmanaged, context-free feed consumption.

▸ Every indicator is tagged at collection, and every advisory is rated against one consistent severity taxonomy.

▸ Global patterns are checked against each tenant's own indicator history before being surfaced as locally relevant.

▸ Every published rating stays live, revisited as new evidence arrives rather than frozen at first assessment.

▸ The function curates and rates threat data, and leaves detection authoring, scoring and response to the core functions built for those decisions.

Forward-looking statement: any statement about future capability is forward-looking and non-binding, and capability described in the present tense is capability available at the publication date, subject to the deployment configuration in force.

ClearSkies™

CORE FUNCTION BRIEF

UEBA

User and Entity Behavior Analytics, a core function of the ClearSkies iISOC platform

Detection that learns what normal looks like for each person, then notices when it breaks.

Contents

SECTION 1

Executive Summary

IN ONE SENTENCE.

UEBA is the core function that learns what normal activity looks like for each individual user and entity, then raises a detection when live behavior breaks that pattern, when an observed value matches known-malicious threat intelligence, or when correlation reveals what a single log source could not show on its own.

Signature-based detection recognizes what it has already been told to look for. That is its strength and its limit. A tool an attacker wrote yesterday has no signature. A set of actions that is entirely ordinary for an administrator can be the clearest possible warning sign when performed by someone in accounts payable. And an endpoint alert that names a device but not a person costs an analyst time at exactly the moment speed matters most.

UEBA answers this by learning rather than matching. Over a training period, the function builds a personal profile of normal activity for each user: which programs they run, which destinations they contact, and at what volume. Once that baseline is established, activity outside it raises a detection, whether or not the behavior has ever been seen anywhere before. Alongside the learned baselines, the function checks observed values against curated threat intelligence and correlates alerts to the user actually responsible for them.

The catalog currently comprises eleven detections across four categories, built on three analytic methods. Each detection publishes into the same shared model every other core function reads, so a behavioral finding is scored, routed and acted on through the same pipeline as any other detection rather than in a separate console.

THE FUNCTION IS PRECISELY SCOPED.

UEBA produces behavioral detections. It does not compute how urgent one is, which is the Risk Scoring Formula. It does not decide who works it, which is Intelligent Alert Routing. It does not decide or execute a response, which is SOAR and Playbooks. And it does not curate the threat intelligence its reputation-matching detections consume, which is Threat Intelligence. Each of those is a separate core function with its own brief.

11 detections

Across four behavioral and reputation categories

3 analytic methods

Learned baseline, reputation match, correlation

Per-user normal

Not one global threshold applied to everyone

SECTION 2

Why Signature Matching Falls Short

Detection content written against behavior somebody has already seen cannot, by construction, recognize behavior nobody has seen yet. Four failure modes follow, and each is answered by a named stage of the mechanism in section 3.

2.1 Threats with no signature to match

Attackers routinely bring in custom tooling, scripts, or malware built for a single campaign. Nothing about it appears on any list, so a detection approach that works by comparison against known-bad has nothing to compare against and stays silent.

2.2 Activity that is only unusual for this specific person

Running a remote administration tool is unremarkable for an IT administrator and a serious warning sign for someone in a finance role. A single global threshold applied across an organization cannot express that difference, so it is set either loose enough to miss the finance case or tight enough to bury the administrator in noise.

2.3 Alerts that do not name a person

An endpoint tool raises an alert about a device. Working out who was actually logged in and active at that moment is manual work an analyst performs under time pressure during an active incident, and the answer determines who to contact and what to contain.

2.4 Checks tied to a single log source

A reputation check implemented inside one product only inspects that product's logs. The same malicious destination contacted through a different system, and recorded in a different log, passes unexamined simply because nothing was looking at that log.

2.5 The cost of the workaround

The usual response is more rules and tighter thresholds, which increases alert volume without addressing any of the four gaps, and a manual enrichment step where an analyst reconstructs identity and context by hand. Cost scales with alert volume and with the number of log sources in scope, while the underlying problem, detection that has no notion of what is normal for a specific person, is untouched.

The scale of that volume is measurable. Microsoft and Omdia, in the State of the SOC, 2026, put the false-positive share of alerts at 46%, and found that highly fragmented security operations spend around 40% more on operational labor than consolidated ones. Neither figure is improved by adding rules, which is the response the four gaps above usually attract.

The share of intrusions that turn on valid credentials or previously unseen tooling, the two things a behavioral baseline is built to catch, is not sourced for this brief.

SECTION 3

How UEBA Works

UEBA is one continuously operating mechanism native to the TDIR core, not a separate analytics product. Normalized activity enters at one end, and a behavioral detection carrying its method, its category and the responsible user leaves at the other.

> *Figure 1. The five-stage behavioral analytics mechanism.*

3.1 The five stages

Each stage is named once here and referenced by name throughout the rest of this brief, paired with the failure mode from section 2 that it answers.

STAGE	WHAT HAPPENS	THE FAILURE MODE IT ANSWERS
Baseline	Over a defined training period, a personal profile of normal activity is built for each user and entity: the programs they run, the destinations they contact, and the volume at which they contact them.	Activity that is only unusual for this specific person
Resolve	Observed activity is attached to the user or entity actually responsible for it, drawing on the normalized entity model the Centric-AI Fabric maintains.	Alerts that do not name a person
Evaluate	Live activity is tested by the method applicable to it: deviation from the learned baseline, a volume anomaly against that baseline, a match against curated threat intelligence, or a similarity comparison against the organization's own trusted domains.	Threats with no signature to match
Detect	A detection is raised carrying its analytic method, its category and the resolved user, so a reviewer can see not only what fired but on what analytic basis.	Checks tied to a single log source

STAGE	WHAT HAPPENS	THE FAILURE MODE IT ANSWERS
Publish	The detection is written into the shared model, where the Risk Scoring Formula scores it and Intelligent Alert Routing assigns it. Baselines are refreshed as legitimate behavior changes.	Threats with no signature to match

Reputation-matching detections skip the learned baseline entirely and enter at Evaluate, because a known-malicious value is malicious regardless of whose behavior it appears in. Every detection still passes through Resolve and Detect, so a reputation match is attributed to a person exactly as a baseline deviation is.

3.2 The three analytic methods

Every detection in the catalog uses one of three methods, and this brief names which applies to each, because the methods carry different evidentiary weight.

METHOD	HOW IT DECIDES	WHAT IT IS GOOD AT, AND WHAT IT IS NOT
Learned per-user baseline	Compares live activity against a profile of that specific user's normal, built over a training period.	Catches previously unseen tooling and behavior. Requires a completed training period before it produces useful output.
Reputation and similarity matching	Compares an observed value, an IP address, a domain or a full web address, against curated threat intelligence or against the organization's own trusted-domain list.	Immediate coverage with no training period, and independent of which log recorded the value. Limited to what intelligence already knows, except for the similarity check, which catches an unreported look-alike domain on first use.
Correlation	Attaches identity and context to an alert another tool has already raised.	Removes manual enrichment work. Adds no new detection of its own; it makes an existing one actionable.

3.3 How a baseline is judged

Quality is measured rather than assumed, on definitions a provider can report and a customer can audit. The measures below are reported natively and per tenant, defined here because the same metric name is measured differently by different vendors. Target values are agreed per engagement rather than asserted.

MEASURE	THE DEFINITION USED	WHAT A POOR VALUE TRIGGERS
Baseline maturity	In-scope users and entities with a completed training period, as a share of the total.	A review of coverage gaps, and of whether the training period suits that population.
First-seen rate stability	The trend in first-seen detections per user after training completes, rather than during it.	Investigation of whether the baseline is converging or the training window is too short.
Identity resolution rate	Endpoint alerts attached to a responsible user, as a share of all endpoint alerts received.	A review of the identity and session data available for the affected systems.
Detection mix by method	The distribution of raised detections across learned baseline, reputation matching and correlation.	A check on whether baselines are contributing at all, or reputation matching is carrying the function alone.
Source coverage	Log sources in scope for the reputation-matching detections, as a share of ingested sources carrying a checkable value.	A review of which ingested logs are being left unexamined.

A first-seen detection is not by itself a finding. ***Every user legitimately runs a new program or contacts a new destination sometimes. A first-seen detection states that something has not been observed before for this person, which is a reason to look rather than a conclusion. The Risk Scoring Formula weighs it alongside everything else known about the entity, which is why UEBA publishes rather than escalates.***

SECTION 4

UEBA in the Platform

4.1 What it draws on, what it writes back, and what it does not do

What it draws on. Normalized telemetry from the sources the platform collects: Windows event and process logs delivered through Endpoint Threat Monitoring and Response, firewall logs, web gateway logs, VPN authentication logs, and alerts raised by endpoint detection tools. It draws the normalized

entity model the Centric-AI Fabric maintains, used at Resolve, and the curated indicators Threat Intelligence publishes, used by the reputation-matching detections at Evaluate. Each is an input to a specific stage rather than a general capability UEBA generates itself.

What it writes back. Behavioral detections, each carrying its analytic method, its category and its resolved user, published into the shared model. The Risk Scoring Formula scores them, Intelligent Alert Routing assigns them, and where a detection is confirmed, SOAR decides the response. The relationship is one way: nothing returned to UEBA changes a detection already published, and a baseline changes only because observed behavior changed, never because a downstream function asked for it to.

What it does not do. It does not compute a risk score, decide assignment or execute a response. It does not curate the threat intelligence its reputation-matching detections consume, which is Threat Intelligence, and it does not author the platform's rule-based detection content or attach ATT&CK technique identifiers, which is Detection Factory. Where a hunter tests a hypothesis against the same telemetry, that is Threat Hunting: UEBA runs continuously and unprompted rather than in response to a hypothesis.

4.2 The detection catalog

The catalog currently comprises eleven detections across four categories. The table states the analytic method behind each and the telemetry it reads, because the method determines what a detection can be relied on to catch and the source determines where it can catch it.

CATEGORY	DETECTIONS	METHOD AND TELEMETRY
Endpoint and process behavior	Security-oriented event detection; first seen process; malicious endpoint activity	Rule-based against a published event catalog, learned baseline, and correlation respectively, over Windows event and process logs and endpoint detection alerts
Network behavior	First seen IP address; unusual network activity by IP address; unusual network activity by destination port	Learned baseline, and volume anomaly against that baseline, over firewall logs and endpoint network visibility
Threat intelligence and reputation	Malicious IP address; malicious domain; malicious URL; domain similarity	Reputation matching against any ingested log carrying the relevant field, and similarity comparison against the organization's own trusted domains over web gateway logs
Identity and access	Geolocation logins	Impossible-travel check against consecutive authentication events in VPN logs

The reputation-matching and similarity detections apply to any ingested log containing the relevant value rather than to one product's logs, which is what gives them coverage across the environment. The specific VPN and web gateway products currently in scope are a deployment matter rather than a fixed platform property.

4.3 A worked example

The chain below is illustrative and technically representative rather than drawn from a named engagement. A finance user's workstation runs a remote administration utility for the first time. Baseline has a completed profile for that user and has never observed the program, so nothing about it matches their normal. Resolve attaches the detection to the specific person logged in at that moment rather than to the device alone. Evaluate raises a first seen process detection, and separately, when the same session contacts an external destination, a first seen IP address detection. Detect records both with their method and category. Publish writes them into the shared model, where the Risk Scoring Formula weighs two independent first-seen signals on a privileged-data-adjacent account and scores the pair well above either alone. For an IT administrator, the identical program would have sat inside an established baseline and raised nothing, which is the whole point of a personal profile.

4.4 Delivery across many tenants

A baseline is personal by construction: it describes one user in one tenant and has no meaning outside that tenant. What travels between tenants is the analytic method and the detection catalog; what never travels is one tenant's baselines, telemetry or detections. A provider therefore runs the same eleven detections across every customer while each customer's notion of normal stays entirely its own.

Because the function is switched on with the platform rather than deployed separately, time to value has two distinct phases, which this brief states plainly rather than blurring. In the first week, the reputation-matching and correlation detections produce output immediately, since they require no training period. Through the first training cycle, the learned-baseline detections are still observing and produce a higher first-seen rate that settles as profiles mature; a tenant is told to expect this rather than surprised by it. At steady state, baseline maturity and first-seen rate stability are tracked as ongoing measures.

SECTION 5

ATT&CK Alignment and Supported Frameworks

UEBA does not map detection content to technique identifiers itself. Where a behavioral detection is translated into mapped detection content, Detection Factory attaches the ATT&CK identifier at that point, and the identifier travels with the detection from there. This function's own contribution is the behavioral finding and its resolved entity, not the technique mapping.

FRAMEWORK OR STANDARD	ITS ROLE IN UEBA
MITRE ATT&CK (Enterprise, Cloud and Identity)	Attached by Detection Factory where a behavioral detection becomes mapped detection content. Neither computed nor validated by this function.
NIST Cybersecurity Framework 2.0	Outcome language used where platform reporting discusses anomalous-activity detection to an executive or board audience.
ISO/IEC 27001 and 27002	Control mapping and audit evidence, where behavioral monitoring of user activity is the control being evidenced.

Behavioral monitoring carries obligations of its own. ***A function that builds a personal profile of each user's activity is processing personal data, and in several jurisdictions that engages works-council consultation, employee notification or data protection impact assessment duties before deployment. The platform produces the evidence those duties call for. Whether a given deployment satisfies them is a legal determination the organization and its counsel make.***

Evidence, not compliance. The platform states which obligations behavioral monitoring produces evidence for. It never states that running the function makes an organization compliant with anything, and control-level framework mapping is held by the Regulatory Frameworks core function.

SECTION 6

Questions Raised in Evaluation

The questions below recur in buyer evaluations, each answered directly against the mechanism described in section 3.

QUESTION, AS A BUYER ASKS IT	ANSWER
How long before UEBA is useful?	The reputation-matching and correlation detections produce output immediately. The learned-baseline detections require a training period first, and this brief states that as a limit rather than working around it.
Do the reputation-matching detections need a baseline too?	No. They skip Baseline and enter the mechanism at Evaluate, because a known-malicious value is malicious whoever contacts it. They still pass through Resolve, so they name a person like every other detection.
Will first-seen detections bury the team in noise?	First-seen rates are highest during and shortly after training and settle as baselines mature. First-seen rate stability, defined in section 3.3, is tracked precisely so this can be observed rather than assumed, and the Risk Scoring Formula weighs a first-seen signal alongside everything else known about the entity rather than escalating it alone.
Does UEBA decide that a behavioral detection is serious?	No. It publishes the detection with its method and resolved user. The Risk Scoring Formula scores it and Intelligent Alert Routing assigns it.
What happens when a user's role legitimately changes?	Baselines are refreshed as observed behavior changes, so a sustained legitimate change is absorbed rather than alerting indefinitely. A sudden change still raises a detection at the point it happens, which is the intended behavior.
Is this monitoring employees?	It is behavioral detection over activity the platform already collects, and it processes personal data. Deployment obligations, including consultation and notification duties where they apply, are a legal matter for the organization, and the platform produces the evidence those duties call for.
Does UEBA replace signature-based detection?	No. It covers what signature matching structurally cannot, and four of the eleven detections are themselves reputation matches. The two approaches are complementary rather than alternatives.

SECTION 7

The Benefits

Behavioral detection earns its place by catching what nothing else in the stack is built to catch, and by removing work an analyst would otherwise do by hand mid-incident. The outcomes are countable: how much of the population has a mature baseline, how many detections arrive already attached to a person, and what share of detections come from learning rather than from matching a list.

7.1 At a glance

BENEFIT	WHAT PRODUCES IT
Tooling nobody has reported anywhere is still caught	Baseline learns what each user normally runs and contacts, so novelty is detectable without a signature.
The same action reads differently for two people	A profile is personal, so no global threshold has to be set loose for one population and tight for another.
Every detection names a person	Resolve attaches the responsible user at the point of detection, not as manual enrichment during an incident.
A reputation check follows the value, not the log	Evaluate inspects any ingested log carrying the relevant field, rather than one product's own data.
Coverage begins on day one and deepens after training	Reputation matching and correlation need no training period; learned baselines add depth as profiles mature.
Whether behavioral coverage is working is a number	Baseline maturity and detection mix by method, defined in section 3.3, make the answer reportable.

7.2 What changes, and for whom

Per-user behavioral detection changes what each role can see, and what each role no longer has to reconstruct by hand.

ROLE	WHAT CHANGES
SOC analyst	An endpoint alert arrives already attached to the person responsible, so investigation starts from the user rather than from working out who they were.
SOC manager	Baseline maturity and detection mix become visible numbers, so the question of whether behavioral coverage is actually working has an answer.
CISO and risk	Coverage extends to previously unseen tooling and to insider and compromised-account activity that signature matching is structurally poor at catching.
Service provider practice lead	The same eleven detections run across every tenant with no per-customer analytic build, while each tenant's baselines stay entirely separate.

7.3 What sets it apart

▸ **Normal is personal, not global.** A baseline describes one user, so the same action reads differently for an administrator and for a finance user without a rule being written for either.

▸ **Catches what has no signature.** A learned baseline flags a program or destination never seen for that person, whether or not it has ever been reported anywhere.

▸ **Checks travel with the value, not the log.** A reputation match inspects an IP address, domain or web address wherever it appears across ingested logs, rather than inside one product's own data.

▸ **Every detection names a person.** Identity resolution happens at the point of detection rather than as manual enrichment during an incident.

The value in one line. Signature matching answers whether the platform has seen this before. A personal baseline answers whether this person has, which is the question that catches the tool written yesterday and the credential stolen this morning.

SECTION 8

Summary

UEBA is how the platform detects what no signature describes. It is one core function among twelve, scoped to a five-stage mechanism that runs continuously inside the TDIR core rather than as a separate analytics product an analyst queries.

▸ Baseline, resolve, evaluate, detect and publish replace detection that can only recognize what it has already been told to look for.

▸ A profile of normal is built per user and per entity, so the same action reads differently for two different people without a rule being written for either.

▸ Eleven detections across four categories run on three analytic methods, and each detection states which method produced it.

▸ Every detection is attached to the responsible user at the point it is raised, rather than enriched by hand during an incident.

▸ The function detects and publishes, and leaves scoring, assignment, response and technique mapping to the core functions built for those decisions.

Forward-looking statement: any statement about future capability is forward-looking and non-binding, and capability described in the present tense is capability available at the publication date, subject to the deployment configuration in force. This document describes detection capability and does not constitute legal advice on any organization's obligations in respect of employee monitoring or personal data.

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.

ClearSkies™

CORE FUNCTION BRIEF

Detection Factory

A core function of the ClearSkies iISOC platform

Detection content as a governed, measured and continuously improving lifecycle.

Contents

SECTION 1

Executive Summary

IN ONE SENTENCE.

The Detection Factory is the core function that governs the whole life of detection content, authoring, validating, deploying, measuring, tuning and retiring the logic that decides what the platform recognizes as suspicious, so coverage is a maintained and provable asset rather than a rule library that quietly decays.

Detection engineering decides, in advance, what a security operation is able to recognize. In most operations that work ends up as a rule library: logic written by an engineer, added to a console, and left alone until an incident, an audit or a noisy alert forces someone back to it. The library grows by accumulation. Nobody owns its decay, because nobody measures it.

The Detection Factory replaces that model with a production process. Threat intelligence, validated hunt findings and coverage gaps enter at one end. Generative and agentic AI draft candidate logic, tagged with its MITRE ATT&CK technique identifier at the moment of creation. Every candidate is tested against the tenant's own historical telemetry and against known-benign traffic before it goes live. Once live, every detection is measured on precision, fidelity and duplicate rate, and a rule that falls below the governed threshold is tuned or formally retired rather than left running on reputation.

Two consequences follow. Coverage becomes something a buying committee can query rather than take on faith, because every rule carries its technique identifier and its measured quality. And the return on one piece of detection engineering compounds, because a validated detection deploys once against the shared entity model and applies to every tenant sharing the relevant exposure profile.

The function is precisely scoped. The Detection Factory decides what is detected. It does not decide how urgent a firing detection is, which is the Risk Scoring Formula; it does not decide who handles it, which is Intelligent Alert Routing; and it does not decide what happens next, which is SOAR and Playbooks. Each of those is a separate core function with its own brief.

Six-stage lifecycle

Source, author, validate, deploy, measure, tune or retire

Coverage that is provable

A queryable ATT&CK matrix per tenant, not a claim

Compounding return

One authoring event serves every relevant tenant

SECTION 2

Why Detection Content Decays

Detection engineering is underinvested compared with the rest of the security operation, and the consequences compound quietly rather than surfacing in an incident review. Four failure modes explain why, and each is answered by a specific stage of the lifecycle in section 3.

2.1 Detection debt

A rule is written once, by a particular engineer, for a particular threat, and then left alone. The environment changes around it, adversaries shift technique variants, and the person who understood the original intent moves on before anyone revisits it. The rule keeps firing, or keeps not firing, and neither outcome is examined.

2.2 Coverage that cannot be proven

Most operations cannot say with confidence which ATT&CK techniques they detect well, which they detect poorly and which they miss entirely. Rule intent lives in prose, in ticket history and in memory rather than in a structure that supports a query. Asked what the organization actually detects, the answer is assembled from recall rather than from evidence.

2.3 Tuning at the pace of headcount

Every noisy or duplicate detection found in triage needs an engineer to open the rule, reconstruct its intent, edit it, test it and redeploy it. That work competes for the same scarce hours as new coverage, and it loses the argument against the next incident. The backlog grows, and the noise it produces is absorbed by analysts instead of being fixed at the source.

2.4 A one-way pipeline from intelligence to content

Threat intelligence and threat hunting produce findings, but a finding has to be turned into a deployable rule by hand, usually in a different tool, on a different schedule, by whichever engineer is free. Days or weeks pass between proving that a technique is present and catching the next occurrence automatically. The organization stays exposed to something it has already confirmed.

2.5 The cost of the workaround

The usual response is more people rather than a different process: engineers hired to keep pace with decay, coverage requests and the tuning backlog. Cost scales with the rule base and with the rate of change in the threat landscape, while the underlying problem, content nobody measures, is untouched.

SECTION 3

How the Detection Factory Works

The Detection Factory is one continuously operating lifecycle native to the TDIR engine, not a set of separate tools. Each stage feeds the next, and the last stage feeds the first.

> *Figure 1. The six-stage lifecycle. Measurement is what closes the loop, because a detection nobody measures cannot be improved or retired on evidence.*

3.1 The six stages

STAGE	WHAT HAPPENS	FAILURE MODE IT ANSWERS
Source	Threat intelligence indicators, validated threat hunting findings and coverage gaps surfaced from normalized telemetry enter as the raw material for new or revised content.	The one-way pipeline
Author	Generative and agentic AI draft candidate logic from that material, each candidate tagged with its ATT&CK technique identifier as native metadata at the moment of creation.	Coverage that cannot be proven
Validate	Candidate logic is tested against historical normalized telemetry from every layer the platform ingests, and against known-benign traffic, so both detection and false-positive behavior are known before promotion.	Detection debt
Deploy	Validated logic is promoted into production across every tenant sharing the relevant exposure profile at once, because every tenant shares the same normalized schema.	The cost of the workaround
Measure	Every live detection carries native quality metrics, tracked per rule rather than for the rule base as a whole.	Coverage that cannot be proven
Tune or retire	A detection below the governed quality threshold is flagged automatically, then tuned and revalidated, or formally retired with its audit trail intact.	Tuning at the pace of headcount

3.2 How a detection is judged

Quality is measured rather than assumed, on three metrics that a provider can report and a client can audit.

METRIC	WHAT IT MEASURES	WHAT A POOR SCORE TRIGGERS
Precision	The share of firings that turn out to be true positives.	Tuning of the logic or its scope, then revalidation before redeployment.
Fidelity	The share of firings that arrive with correct urgency and usable context.	A metadata or enrichment correction, so downstream scoring and routing behave.
Duplicate rate	The share of firings already covered by another rule.	Consolidation, and retirement of the weaker of the two rules.

Retirement is an outcome, not a failure. ***A rule that is closed formally, with its provenance and change history preserved, leaves the coverage matrix accurate. A rule quietly disabled in a console leaves the matrix wrong, which is worse than the noise it made.***

SECTION 4

The Detection Factory in the Platform

4.1 What it draws on, what it writes back, and what it does not do

What it draws on. Normalized telemetry from the iCollector and the entity resolution the Centric-AI Fabric performs, so a candidate rule is written against the same objects every other layer resolves to. Threat intelligence indicators, validated threat hunting findings, and the coverage gaps the platform surfaces from telemetry no existing rule addresses.

What it writes back. Validated, ATT&CK-tagged detection content, and the per-rule quality metrics that come with it, published into the shared model. Risk scoring, alert routing, SOAR and playbooks consume that output. The relationship is one way: nothing is returned to the Detection Factory that changes what it authors, and the quality metrics come from a detection's own firing history rather than from a signal engineered by a downstream function.

What it does not do. It does not decide urgency, ownership or response, and it holds no telemetry of its own. A native add-on contributes cross-layer visibility by adding a signal no other layer can see. The Detection Factory contributes something different in kind: it decides what any of that telemetry is capable of revealing.

4.2 A worked example

The chain below is illustrative and technically representative rather than drawn from a named engagement.

WHAT A CONVENTIONAL OPERATION DOES	WHAT THE DETECTION FACTORY DOES
A hunter confirms a technique is present. The finding is written up and queued for an engineer.	The finding enters the lifecycle at Source, with no separate queue and no hand-off.
An engineer drafts a rule weeks later, scoped to the exact indicator, and tests it against whatever sample data is at hand.	Logic is drafted against the underlying technique rather than the single indicator, tagged to its ATT&CK identifier, and validated against ninety days of normalized cross-layer telemetry.
The rule is deployed to the tenant where the finding originated. Other tenants stay exposed until a similar finding turns up there.	Deploy promotes the validated rule across every tenant sharing that exposure profile in a single event.
The resulting alert reaches triage unscored and unrouted, waiting on a person to establish urgency and owner.	The alert reaches risk scoring and alert routing already ATT&CK-tagged and carrying the rule's precision history.
A noisy or duplicate rule is found months later in a rule-base review, if a review happens at all.	Measure flags the rule on its own metrics within weeks, and Tune or retire acts on it as routine work.

4.3 Delivery across many tenants

A detection validated in one tenant is generalized to the technique and deployed to every tenant whose exposure profile it covers, so detection engineering capacity stops tracking headcount and starts tracking the size of the client base. Tenant separation is unaffected: what travels between tenants is the

logic and its metadata, never the telemetry that produced it, and no tenant sees, holds or infers the data of another. A provider can therefore answer the coverage question per client, from the same matrix, without a manual audit per client.

SECTION 5

ATT&CK Alignment and Supported Frameworks

MITRE ATT&CK is the detection spine, and the Detection Factory is the function that owns the distinction the platform draws throughout its documentation: a technique listed in a coverage matrix is claimed, and a technique exercised in a purple-team run or against MITRE Engenuity ATT&CK Evaluations is demonstrated. Both are reported, and they are never presented as the same thing.

Detection content is mapped across the Enterprise, Cloud and Identity matrices. A technique is cited by identifier and name on first use, for example T1566.002 Spearphishing Link, and by identifier alone thereafter. [CLEARSKIES INPUT: confirm the matrix version in force and the refresh cadence]

LIFECYCLE STAGE	HOW ATT&CK IS USED
Author	Every candidate carries its technique identifier, or identifiers, as native metadata rather than as documentation added afterwards.
Validate	The mapping is checked against the matrix version in force before a candidate is promoted.
Measure	Precision and fidelity are reported per technique as well as per rule, so a weak area of the matrix is visible rather than averaged away.
Tune or retire	A change of rule state updates the coverage matrix immediately, so the matrix describes what is live rather than what was once written.

FRAMEWORK OR STANDARD	ROLE IN THE DETECTION FACTORY
MITRE ATT&CK, Enterprise, Cloud and Identity	Technique mapping for every detection, and the basis of the per-tenant coverage matrix.
MITRE Engenuity ATT&CK Evaluations	Independent validation of detection behavior against an emulated adversary, and the basis of the demonstrated claim.
MITRE D3FEND	Countermeasure vocabulary, cross-referenced where a detection maps to a defensive technique.

FRAMEWORK OR STANDARD	ROLE IN THE DETECTION FACTORY
Sigma	Portable detection format for exchange with other tooling, in both directions.
STIX and TAXII	Structured intelligence intake at the Source stage.

Alignment to OCSF and OpenTelemetry, and the syslog, CEF, LEEF and REST API ingestion formats, are properties of the collection layer and are described in the platform brief rather than restated here.

SECTION 6

Business Impact and Delivery

The Detection Factory reports the measures below natively and per tenant. Definitions are published because the same metric name is measured differently by different vendors. Target values are agreed per engagement rather than asserted here.

MEASURE	DEFINITION USED	REPORTED AS
Technique coverage	Techniques with at least one live, validated detection, as a share of the techniques in scope for the tenant.	A matrix per tenant, queryable at any time.
Precision per rule	True positives as a share of all firings for that rule.	Trended per rule and per technique.
Duplicate rate	Firings already covered by another live rule, as a share of firings.	Per rule, with the consolidation candidates named.
Time from source to deployment	Elapsed time from a finding or indicator entering Source to the validated detection being live.	Trended, and reported per tenant.
Retirement rate	Rules formally retired in the period, as a share of the live rule base.	Trended, with the reason recorded.

6.1 What changes, and for whom

ROLE	WHAT CHANGES
Detection engineer	Authoring and routine tuning move into the lifecycle, so the engineer's time goes to custom requirements and to reviewing what the lifecycle proposes.

ROLE	WHAT CHANGES
SOC analyst	Fewer duplicate and low-precision alerts reach triage, because the rules that produce them are flagged and fixed at the source.
SOC manager	Coverage and detection quality become reportable numbers rather than an assessment gathered by asking the team.
CISO and risk	The coverage question has an evidenced answer, per technique, with claimed and demonstrated distinguished.
MSSP practice lead	Detection engineering capacity scales with the client base rather than with the number of engineers on staff, and per-tenant coverage supports contractual reporting.

6.2 What sets it apart

▸ **Authored, not imported.** Logic is drafted and validated against the tenant's own normalized telemetry rather than adapted from a generic content pack written for an environment nobody involved has seen.

▸ **ATT&CK mapping as native metadata.** The technique identifier is attached when the rule is created, which is what makes the coverage matrix a query rather than a documentation exercise.

▸ **Coverage measured, not asserted.** Precision, fidelity and duplicate rate are tracked per rule, so a claim about detection quality can be produced from the platform rather than from a conversation.

▸ **One authoring event, many tenants.** A validated detection reaches every tenant sharing the exposure profile without separate engineering per tenant, and without any tenant data moving.

6.3 Objection handling

QUESTION, AS A BUYER ASKS IT	ANSWER
Is the Detection Factory a separate product?	No. It is one of the twelve core functions of the TDIR engine, scoped to the lifecycle of detection content. There is no separate license, no separate console and no separate deployment.
Does it replace a detection-engineering team?	It changes what that team spends time on. Authoring, validation and routine tuning run in the lifecycle; the organization's engineers remain the authority for custom requirements and for reviewing what the lifecycle proposes.
How is a claimed detection different from a demonstrated one?	A technique listed in the coverage matrix is claimed. A technique exercised in a purple-team run or validated against MITRE Engenuity ATT&CK Evaluations is demonstrated. Reporting states which of the two applies.

QUESTION, AS A BUYER ASKS IT	ANSWER
What happens to a detection that stops performing?	Measure tracks precision, fidelity and duplicate rate per rule. A rule below the governed threshold is flagged for tuning or formal retirement rather than left live on the strength of having once been useful.
Does a detection written for one tenant help other tenants?	Where tenants share the relevant exposure profile, yes. The logic and its metadata deploy across them in one event. No tenant telemetry moves, and no tenant sees another tenant's data.
Does it receive anything back from risk scoring, routing, SOAR or playbooks?	No. The relationship is one way. The Detection Factory publishes content and quality metrics; those functions consume them. Quality measurement comes from a detection's own firing history.

SECTION 7

Summary

The Detection Factory is how the platform produces and maintains detection content. It is one core function among twelve, scoped to the lifecycle that decides what the platform is capable of recognizing, and it runs continuously inside the TDIR engine rather than as a bundle of separately branded capabilities.

▸ Content is authored and validated against the tenant's own normalized telemetry, not imported from a generic file.

▸ Every rule carries its ATT&CK technique identifier as native metadata from the moment it is created.

▸ Coverage and quality are measured per rule and per technique, so both can be queried rather than asserted.

▸ A validated detection deploys once and applies to every tenant sharing the relevant exposure profile.

▸ The function decides what is detected, and leaves urgency, ownership and response to the core functions built for those decisions.

ClearSkies™

CORE FUNCTION BRIEF

Risk Scoring Formula

A core function of the ClearSkies iISOC platform

One explainable value, correlated across seven dimensions, continuously recalculated.

Contents

SECTION 1

Executive Summary

IN ONE SENTENCE.

The Risk Scoring Formula is the core function that converts every alert into a single, explainable value by continuously correlating seven risk dimensions, so urgency reflects current, correlated risk rather than a severity label fixed at the moment of detection.

A conventional security operation asks one static question of every alert: is this high or low severity? That label is fixed at detection time, blind to context, and identical whether the affected asset is a test server or a domain controller. It does not move as an investigation reveals more, and it means something different in every tool that assigns it.

The Risk Scoring Formula replaces the label with a process. Behavioral, technique, asset, identity, intelligence, temporal and environmental signals enter as seven correlated dimensions. Each carries a dynamic weight, tenant-specific rather than fixed, and the platform recalculates the resulting value the moment any dimension changes. The value falls into one of four bands, Low, Medium, High and Critical, and each band changes what becomes possible next without the formula itself performing the next step.

Two consequences follow. An alert from an identity provider and an alert from an endpoint agent land on the same scale, because both are scored against the same seven dimensions and the same shared entity model. And a decision made a minute ago is revised the moment new evidence arrives, so a slow-burning pattern that stays below any single tool's threshold still climbs the scale as its signals accumulate.

The formula is native to the TDIR engine rather than a module added to it. Every alert the platform ingests, from every layer and every native add-on, passes through the same seven-dimension mechanism before anything downstream acts on it, so the score in a monthly report and the score on a single alert come from one calculation rather than from two tools reconciled by hand.

THE FUNCTION IS PRECISELY SCOPED.

The Risk Scoring Formula decides how urgent an alert is. It does not decide who handles it, which is Intelligent Alert Routing, and it does not decide or execute what happens next, which is SOAR and Playbooks. Each of those is a separate core function with its own brief.

7 dimensions
Correlated, dynamically weighted

1 explainable value
Continuously recalculated

One scale
Consistent across every layer

SECTION 2

Why Static Severity Fails

Prioritization is underinvested relative to the rest of the security operation, and the consequences compound quietly rather than surfacing in an incident review. Four failure modes explain why, and each is answered by a dimension of the mechanism described in section 3.

2.1 A label fixed at the moment of detection

Severity is set once, when a rule fires, and is never revisited. It is identical whether the affected asset is a test server or a domain controller, and it does not change as an investigation reveals more or less about the underlying activity. The measurement that would reveal the drift, whether the original label still fits, belongs to nobody.

2.2 Severities that do not mean the same thing twice

A high severity in one tool reflects that tool's own internal scale, calibrated to its own detection logic. Across a SIEM, an identity provider and an endpoint agent, three unrelated high severities are not comparable, so an analyst prioritizing across consoles is comparing labels rather than risk.

2.3 Weak signals that never accumulate

A single low-risk event is flagged or ignored at the moment it occurs, and nothing in a static model tracks whether the same weak signal is recurring. A low-and-slow pattern that stays below any one alert's threshold can persist for weeks with no rising score to surface it. The analyst who absorbs the resulting noise has no way to see the pattern building underneath it.

2.4 Re-triage as the only way risk updates

When new evidence changes the picture, a fresh threat-intelligence match, a privilege escalation, or an asset reclassified as critical, a static severity does not move on its own. An analyst has to notice the change and manually re-triage, so most alerts are prioritized on the information available the moment they fired rather than on the fuller picture available minutes later. The organization stays exposed to a risk it has, in effect, already confirmed.

2.5 The cost of the workaround

The usual response is more headcount rather than a different mechanism: analysts added to hold several consoles open and reconcile unrelated severity scales by memory. Cost scales with alert volume and with the number of tools in the stack, while the underlying problem, a label that cannot update itself, is untouched

The workaround also fails quietly rather than visibly. A team that adds an analyst to cover the reconciliation work does not see the missed slow-burning pattern in a staffing report. It sees only the incident review after the fact, by which point the cost has been paid in dwell time rather than in headcount.

SECTION 3

How the Risk Scoring Formula Works

The Risk Scoring Formula is one continuously operating mechanism native to the TDIR engine, not a rule evaluated once. Seven dimensions feed one score, the score falls into one of four bands, and the whole calculation runs again whenever new evidence arrives. Figure 1 shows the model and the re-scoring loop.

> *Figure 1. The seven-dimension model and the continuous re-scoring loop. Re-scoring is what keeps the value current as evidence changes.*

3.1 The seven dimensions

Each alert is evaluated across every dimension, never on one in isolation. The table names what each dimension measures, the platform source that supplies its signal, and the failure mode from section 2 that it answers.

DIMENSION	WHAT IT MEASURES, AND WHERE THE SIGNAL COMES FROM	FAILURE MODE IT ANSWERS
Behavioral confidence	Deviation from established baselines and anomalous behavior over time, such as abnormal login times, rare command execution or unusual access paths. Sourced from UEBA.	Weak signals that never accumulate
ATT&CK relevance	Alignment with known techniques and kill-chain progression, such as credential access chaining into lateral movement. Sourced from the technique identifier the Detection Factory attaches to the firing detection.	Severities that do not mean the same thing twice

DIMENSION	WHAT IT MEASURES, AND WHERE THE SIGNAL COMES FROM	FAILURE MODE IT ANSWERS
Asset criticality	Business and operational importance of the affected asset, such as domain controllers, cloud administrator accounts, operational technology or crown-jewel applications. Drawn from the shared entity model.	A label fixed at the moment of detection
Identity risk context	Privilege level, account type and behavior of the identity involved, including service accounts, dormant identities and multi-factor authentication status.	A label fixed at the moment of detection
Threat-intelligence confidence	Correlation with internal or external intelligence, including known malicious addresses and domains, active campaigns and industry-specific threats. Sourced from Threat Intelligence.	Re-triage as the only way risk updates
Temporal and pattern persistence	Whether activity is isolated or recurring, such as low-and-slow persistence or repeated failed actions over days or weeks.	Weak signals that never accumulate
Environmental context	Whether behavior occurs inside or outside expected operating conditions, such as maintenance windows, business hours or planned change events. It lowers the score during an expected anomaly as readily as it raises it.	Severities that do not mean the same thing twice

Risk increases as dimensions correlate rather than in isolation. A single low-risk dimension rarely escalates an alert on its own. Several reinforcing dimensions together drive it up the scale, which is how the formula surfaces a genuine slow-burning pattern while keeping routine noise quiet.

3.2 How urgency is judged

The score is judged against four bands, and each band changes what becomes possible downstream without the formula itself performing the next step.

BAND	WHAT CROSSES INTO THIS BAND	WHAT IT ENABLES DOWNSTREAM
Low	The score stays below the review threshold.	Intelligent Alert Routing keeps the alert aggregated rather than assigning it to an analyst queue.
Medium	The score crosses the review threshold.	Intelligent Alert Routing assigns the alert to an L1 analyst tier.
High	The score crosses the priority threshold.	Intelligent Alert Routing assigns with priority to an L2 or L3 analyst, and SOAR flags the case for expedited handling.

BAND	WHAT CROSSES INTO THIS BAND	WHAT IT ENABLES DOWNSTREAM
Critical	The score crosses the containment threshold.	The governance policy in SOAR evaluates the case for autonomous containment. Where authorized, Playbooks execute while Intelligent Alert Routing maintains oversight.

A low band is not zero risk. ***A Low band means the current evidence does not yet justify analyst attention, not that the event is dismissed. The event remains part of the correlated picture, and a recurring or escalating pattern raises the score as further signals arrive, which is the mechanism section 2.3 describes as missing from a static model.***

3.3 Continuous re-scoring

Risk is never static, so the score is recalculated whenever the picture changes. The formula re-scores when new telemetry arrives, when additional ATT&CK techniques are detected, when identity privileges change, when asset importance is updated, and when threat intelligence is refreshed. Three things follow from that.

▸ **Slow attacks escalate early.** Low-and-slow activity that would evade a static severity climbs the scale as its signals accumulate.

▸ **False positives de-escalate.** Alerts that prove benign fall back down the scale without consuming analyst time.

▸ **Urgency adapts without manual re-triage.** The band follows the risk automatically as conditions change, rather than waiting for someone to notice.

SECTION 4

The Risk Scoring Formula in the Platform

4.1 What it draws on, what it writes back, and what it does not do

What it draws on. The technique identifier and precision history the Detection Factory attaches to a firing detection; behavioral baselines and anomaly measurements from UEBA; indicator confidence and campaign relevance from Threat Intelligence; and the asset, identity and environmental context the shared entity model carries. Each is named as an input to one of the seven dimensions rather than as a general capability the formula generates itself.

What it writes back. A score and an urgency band, published into the shared model. Intelligent Alert Routing, SOAR and Playbooks consume that output. The relationship is one way: nothing is returned to the Risk Scoring Formula that changes how the current score was calculated, and a later re-score is

triggered by new evidence arriving at one of the seven dimensions rather than by a signal a downstream function sends for the purpose.

What it does not do. It does not choose an analyst, a queue or a response action, and it holds no telemetry of its own. Its contribution is not a new signal but a single, comparable answer to how urgent an existing signal is.

Beyond the bands themselves, the same value feeds analyst assignment by skill level and workload, escalation that respects the service levels in force, and immediate handling for high-confidence threats. In every case the formula supplies the value and a sibling core function performs the action.

4.2 A worked example

> **FROM A QUIET FIRST SIGNAL TO A CRITICAL SCORE.**
>
> A service account authenticates from an unfamiliar location outside business hours. On its own the event scores modestly: identity risk context and environmental context contribute, while ATT&CK relevance and behavioral confidence stay low because no technique has matched and the account's baseline tolerates occasional off-hours activity. Minutes later, Detection Factory content flags a technique match for credential use consistent with a known campaign, and UEBA reports the same account performing a rare directory-enumeration command. Behavioral confidence and ATT&CK relevance both rise, and because the signals reinforce rather than stand alone, the score crosses from Medium to High. Threat Intelligence then confirms the source network range is associated with active campaign infrastructure, and the score crosses into Critical. The single off-hours login, considered alone twenty minutes earlier, would not have justified any of this.

4.3 Delivery across many tenants

Weighting is configured per tenant, industry and asset group, so the same seven-dimension mechanism adapts to each client's environment without a bespoke rebuild. What travels between tenants is the weighting configuration and the scoring logic. What never travels is the telemetry that produced any tenant's score. A provider can therefore explain the reasoning behind an urgency decision to any client, from the same mechanism, without a manual audit or a separate scoring engine per tenant.

Every change to a tenant's weighting configuration is logged with its date, its author and the reason recorded against it, so a shift in how an environment is scored is traceable rather than silent. A weighting change is reviewed before it takes effect, and nothing adjusts the dynamic weights in production without passing through that review.

SECTION 5

ATT&CK Alignment and Supported Frameworks

ATT&CK relevance is one of seven correlated dimensions, not the sole basis for a score. The technique identifier a given alert carries is supplied by the Detection Factory at the point of detection. The Risk Scoring Formula consumes that identifier, together with kill-chain progression across chained techniques, as one input among seven. A technique is cited by identifier and name on first use, for example T1078 Valid Accounts, and by identifier alone thereafter.

Technique relevance is scored across the Enterprise, Cloud and Identity matrices, matching the matrices the Detection Factory maps detection content against. Where an alert chains multiple techniques, later positions in the chain, closer to impact than to initial access, weigh more heavily on this dimension, which is what allows a discovery step and a collection step to score differently even when both map to a technique in the same matrix

Mapping is not coverage. A technique listed in the coverage matrix is claimed. A technique exercised in a purple-team run or against MITRE Engenuity ATT&CK Evaluations is demonstrated. The distinction belongs to the Detection Factory, which owns the matrix. The Risk Scoring Formula consumes the identifier rather than validating it.

FRAMEWORK OR STANDARD	ROLE IN THE RISK SCORING FORMULA
MITRE ATT&CK, Enterprise, Cloud and Identity	Supplies the technique identifier and kill-chain position that inform the ATT&CK relevance dimension
NIST Cybersecurity Framework 2.0	Outcome language used where platform reporting discusses aggregate risk posture to an executive or board audience.

Alignment to OCSF and OpenTelemetry, and the syslog, CEF, LEEF and REST API ingestion formats, are properties of the Collection Layer and are described in the ***ClearSkies iISOC platform brief*** rather than restated here.

SECTION 6

Business Impact and Delivery

The Risk Scoring Formula reports the measures below natively and per tenant. Definitions are published because the same metric name is measured differently by different vendors. Target values are agreed per engagement rather than asserted here.

MEASURE	DEFINITION USED	REPORTED AS
Band accuracy	Critical-band and High-band alerts confirmed as true positives on investigation, as a share of all alerts in that band.	Trended per tenant.
Re-scoring latency	Elapsed time from new evidence arriving at any dimension to the score reflecting it.	Trended, with outliers flagged.
De-escalation rate	Alerts that fall back to the Low band as further evidence proves them benign, as a share of all scored alerts.	Reported as analyst time protected from unnecessary triage.
Escalation lead time	For a confirmed slow-burning pattern, the interval between the first weak signal and the score crossing into High or Critical.	Reported per confirmed case.
Dimension coverage	Scored alerts with all seven dimensions populated with a non-default value, as a share of all scored alerts.	Per tenant, with the weakest dimension named.

6.1 What changes, and for whom

ROLE	WHAT CHANGES
SOC analyst	Alerts arrive with a current, correlated urgency rather than a fixed label, so triage time goes to genuine risk rather than to reconciling severities across consoles.
SOC manager	Prioritization quality becomes a reportable, explainable number rather than a judgment call that varies by analyst and by shift.
CISO and risk	Every urgency decision carries a defensible rationale, current at the moment it is read rather than at the moment the alert first fired.
MSSP practice lead	One scoring mechanism applies the same explainable logic across every tenant, and weighting adapts per client without a bespoke engine.

For the organization defending its own operations, effort concentrates where business impact is greatest, because the score reflects asset criticality and identity context rather than the opinion encoded in a rule. For the service provider, the same mechanism removes manual triage as the bottleneck and lets a lean team deliver predictable outcomes across many tenants.

6.2 What sets it apart

▸ **Seven correlated dimensions, one explainable value.** No single weak signal escalates an alert; reinforcing signals across dimensions do.

▸ **Dynamic rather than fixed weighting.** Weight ranges adjust per industry, tenant and asset group, and recalculate as conditions change.

▸ **Continuous re-scoring.** A score changes the moment new telemetry, intelligence, identity or asset context arrives, rather than waiting for manual re-triage.

▸ **One scale across every layer.** An identity event and an endpoint event are ranked on the same scale because both are scored against the same seven dimensions.

6.3 Objection handling

QUESTION, AS A BUYER ASKS IT	ANSWER
Does the Risk Scoring Formula decide which analyst handles an alert?	No. It computes the score and the urgency band. Intelligent Alert Routing reads that output to decide the queue, the analyst tier and the assignment by skill and workload.
Does the score itself trigger autonomous containment?	No. A Critical band makes autonomous containment eligible. The governance policy in SOAR decides whether it is authorized for that class of case, and Playbooks execute the action once authorized.
How is the score different from a single-vendor severity rating?	A single-vendor severity is fixed at the point a rule is defined and reflects one dimension. The Risk Scoring Formula correlates seven dimensions, weights them dynamically per tenant and asset group, and recalculates continuously as new evidence arrives.
Can the weighting be tuned for a specific industry or client?	Yes. Weight ranges are adjustable per industry, tenant and asset group and are recalculated as conditions change, so the same mechanism adapts without a bespoke rebuild. Every change is logged and reviewed before it takes effect.
What happens when new intelligence arrives after an alert has already been scored?	The score recalculates immediately. A refreshed threat-intelligence match, a privilege change or an asset reclassification all trigger re-scoring without waiting for manual re-triage.

QUESTION, AS A BUYER ASKS IT	ANSWER
Does it receive anything back from alert routing, SOAR or playbooks?	No. The relationship is one way. The Risk Scoring Formula publishes the score and the band, and those functions consume them. A later re-score is triggered by new evidence, never by a return signal.

SECTION 7

Summary

The Risk Scoring Formula is how the platform decides how urgent an alert is. It is one core function among twelve, scoped to computing and continuously recalculating a single, explainable value by correlating seven risk dimensions, and it runs inside the TDIR engine rather than as a judgment made once and left to age.

▸ Seven correlated dimensions, sourced from the Detection Factory, UEBA, Threat Intelligence and the shared entity model, replace a severity label fixed at detection time.

▸ Weighting is dynamic and tenant-configurable, and every change to it is logged and reviewed before it takes effect.

▸ The score recalculates continuously as new telemetry, intelligence, identity or asset context arrives.

▸ Every alert, whatever layer it came from, is expressed on the same scale and in the same four bands.

▸ The function decides urgency, and leaves ownership and response to the core functions built for those decisions.

External research is cited inline where it is used, and this brief carries no separate source register. Forward-looking statement: any statement about future capability is forward-looking and non-binding, and capability described in the present tense is capability available at the publication date, subject to the deployment configuration in force.

ClearSkies™

CORE FUNCTION BRIEF

Intelligent Alert Routing

A core function of the ClearSkies iISOC platform

The right alert, to the right analyst, without a human triaging the queue by hand.

Contents

SECTION 1

Executive Summary

IN ONE SENTENCE.

Intelligent Alert Routing is the core function that turns a scored alert into a governed assignment, matching analyst tier to alert difficulty and balancing workload in real time, so who works an alert is decided by policy rather than by whoever happens to be triaging the queue.

In a conventional security operation a person decides who works each alert. That single dependency creates a predictable pattern: experienced analysts spend their day sorting rather than investigating, high-risk items sit in a shared queue behind noise, and when volume spikes there is no elastic capacity to absorb it. Routing by hand scales with neither alert volume nor analyst headcount, and the same alert can land differently depending on who is on shift.

Intelligent Alert Routing replaces the person with a policy. A scored alert is mapped to the analyst tier its difficulty requires, matched against the least-loaded eligible analyst at that tier, and assigned under rules that resolve ties the same way every time. When every eligible analyst at a tier saturates against its service-level threshold, eligible lower to medium complexity alerts route to the platform's autonomous handling capacity instead of waiting, and route back to human analysts as capacity recovers.

Every decision the function makes is visible and reconstructable: which tier an alert was mapped to, which analyst was eligible, and why the assignment landed where it did. Nothing about the decision depends on memory or on which lead happened to be on shift when the alert arrived.

THE FUNCTION IS PRECISELY SCOPED.

Intelligent Alert Routing decides who works an alert. It does not decide how urgent the alert is, which is the Risk Scoring Formula. It does not investigate the alert, which is Generative and Agentic AI. And it does not decide or execute a response, which is SOAR and Playbooks. Each of those is a separate core function with its own brief.

Five stages
Tier, workload, assignment, overflow, state

Three tiers
Matched to alert difficulty

Elastic
Surge absorbed without added headcount

SECTION 2

Why Manual Triage Falls Short

Assignment is underinvested relative to the rest of the security operation, and the consequences compound quietly rather than surfacing in an incident review. Four failure modes explain why, and each is answered by a stage of the mechanism in section 3.

2.1 Skilled analysts consumed by sorting

A senior analyst who triages the queue by hand spends the first minutes of every shift deciding what to work rather than working it. That time is not recovered, and it scales with alert volume rather than with the value the analyst produces once actually investigating.

2.2 High-risk alerts wait behind noise

A shared queue does not differentiate by difficulty or urgency once an alert lands in it. A critical alert filed a minute after a routine one can wait behind it if nobody has yet looked at either, and the delay is invisible until someone happens to notice.

2.3 No headroom under surge

When volume spikes, a fixed analyst headcount has nowhere to send the overflow. Every alert still needs a human decision about who works it, so the same triage bottleneck that slows a quiet day becomes a missed commitment on a busy one.

2.4 Assignment that varies by shift

Two identical alerts, triaged by two different leads, can reach different tiers or different analysts, because the decision rests on individual judgment rather than on a consistent policy. Reviewing why a specific alert reached a specific analyst becomes a question about who was on duty rather than a question the system can answer on its own.

2.5 The cost of the workaround

The usual response is a dedicated triage lead or a rotating duty roster, a role whose entire function is to keep the queue moving rather than to investigate anything in it. The cost scales with alert volume and with the number of shifts covered, while the underlying problem, an assignment decision that depends on a person rather than a policy, is untouched. The workaround also caps how far the operation can grow: adding tenants or alert sources means adding triage capacity in lockstep, so growth is bounded by how fast people can be hired and trained to sort rather than to investigate.

Industry research on security operations consistently ties manual triage overhead to lost investigation time and to analyst attrition, though the specific figures are not yet sourced for this brief.

SECTION 3

How Intelligent Alert Routing Works

Intelligent Alert Routing is one continuously operating mechanism native to the TDIR core, not a queue watched by a person. A scored alert enters at one end and an auditable assignment leaves at the other, with the load picture updated in real time throughout.

> *Figure 1. The five-stage assignment mechanism.*

3.1 The five stages

Each stage is named once here and referenced by name throughout the rest of this brief, paired with the failure mode from section 2 that it answers.

STAGE	WHAT HAPPENS	THE FAILURE MODE IT ANSWERS
Tier mapping	The score band an alert carries from the Risk Scoring Formula is resolved against the tenant's analyst roster and skill levels, so low-score alerts map to L1, mid-score to L2, and top-score or critical alerts to L3.	Assignment that varies by shift
Workload awareness	The pending alert count for every analyst eligible at the required tier is tracked continuously, rather than assumed from a snapshot.	Skilled analysts consumed by sorting
Assignment decision	The eligible analyst with the fewest pending alerts receives the assignment. Where loads are equal, a defined tie-break rule, round robin or longest idle, resolves it the same way every time.	High-risk alerts wait behind noise
SLA overflow routing	When every eligible analyst at a tier saturates against its service-level threshold, eligible lower to medium complexity alerts route to the platform's autonomous handling capacity instead of a human queue.	No headroom under surge
State tracking	Pending counts and pool composition update in real time as alerts are assigned, resolved or reassigned, so the next decision is made on the current picture rather than a stale one.	Skilled analysts consumed by sorting

The mechanism resolves one decision at a time, tier then load then tie-break, rather than weighing every factor at once. That ordering is deliberate: difficulty match is never traded away to balance load, and load is never ignored once difficulty is matched.

3.2 How an assignment is judged

Quality is measured rather than assumed, on definitions a provider can report and a customer can audit. The measures below are reported natively and per tenant. They are defined here because the same metric name is measured differently by different vendors, and target values are agreed per engagement rather than asserted.

MEASURE	THE DEFINITION USED	WHAT A POOR VALUE TRIGGERS
Time to assignment	Elapsed time from a score and band arriving to an assignment being made.	Investigation of the specific stage where the latency is introduced.
Tier match accuracy	Assignments requiring no manual override to reach the correct tier, as a share of all assignments.	A review of the score-to-tier thresholds for the affected alert type.
Load balance variance	The spread of pending-alert counts across eligible analysts within a tier.	A review of tie-break rules, or of roster capacity at that tier.
SLA attainment under surge	Alerts meeting the service-level response target during a saturation event, as a share of all alerts in that event.	A review of tier thresholds and roster size against the observed peak.
Autonomous overflow share	Alerts routed to autonomous handling during saturation, as a share of all alerts in that period.	Nothing on its own. It is read as evidence that overflow is engaging as designed.

Overflow is a designed state, not a failure state. ***A tenant whose queues never saturate is not necessarily healthier than one whose overflow routing engages occasionally. Overflow engaging as designed, absorbing eligible alerts and releasing them back to human analysts as capacity recovers, is the mechanism protecting the service commitment while it works.***

SECTION 4

Intelligent Alert Routing in the Platform

4.1 What it draws on, what it writes back, and what it does not do

What it draws on. The score and urgency band the Risk Scoring Formula publishes for each alert, the tenant's analyst roster and skill levels, and the service-level thresholds the KPIs and SLAs core function defines. Each is an input to a specific stage rather than a general capability the function generates itself.

What it writes back. The assignment decision and its audit trail, published into the shared model. The assigned analyst tier or the platform's autonomous handling capacity acts on it, and KPIs and SLAs consumes the queue and assignment timing for reporting. The relationship is one way: nothing returned to Intelligent Alert Routing changes an assignment already made, and the measures in section 3.2 come from the mechanism's own decision history rather than from a signal a downstream function engineers for the purpose.

What it does not do. It does not compute a risk score, and it does not investigate or act on an alert once assigned. Its contribution is not a new signal or a new action but a consistent, auditable answer to who should work an existing, already-scored alert.

4.2 A worked example

The comparison below is illustrative and technically representative rather than drawn from a named engagement.

WHAT MANUAL TRIAGE DOES	WHAT INTELLIGENT ALERT ROUTING RESOLVES
A shift lead scans the queue periodically and assigns alerts by impression, in the order noticed rather than in order of urgency.	Tier mapping resolves the alert's score band the moment the score arrives, before any human has looked at the queue.
The lead checks who seems free, often from memory rather than from a current count.	Workload awareness tracks the pending count for every eligible analyst continuously.

WHAT MANUAL TRIAGE DOES	WHAT INTELLIGENT ALERT ROUTING RESOLVES
Two analysts with similar loads both seem available, and the lead picks one, sometimes the one who is easier to reach.	Assignment decision selects the least-loaded eligible analyst and applies a defined tie-break rule when loads are equal.
A surge arrives. The lead has no capacity to create and starts a waiting list, while the clock keeps running.	SLA overflow routing detects saturation against the threshold and routes eligible alerts to autonomous handling immediately.
The queue picture is only as current as the lead's last scan.	State tracking updates the load picture on every assignment, resolution and reassignment, in real time.

4.3 Delivery across many tenants

Tier thresholds, roster composition and service-level targets are configured per tenant, so the same five-stage mechanism serves every tenant's policy without a bespoke rebuild. What travels between tenants is the routing logic and its configuration; what never travels is one tenant's roster, queue contents or assignment history. A provider can therefore explain any assignment to any customer, from the same mechanism, without a manual audit or a per-tenant routing engine.

Because the function is switched on with the platform rather than deployed separately, time to value is a matter of tuning thresholds to a roster rather than an onboarding project. In the first week, default tier thresholds apply against the imported roster, so assignment begins immediately even before thresholds are tuned. Through the first month, tier match accuracy and load balance variance surface any threshold that needs adjustment for that tenant's alert mix and skill distribution. At steady state, SLA attainment under surge is tracked as an ongoing measure, reviewed whenever roster size or alert volume shifts materially.

SECTION 5

ATT&CK Alignment and Supported Frameworks

Intelligent Alert Routing does not map detection content and does not compute technique relevance itself. Where a technique identifier is present, it travels with the score and band the Risk Scoring Formula supplies, and it is carried into the assignment audit trail so an analyst can see, at the moment of

assignment, the adversary behavior the routing decision was made against.

FRAMEWORK OR STANDARD	ITS ROLE IN INTELLIGENT ALERT ROUTING
MITRE ATT&CK, Enterprise, Cloud and Identity	Carried through from the Risk Scoring Formula's output into the assignment audit trail. It is not computed or validated by this function.
NIST Cybersecurity Framework 2.0	Outcome language used where platform reporting discusses assignment and service-level performance to an executive or board audience.

Mapping is inherited, not asserted. Because this function neither produces nor validates a technique mapping, it makes no coverage claim of its own. Control-level mapping to the frameworks a customer reports against is held by the Regulatory Frameworks core function.

SECTION 6

Questions Raised in Evaluation

The questions below recur in buyer evaluations, each answered directly against the mechanism described in section 3.

QUESTION, AS A BUYER ASKS IT	ANSWER
Does Intelligent Alert Routing decide how urgent an alert is?	No. It consumes the score and band the Risk Scoring Formula computes. Its own decision is which tier and which analyst, not how urgent the alert is.
Does the autonomous handling capacity investigate alerts on its own authority?	Routing decides that an eligible alert moves to autonomous handling when human capacity saturates. The investigation itself is performed by Generative and Agentic AI, under its own governance, not by this function.
What happens when every analyst at a tier is at capacity?	Eligible lower to medium complexity alerts route to autonomous handling. Higher-tier cases wait for the next available analyst under policy rather than being downgraded to a tier they do not fit.
Can routing rules be customized per tenant?	Yes. Tier thresholds, roster composition and service-level targets are configured per tenant, and the same mechanism applies the configuration consistently.

QUESTION, AS A BUYER ASKS IT	ANSWER
Does routing ever override the risk score?	No. Tier mapping follows the score as published. Routing does not re-score an alert or adjust the urgency it was given.
Can a specific analyst be favored or bypassed?	No. Assignment follows tier eligibility, then load, then a defined tie-break rule, which is what removes shift-dependent variation from the decision.

SECTION 7

The Benefits

Policy-driven assignment is where a security operation stops paying for sorting, and the outcomes are countable: how much analyst time goes to investigation rather than triage, whether the commitment holds when volume spikes, and whether any assignment can be explained after the fact.

7.1 At a glance

BENEFIT	WHAT PRODUCES IT
Analyst time goes to investigation rather than sorting	Tier mapping and assignment decision run the moment a score arrives, so no one triages the queue by hand.
A high-risk alert does not wait behind a routine one	The score band, not queue order, determines the tier an alert is mapped to.
Commitments hold when volume spikes	SLA overflow routing absorbs eligible alerts into autonomous handling at saturation, and releases them back as capacity recovers.
The same alert reaches the same class of decision on any shift	A defined tie-break rule and continuous state tracking, rather than a lead's judgment and last scan.
Assignment quality becomes a reportable number	Five measures defined in section 3.2, reported natively and per tenant.

BENEFIT	WHAT PRODUCES IT
Any assignment can be explained after the fact	An audit trail carrying the score, tier and workload in force at the moment the decision was made.

7.2 What changes, and for whom

Policy-driven assignment changes what each role spends time on, and what each role can rely on being consistent.

ROLE	WHAT CHANGES
SOC analyst	Time goes to investigation rather than to sorting, and assignment matches difficulty to skill without waiting on a shift lead.
SOC manager	Assignment quality becomes a reportable, auditable number rather than a judgment call that varies by shift.
CISO and risk	Every assignment carries a defensible rationale, reconstructable after the fact rather than dependent on who was on duty.
Service provider practice lead	Routing capacity scales with the customer base rather than with the number of triage leads on staff, and service commitments hold under surge without over-staffing for peak.

7.3 What sets it apart

▸ **Policy, not memory.** Tier and analyst selection follow a defined mechanism, so the same alert reaches the same class of decision regardless of who is on shift.

▸ **Workload-aware, not tier-blind.** The least-loaded eligible analyst is selected automatically, rather than the first one a triage lead happens to think of.

▸ **Elastic without added headcount.** Saturation routes eligible alerts to autonomous handling instead of breaching a commitment, and releases capacity back as load recovers.

THE VALUE IN ONE LINE.

A security operation without this function pays a person to decide who works each alert, and gets a different answer depending on who that person is. With it, the answer follows a policy, holds under surge, and can be explained afterwards.

SECTION 8

Summary

Intelligent Alert Routing is how the platform decides who works an alert. It is one core function among twelve, scoped to a five-stage assignment mechanism that runs continuously inside the TDIR core rather than as a decision made by whoever is triaging the queue that shift.

▸ Tier mapping, workload awareness, assignment decision, SLA overflow routing and state tracking replace a person deciding who works each alert.

▸ The score and band the function consumes come from the Risk Scoring Formula. Routing never re-scores an alert.

▸ Saturation routes eligible alerts to autonomous handling instead of breaching a commitment, and releases capacity back as load recovers.

▸ Every assignment is auditable, traceable to the score, tier and workload in force at the moment it was made.

▸ The function decides who works an alert, and leaves urgency and response to the core functions built for those decisions.

Forward-looking statement: any statement about future capability is forward-looking and non-binding, and capability described in the present tense is capability available at the publication date, subject to the deployment configuration in force.

ClearSkies™

CORE FUNCTION BRIEF

Generative and Agentic AI

A core function of the ClearSkies ilSOC platform

Two AI modalities, one governance gate, drafting and investigating on behalf of the function that owns the decision.

Contents

SECTION 1

Executive Summary

IN ONE SENTENCE.

Generative and Agentic AI is the core function that drafts, investigates and validates on behalf of every other core function that needs it, generative for narrative and explanation, agentic for correlation and candidate action, with every output passing the requesting function's own governance gate before it is used.

Drafting and investigating consume analyst time in a conventional security operation. Writing an incident summary takes nearly as long as the investigation itself. Correlating weak signals by hand is capped by the hours available rather than by what is worth investigating. And a candidate detection or response step that exists in someone's head still has to be typed, tested and reviewed before it helps anyone. None of this scales with alert volume, and none of it improves unless someone remembers to change the process next time.

Generative and Agentic AI answers this with two modalities behind one governed loop. Generative AI turns normalized context into a narrative, a summary, or a draft artifact. Agentic AI investigates, correlates and validates evidence, producing a candidate finding or a candidate action for review. Neither modality decides anything on its own authority: every output passes the review gate the requesting function defines before it reaches production, and the outcome of that review feeds back to improve the next draft.

The same two modalities serve every core function that needs them: Detection Factory for authoring and validating detection content, SOAR for investigation and candidate response options, Threat Hunting for hypothesis generation, and KPIs and SLAs for report drafting. One capability, bounded task by bounded task, rather than a separate AI feature built inside each.

THE FUNCTION IS PRECISELY SCOPED.

Generative and Agentic AI drafts and investigates on request. It does not compute a risk score, which is the Risk Scoring Formula. It does not decide who works an alert, which is Intelligent Alert Routing. And it does not authorize or execute a response, which is SOAR and Playbooks. Each of those is a separate core function with its own brief, and none of them is described here as part of this one.

Two modalities

One governance gate

Four consuming functions

Generative for narrative, agentic for investigation	Nothing production-facing bypasses review	One capability, bounded task by bounded task

SECTION 2

Why Manual Drafting and Investigation Fall Short

Drafting and investigation are underinvested relative to the rest of the security operation, and the consequences compound quietly rather than surfacing in an incident review. Four failure modes explain why, and each is answered by a named stage of the loop in section 3.

2.1 Narrative work consumes analyst time

Writing an incident summary, a forensic timeline, or an executive report by hand takes time comparable to the investigation itself, and the work starts over from a blank page for every incident regardless of how similar it is to the last one.

2.2 Investigation depth capped by available hours

An analyst can correlate only so many weak signals manually before the return stops justifying the hours. Deep investigation is reserved for alerts that already look serious, so a subtle pattern that never quite looks serious enough goes uninvestigated.

2.3 Draft-to-decision lag

A candidate detection, a candidate response step, or a report exists as an idea before it exists as something reviewable. Turning the idea into a draft takes time a live incident does not have, so the gap between knowing what should happen and having something to approve is where urgency is lost.

2.4 Improvement that depends on memory

The lesson from a closed incident lives in a person's head or a retrospective document, not in a structure that changes how the next similar draft is written or the next similar signal is investigated. The same gap gets rediscovered rather than closed.

2.5 The cost of the workaround

The usual response is more senior analyst time allocated to writing and correlating rather than to deciding, which is the scarcest and most expensive hour in the operation. Cost scales with incident volume and with reporting obligations, while the underlying problem, drafting and investigating that do not compound into anything reusable, is untouched.

Industry research consistently ties a substantial share of analyst time to report writing and manual correlation rather than to decision-making, though the specific figure is not yet sourced for this brief.

SECTION 3

How Generative and Agentic AI Works

Generative and Agentic AI is one continuously operating loop native to the TDIR core, invoked task by task rather than running unbounded. A sibling core function submits a bounded request at one end, and a governed, auditable output returns to it at the other.

> *Figure 1. The five-stage governance loop.*

3.1 The five stages

Each stage is named once here and referenced by name throughout the rest of this brief, paired with the failure mode from section 2 that it answers.

STAGE	WHAT HAPPENS	THE FAILURE MODE IT ANSWERS
Request	A sibling core function, Detection Factory, SOAR, Threat Hunting or KPIs and SLAs, submits a bounded task and the normalized context it requires.	Draft-to-decision lag
Generate	The generative modality drafts a narrative, translation, or candidate artifact, a summary, an explanation, or draft content, from that context.	Narrative work consumes analyst time
Investigate	The agentic modality investigates, correlates and validates evidence relevant to the task, producing a candidate finding or a candidate action for review.	Investigation depth capped by available hours
Govern	The generated or investigated output passes the review gate the requesting function defines, human or policy based, before it is used. Nothing production-facing skips this stage.	Draft-to-decision lag
Learn	The gate outcome, accepted, edited or rejected, feeds back to improve future drafts and investigations for that same task type.	Improvement that depends on memory

The two modalities are distinct in kind. Generative AI produces language: a narrative, a translation, a summary. Agentic AI produces a validated finding or a candidate action. Both are drafts until the Govern stage, and neither reaches production on its own authority.

3.2 How an output is judged

Quality is measured rather than assumed, on definitions a provider can report and a customer can audit. The measures below are reported natively and per task type. They are defined here because the same metric name is measured differently by different vendors, and target values are agreed per engagement rather than asserted.

MEASURE	THE DEFINITION USED	WHAT A POOR VALUE TRIGGERS
Gate acceptance rate	Outputs accepted without edits at the requesting function's gate, as a share of all outputs for that task type.	A review of the prompt, the context or the training signal for that task type.
Time to draft	Elapsed time from a task request to a reviewable output being ready for the gate.	Investigation of the specific stage introducing the latency.
Edit rate at gate	Outputs modified by the reviewer before use, as a share of all outputs for that task type.	A review of whether the edits reflect a pattern the Learn stage should absorb.
Investigation depth	Tasked alerts where agentic investigation added at least one corroborating or disconfirming signal beyond the original request.	A review of the context supplied with the task, which may be too narrow to work from.
Learning uptake	Gate decisions that measurably change draft quality for the same task type on subsequent runs.	A review of whether gate outcomes are being recorded in a form the Learn stage can use.

A rejected draft is a working gate, not a failed function. ***A task type with a nonzero rejection rate is not necessarily underperforming. The gate rejecting a candidate that does not meet the requesting function's bar, and the Learn stage absorbing why, is the mechanism protecting production quality working as intended.***

SECTION 4

Generative and Agentic AI in the Platform

4.1 What it draws on, what it writes back, and what it does not do

What it draws on. Normalized telemetry and the shared entity model from the Centric-AI Fabric; the bounded task and its context from the requesting core function; and, for the Learn stage, the prior gate outcomes recorded for that task type. Each is an input to a specific stage rather than a general capability the function generates itself.

What it writes back. The gated draft, finding or candidate, together with its audit trail, published into the shared model and consumed by the requesting core function. The relationship is one way: nothing returned to this function changes an output already gated, and the Learn stage draws on the requesting function's own recorded outcome for that task rather than on a separate, untraceable score.

What it does not do. It does not compute a risk score, decide who works an alert, or authorize or execute a response. It holds no telemetry of its own, and it owns no governance gate other than the general checkpoint every output passes: the criteria a specific gate applies belong to the function that requested the task.

4.2 A worked example

The chain below is illustrative and technically representative rather than drawn from a named engagement. Detection Factory submits a Request: draft and validate candidate detection logic from a confirmed hunting finding. Generate drafts three candidates generalized to the underlying technique, each tagged with the ATT&CK identifier Detection Factory supplied. Investigate tests the candidates against ninety days of normalized telemetry, producing a precision estimate for each. Govern applies Detection Factory's own promotion criteria: two candidates clear the threshold and one does not. Detection Factory promotes the two into production. Learn records the rejected candidate's shortfall against the threshold, which sharpens the next draft Generate produces for a similar finding.

4.3 Delivery across many tenants

The same two modalities and the same five-stage loop serve every tenant. What changes per tenant is the context supplied with each task and the gate criteria the requesting function applies. What travels between tenants is the mechanism and the task-type learning it accumulates; what never travels is one tenant's telemetry, drafts or gate outcomes. A provider can therefore explain any drafted or investigated output to any customer, from the same mechanism, without a bespoke AI build per tenant.

Because the function is switched on with the platform rather than deployed separately, time to value is a matter of the first task types being exercised rather than an onboarding project. In the first week, requesting functions begin submitting tasks and every output passes a gate from the outset. Through

the first month, gate acceptance rate and edit rate accumulate enough history per task type to be read as figures rather than defaults. At steady state, learning uptake is tracked as an ongoing measure of whether the loop is closing.

SECTION 5

ATT&CK Alignment and AI Governance

Generative and Agentic AI does not map detection content and does not compute technique relevance itself. Its generative modality translates a technique identifier Detection Factory has already attached into a human-readable narrative for an analyst, and its agentic modality assists Detection Factory in drafting and testing a candidate mapping, which Detection Factory alone promotes.

> **GOVERNANCE IS THE SUBJECT, NOT A SIDE NOTE.**
>
> Because this function is the platform's AI capability itself, its governance obligations are stated directly rather than folded into a framework table. Every generated or investigated output passes a human or policy review gate, the audit trail records what was generated, what was reviewed and what was decided, and no sentence in this brief implies an output reaches production without that review.

FRAMEWORK OR STANDARD	ITS ROLE IN GENERATIVE AND AGENTIC AI
MITRE ATT&CK (Enterprise, Cloud and Identity)	Technique identifiers translated into narrative form, and drafting assistance for Detection Factory's own mapping. Neither computed nor promoted by this function.
EU AI Act	The governance obligations attaching to the generative and agentic layer specifically: the review gate, the audit trail and the human decision point named in section 4.1.
NIST Cybersecurity Framework 2.0	Outcome language used where a generated report discusses aggregate posture to an executive or board audience.

Evidence, not compliance. The platform states which obligations the governed loop produces evidence for. It never states that the presence of a review gate makes an organization compliant, and control-level mapping is held by the Regulatory Frameworks core function.

SECTION 6

Questions Raised in Evaluation

The questions below recur in buyer evaluations, each answered directly against the mechanism described in section 3.

QUESTION, AS A BUYER ASKS IT	ANSWER
Does Generative and Agentic AI decide anything on its own authority?	No. It drafts and investigates against a bounded task. The requesting function's gate decides whether the output is used.
Does it compute the risk score or decide who handles an alert?	No. Those are the Risk Scoring Formula and Intelligent Alert Routing. This function is not described as performing either.
Does it ever execute a response action directly?	No. It may investigate and draft a candidate action. SOAR authorizes a response and Playbooks executes it.
What stops a poor-quality draft from reaching production?	The governance gate the requesting function defines. No sentence in this brief, and nothing in the platform's design, allows an output to bypass it.
How does it improve over time?	The Learn stage draws on the requesting function's own recorded gate outcomes, accepted, edited or rejected, for that task type.
Is this the same thing as the AI-SecOps Autonomous Analysts?	No. This is a core function that drafts and investigates on request. The AI-SecOps Autonomous Analysts are a native add-on with their own brief, licensed separately and operating under their own authorized scopes.
Does the same capability behave differently per tenant?	The task context and the gate criteria are configured per tenant. The five-stage mechanism itself is the same across every tenant.

SECTION 7

The Benefits

The value of an AI capability in security operations is settled by what it removes from an analyst's day without removing anyone's judgment from the decision. The outcomes below are countable: how much of the drafting starts from context rather than a blank page, how much investigation happens that would otherwise not have been afforded, and whether every one of those outputs can be shown to have been reviewed.

7.1 At a glance

BENEFIT	WHAT PRODUCES IT
Drafting starts from context rather than a blank page	Generate turns the normalized context supplied with the task into a first draft for review.
Investigation depth stops being capped by available hours	Investigate correlates and validates evidence on every tasked alert, not only the ones that already look serious.
Nothing production-facing is used unreviewed	Govern applies the requesting function's own gate to every generated or investigated output.
Quality improves without anyone remembering to improve it	Learn absorbs the accepted, edited or rejected outcome for each task type and shapes the next draft.
AI governance is evidenced rather than asserted	The audit trail records what was generated, what was reviewed and what was decided, output by output.
One capability rather than an AI feature per function	The same two modalities serve Detection Factory, SOAR, Threat Hunting and KPIs and SLAs from one mechanism.

7.2 What changes, and for whom

A governed drafting and investigation capability changes what each role spends time on, and what each role can show for it afterwards.

ROLE	WHAT CHANGES
SOC analyst	Time goes to reviewing and validating a draft rather than producing it from a blank page, and the investigation arrives with corroborating evidence already gathered.
SOC manager	Drafting and investigation throughput becomes a reportable, auditable number rather than an assumption about how busy analysts are.
CISO and risk	Executive-ready narrative is available without a manual writing cycle, and AI governance is evidenced by an audit trail rather than asserted in a policy document.
Service provider practice lead	The same drafting and investigation capacity applies across every tenant, scaling with the customer base rather than with headcount dedicated to writing and correlating.

7.3 What sets it apart

▸ **Bounded by the requesting function.** Every task is scoped by the core function that owns the decision, so drafting and investigating never substitute for that function's own governance.

▸ **Nothing skips review.** One governance checkpoint applies to every generated or investigated output before it reaches production, whichever modality produced it.

▸ **Learning uses the outcome the requester already measures.** Closed-loop improvement draws on the requesting function's own quality decision rather than on a separate, untraceable score nobody can audit.

THE VALUE IN ONE LINE.

An ungoverned AI produces output faster than anyone can check it. A governed loop produces output that has already been checked, by the function that owns the decision, with the record to prove it.

SECTION 8

Summary

Generative and Agentic AI is how the platform drafts and investigates on behalf of the functions that decide. It is one core function among twelve, scoped to a five-stage governance loop that runs continuously inside the TDIR core rather than as an unbounded intelligence claiming every decision in the security operation.

▸ Generative AI drafts narrative and explanation; agentic AI investigates and validates. Neither decides on its own authority.

▸ Every task is bounded by the core function that requested it, and every output passes that function's own governance gate.

▸ Learning draws on the requesting function's own recorded outcome, not a separate, untraceable score.

▸ The same two modalities serve Detection Factory, SOAR, Threat Hunting and KPIs and SLAs from one mechanism.

▸ The function drafts and investigates, and leaves scoring, assignment and response authorization to the core functions built for those decisions.

Forward-looking statement: any statement about future capability is forward-looking and non-binding, and capability described in the present tense is capability available at the publication date, subject to the deployment configuration in force. This document describes drafting and investigation capability and does not constitute legal advice on any customer's obligations under the EU AI Act or any other instrument.

ClearSkies™

CORE FUNCTION BRIEF

SOAR

A core function of the ClearSkies iISOC platform

Whether, how, and under what governance a confirmed incident is acted on.

Contents

SECTION 1

Executive Summary

IN ONE SENTENCE.

SOAR is the core function that decides whether, at what autonomy level, and through which tools a confirmed incident is acted on, orchestrating the response across every applicable control under a governance ladder that never means loss of control.

Detection tells an organization what is happening. Investigation tells it why. Response is where speed and precision decide the outcome, and in a conventional security operation response is manual: an analyst pivots across the SIEM, the endpoint console, the identity provider and a firewall, executing each containment step by hand while the attacker's window stays open. The same incident produces a different response depending on who is on shift, and reconstructing what was done afterwards means stitching together several tools' separate logs.

SOAR replaces that with a single orchestration decision. Once a detection is confirmed and scored, SOAR matches it to the applicable response sequence, checks the autonomy level configured for that use case and tenant, and coordinates execution across the native add-ons built to execute a response action, namely Endpoint Threat Monitoring and Response, Active Defense and Identity Threat Protection, and, through the Third-Party Add-ons Marketplace, every connected third-party tool exposing a response interface. Every action, approval and outcome is captured as it happens.

Autonomy is never all or nothing. A response use case sits on a ladder of four levels, assistive, semi-autonomous, approval-based and fully automated, selectable per use case and per tenant, so a security operation can automate isolating a confirmed-malicious endpoint while still requiring a human gate before disabling a privileged account.

THE FUNCTION IS PRECISELY SCOPED.

SOAR decides whether, how, and at what governance level a response executes. It does not decide how urgent the incident is, which is the Risk Scoring Formula. It does not decide who works it, which is Intelligent Alert Routing. And it does not define the action sequence it invokes, which is Playbooks. Each of those is a separate core function with its own brief.

Five stages
Trigger, match, autonomy, orchestrate, record

Four autonomy levels
Assistive through to fully automated

Every response control
Response-capable add-ons and connected tools

SECTION 2

Why Tool-by-Tool Response Falls Short

Response execution is underinvested relative to the rest of the security operation, and the consequences compound at the worst possible moment, during an active incident. Four failure modes explain why, and each is answered by a named stage of the mechanism in section 3 rather than by a faster version of the same manual workflow.

2.1 Manual response is slow and error-prone

An analyst pivoting across consoles to isolate an endpoint, disable an account and block a destination takes minutes an attacker's dwell time does not allow, and each manual step under pressure is a chance to miss one.

2.2 Response quality depends on who executes it

The same confirmed incident can produce a different response depending on which analyst is on shift and how familiar they are with a given tool's console, so consistency is a matter of who happened to be on duty rather than of policy.

2.3 All-or-nothing automation is too risky to adopt

A security operation that could automate part of its response often does not, because the only alternative to manual execution looks like unattended automation, and few teams accept that trade for anything beyond the most trivial case.

2.4 No consolidated record of what happened

When a response spans several consoles, reconstructing exactly what was done, by what or whom, and when, means stitching together separate logs after the fact rather than reading one record.

2.5 The cost of the workaround

The usual response is a detailed runbook and a well-drilled on-call rotation, which reduces variance but removes neither the manual execution time nor the reliance on a specific person being available. The cost scales with incident volume and with the number of tools in the response path, while the underlying problem, no single orchestration decision governing the response, is untouched.

Industry research on incident response consistently ties orchestrated, governed execution to materially shorter containment windows than manual, tool-by-tool response, though the specific improvement figure is not yet sourced for this brief.

SECTION 3

How SOAR Works

SOAR is one continuously operating mechanism native to the TDIR core, not a console a person operates. A confirmed, scored incident enters at one end, and a governed, audited response leaves at the other.

> *Figure 1. The five-stage orchestration mechanism.*

3.1 The five stages

Each stage is named once here and referenced by name throughout the rest of this brief, paired with the failure mode from section 2 that it answers.

STAGE	WHAT HAPPENS	THE FAILURE MODE IT ANSWERS
Trigger	A confirmed incident reaches SOAR, carrying the score and band the Risk Scoring Formula published and the technique context the Detection Factory attached.	No consolidated record of what happened
Playbook selection	The incident's technique and affected asset are matched to the applicable action sequence that Playbooks defines.	Response quality depends on who executes it
Autonomy check	The use case's configured autonomy level, assistive, semi-autonomous, approval-based or fully automated, is applied, and any required approval gate is raised.	All-or-nothing automation is too risky to adopt
Orchestrate	Execution is coordinated across every applicable response control: the native add-ons built to execute a response action, directly, and third-party tools through the Marketplace, invoking the matched playbook at the authorized level.	Manual response is slow and error-prone
Record	Every action, approval and outcome is captured to an immutable audit trail as it happens, feeding the next autonomy and playbook decision.	No consolidated record of what happened

The stages resolve in order: what the incident is, what sequence fits it, what level of oversight it requires, and only then execution. Autonomy is checked before anything runs, never inferred from what the orchestration layer happens to be capable of.

3.2 How a response is judged

Quality is measured rather than assumed, on definitions a provider can report and a customer can audit. The measures below are reported natively and per tenant. They are defined here because the same metric name is measured differently by different vendors, and target values are agreed per engagement rather than asserted.

MEASURE	THE DEFINITION USED	WHAT A POOR VALUE TRIGGERS
Mean time to respond	Elapsed time from a confirmed incident reaching SOAR to the response completing.	Investigation of the stage introducing the delay: playbook match, approval wait or execution.
Playbook match rate	Confirmed incidents matched to an applicable playbook automatically, as a share of all confirmed incidents.	A review of playbook coverage for the unmatched technique or asset type.
Autonomy level distribution	The share of responses executed at each level of the autonomy ladder.	A review of whether a use case sits at the right level for the tenant's risk tolerance.
Approval latency	Elapsed time from an approval gate being raised to a human decision, for approval-based use cases.	A review of on-call coverage, or of whether the use case belongs at a different autonomy level.
Audit completeness	Orchestrated actions with a complete record of the action, the approver where applicable and the outcome, as a share of all actions.	Investigation of the orchestration path that did not record its outcome.

A raised approval gate is the mechanism working, not a delay to eliminate. ***A use case configured as approval-based is meant to pause for a human decision. Removing that pause to improve mean time to respond would remove the governance the tenant chose for that class of action, which is a policy decision rather than a performance defect.***

SECTION 4

SOAR in the Platform

4.1 What it draws on, what it writes back, and what it does not do

What it draws on. The score and urgency band the Risk Scoring Formula publishes, the technique context the Detection Factory attaches, the normalized entity and asset context the Centric-AI Fabric carries, the tenant's autonomy-ladder policy and approval configuration, and the action sequence

Playbooks defines for the matched use case. Each is an input to a specific stage rather than a general capability the function generates itself.

What it writes back. The orchestration decision, the autonomy level applied, any approvals obtained and the outcome, published into the shared model as an immutable audit trail. Playbooks executes the sequence SOAR invokes, and KPIs and SLAs consumes the outcome for response-time reporting. The relationship is one way: nothing returned to SOAR changes a decision already made.

What it does not do. It does not compute a risk score, decide who works an incident, or define the content of the action sequence it invokes, which is Playbooks. It does not orchestrate through DNS Shield or Attack Surface Monitoring, which report findings to the TDIR core under the one-way design and receive no instruction back. Where a response requires investigation first, Generative and Agentic AI performs that on request. SOAR does not investigate, and it holds no telemetry of its own.

4.2 A worked example

The chain below is illustrative and technically representative rather than drawn from a named engagement. DNS Shield contributes a tunneling verdict and Identity Threat Protection adds phishing context, and the TDIR core correlates them into one incident: unusual DNS tunneling from a developer workstation to confidential repositories after hours, tied to a user who recently received a phishing email. Trigger receives it. Playbook selection matches the technique to the applicable containment sequence. Autonomy check finds the use case configured as fully automated for a non-privileged workstation account, so no gate is raised. Orchestrate invokes the playbook: isolate the workstation through Endpoint Threat Monitoring and Response, disable the account through Identity Threat Protection, and block the destination through a Marketplace-connected network control. Record captures the sequence for audit. Had the account carried privileged access, the same autonomy check would have raised an approval gate before the disable-account step.

4.3 Delivery across many tenants

Autonomy-ladder configuration, playbook mapping and approval policy are set per tenant and per use case, so one mechanism serves every tenant's risk tolerance without a bespoke rebuild. What travels between tenants is the mechanism itself; what never travels is one tenant's incident data, approvals or audit trail.

Because the function is switched on with the platform rather than deployed separately, time to value is a tuning exercise rather than an onboarding project. In the first week, confirmed incidents match against the default playbook set at an autonomy level held at assistive until risk tolerance is confirmed. Through the first month, autonomy levels are raised use case by use case as playbook match rate and approval latency show the mechanism behaving as expected. At steady state, autonomy level distribution and mean time to respond are tracked as tuning inputs, adjusted as infrastructure, staffing or risk appetite changes.

SECTION 5

ATT&CK Alignment and Supported Frameworks

SOAR does not map detection content and does not compute technique relevance. The technique identifier a confirmed incident carries, attached by the Detection Factory and carried through the Risk Scoring Formula's output, is what playbook selection matches against, and it is recorded in the orchestration audit trail so a reviewer can see which adversary behavior a given response addressed.

FRAMEWORK OR STANDARD	ITS ROLE IN SOAR
MITRE ATT&CK, Enterprise, Cloud and Identity	Carried through from the Detection Factory's tagging into playbook matching and the audit record. It is not computed or validated by this function.
MITRE D3FEND	Countermeasure vocabulary describing the response actions Playbooks executes, referenced in the orchestration record rather than authored here.
NIST Cybersecurity Framework 2.0	Outcome language used where platform reporting discusses response performance to an executive or board audience.

SECTION 6

Questions Raised in Evaluation

The questions below recur in buyer evaluations, each answered directly against the mechanism described in section 3.

QUESTION, AS A BUYER ASKS IT	ANSWER
Does SOAR decide what actions a playbook takes?	No. Playbooks defines the action sequence. SOAR decides whether, at what autonomy level, and through which tools a matched playbook executes.
Does automation mean losing control?	No. Every response sits on a configurable autonomy ladder with policy bounds, approval gates where configured, and a full audit trail.
What happens when a privileged account is involved?	The use case's configured autonomy level applies. A higher-sensitivity action can require an approval gate even when a lower-sensitivity action in the same incident runs automatically.

QUESTION, AS A BUYER ASKS IT	ANSWER
Does SOAR compute urgency or decide who works an incident?	No. Those are the Risk Scoring Formula and Intelligent Alert Routing.
Can a tenant choose different autonomy levels for different use cases?	Yes. The autonomy ladder is configured per use case and per tenant.
Does SOAR act on tools the organization does not already run?	It acts directly on the native add-ons built to execute a response action, namely Endpoint Threat Monitoring and Response, Active Defense and Identity Threat Protection, and on third-party tools through the Marketplace where the vendor exposes a response interface. DNS Shield and Attack Surface Monitoring report findings to the core under the one-way design and are not orchestration targets. No existing tool has to be replaced.

SECTION 7

The Benefits

Governed orchestration is where a security operation stops paying for console pivoting, and the outcomes are countable: how long containment takes, how much of the response ran without a human, and whether the record stands up to a reviewer.

7.1 At a glance

BENEFIT	WHAT PRODUCES IT
Containment happens in seconds rather than console pivots	Orchestrate coordinates every applicable control from one decision, rather than an analyst working through them in turn.
The same incident produces the same response on any shift	Playbook selection matches technique and asset to a defined sequence, rather than to an analyst's familiarity with a console.
Automation becomes adoptable rather than all or nothing	A four-level autonomy ladder, configured per use case and per tenant, with approval gates where the tenant wants them.

BENEFIT	WHAT PRODUCES IT
A privileged action can be gated while a routine one runs	Autonomy check applies per use case, so sensitivity decides oversight within a single incident.
The record is one trail rather than several consoles	Record captures every action, approval and outcome to an immutable trail as it happens.
Response performance becomes a tunable number	Five measures defined in section 3.2, reported per tenant and per use case.

7.2 What changes, and for whom

Governed orchestration changes what each role spends time on, and what each role can see.

ROLE	WHAT CHANGES
SOC analyst	Time goes to supervisory judgment and approval decisions rather than to repetitive, tool-by-tool execution.
SOC manager	Autonomy-level tuning and mean time to respond become visible, adjustable numbers rather than an impression of how the last incident went.
CISO and risk	Every automated action is bounded by policy and produces a defensible, immutable record.
Service provider practice lead	Governed automation scales response across tenants without a linear increase in analyst headcount, and autonomy is tuned by customer and by tier.

7.3 What sets it apart

▸ **Governed by design, not bolted on.** Every response sits on a selectable autonomy ladder rather than on a binary choice between manual and unattended.

▸ **It fires on full context.** Orchestration happens inside the same core that correlated the incident, so the decision uses the enriched picture rather than a single stripped-down alert.

▸ **One coordination layer, every response control.** The native add-ons built for response and, through the Marketplace, connected third-party tools are coordinated from the same orchestration decision.

THE VALUE IN ONE LINE.

Manual response asks an analyst to be fast, accurate and consistent across four consoles during the worst hour of the week. Governed orchestration makes that one decision, bounded by a policy the tenant chose, with a record that survives the review afterwards.

SECTION 8

Summary

SOAR is how the platform decides whether, how, and under what governance a confirmed incident is acted on. It is one core function among twelve, scoped to a five-stage orchestration mechanism that runs continuously inside the TDIR core rather than as a console an analyst operates by hand.

▸ Trigger, playbook selection, autonomy check, orchestrate and record replace tool-by-tool manual response.

▸ The score and technique context the function consumes come from the Risk Scoring Formula and the Detection Factory. SOAR computes neither itself.

▸ Every response sits on a configurable autonomy ladder, assistive through to fully automated, selectable per use case and per tenant.

▸ Playbooks defines the action sequence, and orchestration reaches only the native add-ons built to execute a response action and Marketplace-connected third-party tools.

▸ The function orchestrates and governs the response, and leaves urgency, assignment and the action sequence itself to the core functions built for those decisions.

Forward-looking statement: any statement about future capability is forward-looking and non-binding, and capability described in the present tense is capability available at the publication date, subject to the deployment configuration in force.

ClearSkies™

CORE FUNCTION BRIEF

Playbooks

A core function of the ClearSkies iISOC platform

The defined, verified action sequences that carry out the response SOAR authorizes.

Contents

SECTION 1

Executive Summary

IN ONE SENTENCE.

Playbooks is the core function that defines, executes and verifies the specific containment, mitigation and recovery steps a response carries out, at the autonomy level SOAR authorizes, across whichever domain the incident touches: endpoint, identity, email, cloud or network.

A containment procedure in a conventional security operation lives in a runbook document or in an experienced responder's memory, and is executed a little differently each time. A multi-step response, isolate a host, disable an account, revoke a token, can stop partway without anyone confirming that every step actually completed. A sequence that has quietly gone stale as a tool's interface changed is not discovered until it fails during a real incident.

Playbooks replaces improvised execution with a maintained catalog of versioned action sequences. Once SOAR authorizes a specific sequence at a specific autonomy level for a confirmed incident, Playbooks executes the steps across the domains involved, confirms that each one succeeded, and records the outcome. A sequence that fails or drifts out of date is flagged for revision rather than left to fail again during the next incident.

Where a sequence needs to adapt to a specific incident, or a report needs drafting once execution completes, Playbooks requests that from Generative and Agentic AI, and the draft passes Playbooks' own review before it changes what executes or what is reported. Nothing adapts unreviewed mid-incident.

THE FUNCTION IS PRECISELY SCOPED.

Playbooks defines and executes the specific action sequence. It does not decide whether, when, or at what autonomy level a response runs, which is SOAR. It does not decide how urgent the incident is, which is the Risk Scoring Formula. And it does not draft or investigate on its own authority, which is Generative and Agentic AI. Each of those is a separate core function with its own brief.

Five stages

Author, invoke, execute, verify, record and learn

Five domains

Endpoint, identity, email, cloud, network

Every step verified

Nothing marked complete unconfirmed

SECTION 2

Why Ad Hoc Execution Falls Short

Response execution is underinvested relative to the rest of the security operation, and the consequences surface only when a sequence is trusted but has not actually run to completion. Four failure modes explain why, and each is answered by a named stage of the mechanism in section 3.

2.1 Steps recreated from memory each time

A containment procedure that lives in a document or in a responder's experience gets executed a little differently every time, depending on who is handling the incident and how well they remember the last one.

2.2 Partial execution goes unnoticed

A multi-step response can stop partway, leaving a token that did not actually revoke or an account still active, without anyone confirming that every step in the sequence completed as expected.

2.3 Sequences that do not fit the domain touched

A generic instruction to contain an incident does not specify what containment means for an email account, a cloud workload or a network segment, so responders adapt on the fly, inconsistently, for each domain.

2.4 Nothing measures which sequences work

A sequence that quietly fails a step, or has gone stale as a tool's interface changed, is not caught until it fails visibly during a real incident, because nothing tracks whether the catalog still matches the tools it calls.

2.5 The cost of the workaround

The usual response is a manually maintained runbook library and a reliance on senior responders who know which steps actually still work. The cost scales with the number of tools and domains in scope, while the underlying problem, sequences that are not versioned, verified or measured, is untouched.

Industry research on incident response consistently ties verified, automated execution to shorter containment times than manually run runbooks, though the specific improvement figure is not yet sourced for this brief.

SECTION 3

How Playbooks Works

Playbooks is one continuously operating mechanism native to the TDIR core, not a document a responder consults. SOAR's authorization enters at one end, and a verified, audited execution leaves at the other.

> *Figure 1. The five-stage authoring and execution mechanism.*

3.1 The five stages

Each stage is named once here and referenced by name throughout the rest of this brief, paired with the failure mode from section 2 that it answers.

STAGE	WHAT HAPPENS	THE FAILURE MODE IT ANSWERS
Author	An action sequence is defined and versioned: ordered steps mapped to the domain and to the incident class it addresses. A candidate sequence may be drafted or adapted with Generative and Agentic AI's help, gated by Playbooks' own review before it enters the catalog.	Steps recreated from memory each time
Invoke	SOAR's orchestration decision hands Playbooks a specific sequence and an authorized autonomy level for a confirmed incident. Playbooks selects neither the sequence nor the level itself.	Sequences that do not fit the domain touched
Execute	The sequence runs step by step across the domains involved, isolating, disabling, revoking or blocking, chaining steps where the sequence specifies it.	Sequences that do not fit the domain touched
Verify	Each action's outcome is confirmed, endpoint isolated, token revoked, account disabled, before the sequence is marked complete.	Partial execution goes unnoticed
Record and learn	The confirmed result, and any deviation from the expected outcome, is captured to the audit trail, and the sequence is flagged for review if it underperforms.	Nothing measures which sequences work

Playbooks never advances from Invoke to Execute without an authorization from SOAR attached to a specific incident, and never marks a sequence complete at Verify without confirming every step in it.

3.2 How a sequence is judged

Quality is measured rather than assumed, on definitions a provider can report and a customer can audit. The measures below are reported natively and per tenant. They are defined here because the same metric name is measured differently by different vendors, and target values are agreed per engagement rather than asserted.

MEASURE	THE DEFINITION USED	WHAT A POOR VALUE TRIGGERS
Step success rate	Individual actions verified successful on first attempt, as a share of all actions executed.	Investigation of the specific step and the tool integration behind it.
Full-sequence completion rate	Sequences completing every step without manual intervention, as a share of all invocations.	A review of the sequence for a step that needs redesign.
Time to verified containment	Elapsed time from invocation to every step in the sequence being verified.	Investigation of the step introducing the delay.
Sequence currency	Catalog sequences reviewed or updated within a defined period, as a share of the catalog.	Prioritized review of the oldest sequences in the catalog.
Domain coverage	Technology domains with at least one defined, current sequence, of the five in scope.	Authoring work for the domain left uncovered.

A flagged sequence is the mechanism working, not a hidden failure. ***A sequence that Record and learn flags for revision has been caught before it fails silently during a live incident. Treating a flagged sequence as a defect in the platform, rather than as the intended outcome of Verify doing its job, misreads what the mechanism is for.***

SECTION 4

Playbooks in the Platform

4.1 What it draws on, what it writes back, and what it does not do

What it draws on. SOAR's invocation, naming the sequence and the authorized autonomy level for a confirmed incident; the normalized entity and asset context the Centric-AI Fabric carries, used to resolve the specific target of each action; and, where a sequence is drafted or adapted, Generative and Agentic AI's output, gated by Playbooks' own review. Each is an input to a specific stage rather than a general capability the function generates itself.

What it writes back. The confirmed result of the executed sequence, step by step and overall, published as part of the shared audit trail. SOAR's own Record stage consumes it to close the orchestration loop, and KPIs and SLAs consumes it for containment-time reporting. The relationship is one way: nothing returned to Playbooks changes an outcome already recorded.

What it does not do. It does not decide whether, when, or at what autonomy level a response runs, which is SOAR. It does not compute a risk score or decide who is assigned. It does not investigate or draft on its own authority: where a sequence needs adaptation or a report needs drafting, Generative and Agentic AI performs that on request, under its own gate. It does not execute through DNS Shield or Attack Surface Monitoring, which report findings to the TDIR core under the one-way design and receive no instruction back. It holds no telemetry of its own.

Execution reaches the native add-ons built to carry out a response action, principally Endpoint Threat Monitoring and Response for endpoint steps and Identity Threat Protection for identity steps, with Active Defense where a deception action applies. For a domain the native add-ons do not cover directly, email or cloud for example, a step executes through a third-party tool connected via the Third-Party Add-ons Marketplace, where the vendor exposes a response interface.

4.2 A worked example

The chain below is illustrative and technically representative rather than drawn from a named engagement.

STEP	DOMAIN	THE CONFIRMED RESULT
Isolate the affected endpoint	Endpoint	Host removed from the network, confirmed by loss of active agent heartbeat.
Suspend the associated domain account	Identity	Account disabled, confirmed by directory query.
Scan peer endpoints for similar behavior	Endpoint, network	No secondary indicators found on related hosts.
Draft the incident summary for the queue	Requested from Generative and Agentic AI, gated by Playbooks	Summary reviewed and issued: ransomware identified, host isolated, credentials disabled, no lateral spread observed.

The isolate step executes through Endpoint Threat Monitoring and Response and the suspend step through Identity Threat Protection. SOAR authorized this sequence at a fully automated level for this asset class. Had the account carried privileged access, SOAR's own autonomy check would have raised an approval gate before Playbooks executed the account-suspension step.

4.3 Delivery across many tenants

The action-sequence catalog and its domain mappings are configured to each tenant's tools and policy, so the same five-stage mechanism serves every tenant without a bespoke rebuild. What travels between tenants is the catalog structure and its versioning discipline; what never travels is one tenant's incident data, execution outcomes or audit trail.

Because the function is switched on with the platform rather than deployed separately, time to value is a matter of mapping the catalog to a tenant's own tools rather than an onboarding project. In the first week, the default catalog is mapped to the domains and tools a tenant already runs, so SOAR can invoke a first sequence as soon as autonomy levels are configured. Through the first month, step success rate and full-sequence completion rate surface any sequence needing redesign for that tenant's tool versions. At steady state, sequence currency is tracked as an ongoing measure, reviewed whenever a tool's interface changes or a new domain comes into scope.

SECTION 5

ATT&CK and D3FEND Alignment

Playbooks does not map detection content and does not compute technique relevance. Sequences in the catalog are tagged by the technique class they address, so SOAR's playbook-selection stage can match efficiently, and the technique identifier itself is attached by the Detection Factory and carried through the incident SOAR invokes Playbooks against.

Of the frameworks below, MITRE D3FEND is the one that describes this subject most directly. Where ATT&CK is the vocabulary for adversary technique, D3FEND is the vocabulary for the countermeasure, and Playbooks is the function that turns a D3FEND-classified countermeasure, isolate, revoke, block, into an executed and verified action.

FRAMEWORK OR STANDARD	ITS ROLE IN PLAYBOOKS
MITRE D3FEND	Countermeasure vocabulary for the specific containment, mitigation and recovery actions Playbooks defines and executes.
MITRE ATT&CK, Enterprise, Cloud and Identity	Carried through from the Detection Factory's tagging to classify sequences in the catalog by the technique class they address. It is not computed by this function.
NIST Cybersecurity Framework 2.0	Outcome language used where platform reporting discusses containment and recovery performance to an executive or board audience.

SECTION 6

Questions Raised in Evaluation

The questions below recur in buyer evaluations, each answered directly against the mechanism described in section 3.

QUESTION, AS A BUYER ASKS IT	ANSWER
Does Playbooks decide when or at what autonomy level a response runs?	No. SOAR decides that and invokes Playbooks with a specific sequence and level. Playbooks executes what it is authorized to execute.
What happens if a step fails partway through?	Verify catches it. The sequence is not marked complete, and the incomplete outcome is recorded for review rather than assumed successful.
Does AI adapt a sequence on its own during a live incident?	No. Any adaptation is drafted by Generative and Agentic AI on request and passes Playbooks' own review before it changes what executes. Nothing adapts unreviewed mid-incident.
How does Playbooks stay current as tools change?	Sequence currency is tracked and reported. A sequence that fails or goes stale is flagged for update or retirement rather than left in the catalog untested.
Can sequences be customized per tenant?	Yes. Sequences are configured to each tenant's tools and policy, drawn from the same catalog structure.
Does Playbooks decide who approves an action?	No. Approval follows SOAR's autonomy check and the tenant's policy. Playbooks executes once SOAR has authorized the sequence and the level.

SECTION 7

The Benefits

A maintained catalog is where a response stops depending on who remembers the procedure, and the outcomes are countable: whether every step actually completed, whether the catalog still matches the tools it calls, and how long verified containment takes.

7.1 At a glance

BENEFIT	WHAT PRODUCES IT
A sequence completes, or the gap is visible	Verify confirms each action's outcome before the sequence is marked complete.
The same incident class is handled the same way every time	A versioned catalog entry, rather than a procedure recalled from the last incident.
Containment fits the domain it is applied to	Sequences are authored per domain, across endpoint, identity, email, cloud and network.
A stale sequence is caught before an incident finds it	Record and learn flags an underperforming sequence, and sequence currency is reported.
Adaptation never happens unreviewed	Generative and Agentic AI drafts on request, and the draft passes Playbooks' own review before it changes what executes.
Execution quality becomes a reportable number	Five measures defined in section 3.2, reported per tenant and per domain.

7.2 What changes, and for whom

Verified, catalog-driven execution changes what each role spends time on, and what each role can trust.

ROLE	WHAT CHANGES
SOC analyst	Time goes to monitoring and approving execution rather than performing each containment step by hand.
SOC manager	Sequence currency and completion rates become visible, manageable numbers rather than an assumption that the runbook still works.
CISO and risk	Every action produces a verified, audited outcome, which supports the account a regulator or auditor asks for.
Service provider practice lead	The same maintained catalog executes consistently across every tenant, so response quality does not depend on which sequence a specific responder remembers.

7.3 What sets it apart

▸ **Verified, not assumed.** Every action's outcome is confirmed before a sequence is marked complete, which is the difference between a response that ran and a response believed to have run.

▸ **A maintained catalog, not a static runbook.** Sequences are versioned, reviewed for currency, and retired when they no longer match the tools they call.

▸ **Drafting on request, never on its own authority.** Where a sequence needs adapting, Generative and Agentic AI performs that under its own gate, and Playbooks reviews the result before anything executes.

> **THE VALUE IN ONE LINE.**
>
> A runbook tells a responder what should happen. A verified sequence records what did happen, step by step, and says so plainly when a step did not.

SECTION 8

Summary

Playbooks is how the platform carries out the response SOAR authorizes. It is one core function among twelve, scoped to a five-stage mechanism that defines, executes and verifies action sequences rather than deciding whether or how governed a response should be.

▸ Author, invoke, execute, verify, and record and learn replace ad hoc, memory-dependent execution.

▸ SOAR decides whether, at what level, and through which tools a response runs. Playbooks executes the specific sequence SOAR invokes, reaching only the native add-ons built for response and Marketplace-connected third-party tools.

▸ Every action is verified before a sequence is marked complete, and a failed or stale sequence is flagged for revision.

▸ A sequence spans whichever of the five domains the incident touches, and adapts or drafts a report only through Generative and Agentic AI's own governed request.

▸ The function executes and verifies the response, and leaves the decision to run it, and how urgently, to the core functions built for those decisions.

Forward-looking statement: any statement about future capability is forward-looking and non-binding, and capability described in the present tense is capability available at the publication date, subject to the deployment configuration in force.

ClearSkies™

CORE FUNCTION BRIEF

KPIs and SLAs

A core function of the ClearSkies iISOC platform

Live status, breach risk flagged before the deadline, and metrics that are never read alone.

Contents

SECTION 1

Executive Summary

IN ONE SENTENCE.

KPIs and SLAs is the core function that defines what to measure, tracks every open incident against its live target, predicts a breach before the deadline passes, and reports metrics in pairs so a single green number never hides the whole picture.

A metric nobody acts on is dashboard clutter rather than a control. A monthly spreadsheet reports a breach after it happened rather than flagging it while there is still time to act. A compliance rate on its own can hide an unhappy customer who is technically satisfied. And a report built for an executive rarely serves the analyst who needs a different number at a different moment.

KPIs and SLAs answers this with a five-stage mechanism. Measurement is organized into four layers, executive, service-level, operational and experience, each built for the decision its audience owns. Every open incident carries a live countdown against its applicable target, and a breach-risk model scores tickets before the deadline passes rather than after. The resulting status and risk flags are published into the shared model for other core functions to act on, and metrics are always reported in pairs, delivery against perception, speed against stability, on a cadence matched to each audience.

Because the four layers draw on the same underlying data, an executive and an analyst are never looking at figures reconciled from separate systems. They are looking at the same measurement stack rendered at a different altitude, so a number cited in a board report and a number an analyst investigates trace back to one source rather than two.

THE FUNCTION IS PRECISELY SCOPED.

KPIs and SLAs measures, tracks and reports. It does not decide how urgent an incident is, which is the Risk Scoring Formula. It does not decide who works it or reassign it, which is Intelligent Alert Routing. And it does not escalate or execute a response, which is SOAR and Playbooks. It defines no commercial tier or price: that is a platform commercial decision rather than a measurement one.

Five stages
Define, track, predict, publish, report

Four layers
Executive, service-level, operational, experience

Read in pairs
Never delivery without perception

SECTION 2

Why Unmanaged Reporting Falls Short

Measurement is underinvested relative to the rest of the security operation, and the consequences surface as a number nobody trusted rather than as a single visible failure. Four failure modes explain why, and each is answered by a named stage of the mechanism in section 3.

2.1 A metric that never changes a decision

A dashboard full of numbers nobody acts on is clutter rather than control, and without a defined trigger attached to each one there is no way to tell which numbers matter until someone decides that by convention, inconsistently, each time. The dashboard grows more crowded with every reporting cycle without becoming more useful.

2.2 Breaches discovered after they happen

A spreadsheet compiled monthly reports what already went wrong. By the time a breach appears in a report the window to act pre-emptively has closed, and the report becomes a record of the miss rather than a chance to prevent it.

2.3 A single metric hides the truth

Compliance without customer-experience data hides an unhappy customer who is technically satisfied, and response speed without reliability data hides a service that reacts fast but fails often. Either number alone reads as good news that a paired figure would immediately complicate.

2.4 The same numbers, retold at the wrong altitude

An executive drowning in ticket-level detail and an analyst missing the one number that matters right now are both symptoms of a single report trying to serve every audience at once. Neither gets the number their decision actually needs.

2.5 The cost of the workaround

The usual response is a manually compiled report assembled before each business review, reconciling numbers from several consoles by hand. The cost scales with the number of audiences and cadences a provider has to serve, while the underlying problem, metrics that are not organized by decision and not tracked live, is untouched.

Industry research on service management consistently ties manual, spreadsheet-based reporting to significant analyst and manager time, and separately finds customer churn linked to experience decline that a compliant number alone did not surface, though the specific figures are not yet sourced for this brief.

SECTION 3

How KPIs and SLAs Works

KPIs and SLAs is one continuously operating mechanism native to the TDIR core, not a report compiled before a meeting. Incident activity enters at one end, and a live, paired, audience-matched measurement leaves at the other.

> *Figure 1. The five-stage measurement mechanism.*

3.1 The five stages

Each stage is named once here and referenced by name throughout the rest of this brief, paired with the failure mode from section 2 that it answers.

STAGE	WHAT HAPPENS	THE FAILURE MODE IT ANSWERS
Define	Metrics are organized into four layers, executive, service-level, operational and experience, each drawing on the same underlying data but built for the decision its audience owns.	The same numbers, retold at the wrong altitude
Track	Every open incident carries a live countdown, continuously compared against its applicable target as elapsed time accrues, with a defined status: on track, approaching, at risk or breached.	A metric that never changes a decision
Predict	A model scores each open incident for breach risk from incident metadata, analyst workload, ticket age and external context, moving a ticket to at risk before its deadline passes rather than after.	Breaches discovered after they happen
Publish	The current status and any at-risk flag are published into the shared model, for Intelligent Alert Routing and SOAR to act on. This function does not itself reprioritize, reassign or escalate.	A metric that never changes a decision
Report	Metrics are always delivered in pairs, delivery against perception, speed against stability, on a cadence matched to each audience.	A single metric hides the truth

Predict hands off at the flag. What happens to an at-risk ticket after that, whether it is reassigned, escalated or accelerated, is a decision Intelligent Alert Routing or SOAR makes, using the flag this function published.

3.2 How a prediction is judged

Quality is measured rather than assumed, on definitions a provider can report and a customer can audit. The measures below are reported natively and per tenant. They are defined here because the same metric name is measured differently by different vendors, and target values are agreed per engagement rather than asserted.

MEASURE	THE DEFINITION USED	WHAT A POOR VALUE TRIGGERS
Prediction accuracy	Breach-risk predictions confirmed correct against the actual outcome, as a share of all predictions.	A retraining review of the model against recent misses.
Lead time	Elapsed time between an at-risk flag and the incident's actual deadline.	A review of whether the flag threshold fires early enough to be useful.
Pairing completeness	Published KPIs presented with their defined paired counter-metric, as a share of all published KPIs.	A review of the report template for the missing pair.
Report timeliness	Scheduled reports delivered on their configured cadence, as a share of all scheduled reports.	Investigation of the cadence or the audience layout that is slipping.
Metric-to-decision rate	Published metrics tied to a named decision or trigger, as a share of the full metric set.	A review of the audience layer carrying metrics nobody acts on.

A flag that does not become a breach is not a false alarm. ***An at-risk ticket that is reassigned in time and never breaches is Predict and Publish working exactly as intended. Judging the model only by how many predicted breaches actually occurred undercounts every one it helped prevent.***

SECTION 4

KPIs and SLAs in the Platform

4.1 What it draws on, what it writes back, and what it does not do

What it draws on. Incident metadata, timing and the score the Risk Scoring Formula publishes; assignment and workload data from Intelligent Alert Routing; action and outcome timestamps from SOAR and Playbooks; and each tenant's configured target for the incident in question. Each is an input to a specific stage rather than a general capability the function generates itself.

What it writes back. Computed KPIs, live status and breach-risk flags, published into the shared model. Intelligent Alert Routing and SOAR consume an at-risk flag to consider reprioritization or escalation, and scheduled reports deliver the paired metrics to each audience. The relationship is one way: nothing returned to this function changes a status or a prediction already published.

What it does not do. It does not decide urgency, assignment or response, and it defines no commercial service tier or price, which is a platform commercial decision rather than a measurement one. It holds no telemetry of its own beyond the incident metadata and timing already present in the shared model.

4.2 A worked example

The chain below is illustrative and technically representative rather than drawn from a named engagement. A critical incident carrying a fast response commitment is assigned to an analyst already holding a heavy ticket load. Track shows the countdown approaching its threshold. Predict combines the analyst's current load, the historical resolution time for this incident type and the elapsed time into a high breach-risk score, moving the ticket to at risk well before the deadline. Publish surfaces the flag. Intelligent Alert Routing reads it and reassigns the ticket to an available analyst at the appropriate tier, and the incident resolves inside its commitment. Report shows the save the next morning: the prediction, the reassignment and the met deadline read together rather than as a bare compliance percentage, so the manager sees why the commitment held rather than only that it did.

4.3 Delivery across many tenants

Targets, report cadence and audience layout are configured per tenant, so the same five-stage mechanism serves every tenant's commitments without a bespoke reporting build. What travels between tenants is the measurement mechanism itself; what never travels is one tenant's incident data or performance history. The layers, the pairing discipline and the breach-risk model are the same framework applied to a different set of targets and a different incident history, so adding a customer does not add a reporting project.

Because the function is switched on with the platform rather than deployed separately, time to value is a matter of configuring targets and layers rather than an onboarding project. In the first week, targets are configured and Track begins carrying a live countdown for every open incident. Through the first month, Predict accumulates enough history for prediction accuracy and lead time to become meaningful per-tenant figures rather than defaults. At steady state, pairing completeness and report timeliness are tracked as ongoing measures.

SECTION 5

Compliance and Reporting Framework Alignment

KPIs and SLAs does not map detection content and does not compute technique relevance, so MITRE ATT&CK plays a minimal role for this subject. The frameworks that matter here are the ones governing outcome reporting and audit evidence, which the function is built to produce.

FRAMEWORK OR STANDARD	ITS ROLE IN KPIS AND SLAS
NIST Cybersecurity Framework 2.0	Outcome language for the executive layer of the measurement stack, used in board and executive reporting on overall security posture rather than on any single incident.
ISO/IEC 27001 and 27002	Control mapping and the audit evidence a compliance report produces, drawn from the same measurement stack an internal audience already reads.
NIS2, DORA and GDPR	European reporting, resilience and privacy obligations the framework evidences, each with its own reporting cadence and audience that the Define stage can accommodate as a layer.

Reporting evidences a duty. It does not discharge one. The platform states which obligations the measurement stack produces evidence for. It never states that reporting alone makes an organization compliant, and control-level mapping is held by the Regulatory Frameworks core function.

SECTION 6

Questions Raised in Evaluation

The questions below recur in buyer evaluations, each answered directly against the mechanism described in section 3.

QUESTION, AS A BUYER ASKS IT	ANSWER
Does KPIs and SLAs decide how to respond to a breach risk?	No. It publishes the at-risk flag. Intelligent Alert Routing and SOAR decide whether to reprioritize, reassign or escalate. The function's role ends at a correctly timed, correctly scoped flag.
Does this framework define commercial service tiers or pricing?	No. Tier and pricing structures are commercial decisions made elsewhere. This function measures and reports against whatever target a tenant is configured with.
How is a breach prediction validated?	Prediction accuracy is tracked against actual outcomes, and lead time, meaning how much warning a flag gives, is reported alongside it, so the model is judged on both correctness and timeliness.
Can a metric be reported without its paired counterpart?	The framework's own discipline is to report metrics in pairs. A metric presented alone is treated as an incomplete picture rather than a valid report, whatever audience is reading it.
Does the same dashboard serve an analyst and an executive?	No. The measurement stack organizes into four layers by audience, and each sees the numbers built for the decision they own rather than a version of someone else's report.
Can reporting cadence be customized per tenant?	Yes. Daily, weekly and monthly cadences are configured per tenant and per audience, though the underlying data is the same regardless of how often a given audience sees it.

SECTION 7

The Benefits

Measurement is where a security operation either proves what it delivered or asks to be believed, and the outcomes are countable: how many commitments were saved rather than reported on, how many numbers tie to a decision, and whether a board figure and an analyst figure come from the same place.

7.1 At a glance

BENEFIT	WHAT PRODUCES IT
A commitment at risk is saved rather than reported on	Predict scores an open incident before the deadline, and Publish surfaces the flag while there is still time to act.
One green number cannot stand in for the picture	Report pairs delivery with perception and speed with stability, every time.
Each audience gets the number its decision needs	Define organizes measurement into four layers drawing on the same underlying data.
A board figure and an analyst figure agree	One measurement stack rendered at different altitudes, rather than two systems reconciled by hand.
A metric that changes nothing does not survive	Metric-to-decision rate, defined in section 3.2, makes an unused metric visible.
Audit evidence is a by-product of operating	The same stack an internal audience reads produces the control mapping a compliance report needs.

7.2 What changes, and for whom

Live, paired measurement changes what each role spends time on, and what each role can act on before it becomes a problem.

ROLE	WHAT CHANGES
SOC analyst	A shift starts with a prioritized queue and a predicted risk rather than a backlog and a breach discovered in a monthly report.
SOC manager	Workload distribution and assurance become visible, real-time numbers, so a staffing decision is informed by the current queue rather than last month's.
CISO and risk	Executive KPIs arrive paired and at the altitude a board needs, with evidence traceable to a defined framework rather than assembled ahead of each review.
Service provider practice lead	The same measurement stack reports consistently across every tenant, without a bespoke build per customer or per cadence.

7.3 What sets it apart

▸ **Every metric changes a decision.** A number not tied to a defined trigger does not earn a place in the framework, so the dashboard stays a working tool rather than accumulating figures nobody consults.

▸ **Breach risk flagged before the deadline, not after.** Predict surfaces an at-risk ticket while there is still time to act on it, which turns the deadline into a threshold that triggers action rather than a line a report crosses afterwards.

▸ **Metrics are read in pairs, never alone.** Delivery is always paired with perception and speed with stability, so one green number cannot hide the whole picture from the audience reading it.

> **THE VALUE IN ONE LINE.**
>
> Retrospective reporting tells an organization what it missed. Live, paired measurement tells it what it is about to miss, in time to do something, and shows the number next to the one that would contradict it.

SECTION 8

Summary

KPIs and SLAs is how the platform proves that the rest of it is working. It is one core function among twelve, scoped to a five-stage measurement mechanism that runs continuously inside the TDIR core rather than as a report compiled before a business review.

▸ Define, track, predict, publish and report replace unmanaged, retrospective measurement.

▸ Every open incident is tracked continuously against its target, and breach risk is flagged before the deadline passes.

▸ The function publishes the at-risk flag. Intelligent Alert Routing and SOAR decide what happens next.

▸ Metrics are always read in pairs and delivered at the altitude each audience needs.

▸ The function measures, tracks and reports, and leaves urgency, assignment, response and commercial structure to the functions and decisions built for them.

Forward-looking statement: any statement about future capability is forward-looking and non-binding, and capability described in the present tense is capability available at the publication date, subject to the deployment configuration in force. This document describes measurement and reporting capability and does not constitute legal advice on any customer's regulatory obligations.

ClearSkies™

CORE FUNCTION BRIEF

Regulatory Frameworks

A core function of the ClearSkies iISOC platform

One detection, mapped once through a shared technique language, evidences every regime that references it.

Contents

SECTION 1

Executive Summary

IN ONE SENTENCE.

Regulatory Frameworks is the core function that decomposes a regulation's obligations into MITRE ATT&CK techniques, so a single detection automatically produces evidence for every framework whose controls reference that technique, without separate mapping work per regulation.

Compliance evidence is conventionally assembled once a year under audit deadline pressure, framework by framework, because nothing links the same underlying control across the several regulations that reference it. An organization subject to five regimes maps the same control five separate times in five separate spreadsheets. A framework gets marked addressed in a tracker without a clear technical basis for the claim. And an organization operating across several jurisdictions has no single, current view of which frameworks are meaningfully covered rather than nominally listed.

Regulatory Frameworks answers this with a five-stage mechanism. Each supported framework's obligations are decomposed into the specific ATT&CK techniques that satisfy them. Every detection the platform raises already carries its technique identifier, attached by Detection Factory at the point of authoring. Because both use the same technique language, a firing detection is automatically attributed to every framework whose control set references that technique, continuously rather than seasonally. A relevance score indicates how directly a framework's technical obligations align to the platform's detection coverage, and a per-framework, board-ready evidence report is exported from the same underlying detections rather than assembled by hand.

That same shared language is what lets an organization operating across several jurisdictions see its actual, current coverage per region rather than a policy document's nominal list, because a new region's framework reuses a technique decomposition already in place wherever the same adversary behavior is regulated elsewhere.

THE FUNCTION IS PRECISELY SCOPED.

Regulatory Frameworks maps detections to regulatory obligations and reports the evidence. It does not attach the technique identifier itself, which is Detection Factory. It does not compute a risk score, which is the Risk Scoring Formula. And it produces evidence of specific controls firing, not a legal determination of compliance, which remains an organizational and legal judgment. Its evidence stands apart from the platform's data residency architecture, which is described in ClearSkies iISOC Data Sovereignty.

Five stages
Decompose, tag, attribute, score, report

Map once
One technique language across every supported region

Evidence, not assurance
A control that fired, not a compliance guarantee

SECTION 2

Why Manual Mapping Falls Short

Compliance evidencing is underinvested relative to the rest of the security operation, and the consequences surface as a scramble before an audit rather than as a single visible failure. Four failure modes explain why, and each is answered by a named stage of the mechanism in section 3.

2.1 The same control mapped by hand, once per framework

An organization subject to several regulations conventionally maps the same underlying control separately for each one, in separate spreadsheets, because nothing links the mappings through a shared technique language. The effort scales with the number of regulations rather than with the number of distinct controls actually in place.

2.2 Coverage claimed without a technical basis

A framework is marked addressed in a compliance tracker without a clear link to the specific detection that evidences it, so the claim cannot survive a specific auditor question about which control actually fired. The tracker records an intention rather than a demonstrated fact.

2.3 Evidence assembled once a year under deadline pressure

Mapping detections to a regulation's requirements is treated as an audit-season project rather than a continuous byproduct of running the security operation, so the evidence compiled is already stale relative to the current detection posture by the time an auditor reviews it.

2.4 A global footprint invisible per jurisdiction

An organization operating across several regions has no single, current view of which of the many frameworks it may be subject to are meaningfully covered by its actual detection coverage, as opposed to nominally listed in a policy document. A jurisdiction added after the initial compliance assessment is especially likely to be listed without being checked.

2.5 The cost of the workaround

The usual response is a compliance team or an external auditor re-deriving the same evidence from console exports and interview notes, once per framework, once per year. Cost scales with the number of regulations an organization is subject to, while the underlying problem, mappings that are not linked through a shared technique language, is untouched.

Industry research on compliance operations consistently finds audit preparation time dominated by manually re-deriving evidence framework by framework, though the specific figure for that cost, and for the share of it attributable to duplicate mapping work, is not yet sourced for this brief.

SECTION 3

How Regulatory Frameworks Works

Regulatory Frameworks is one continuously operating mechanism native to the TDIR core, not an annual mapping exercise. A regulation's text enters at one end, and continuous, per-framework evidence leaves at the other.

> *Figure 1. The five-stage evidencing mechanism.*

3.1 The five stages

Each stage is named once here and referenced by name throughout the rest of this brief, paired with the failure mode from section 2 that it answers.

STAGE	WHAT HAPPENS	THE FAILURE MODE IT ANSWERS
Decompose	A supported framework's obligations are decomposed into the specific MITRE ATT&CK techniques that represent the adversary behavior it requires an organization to defend against and evidence.	The same control mapped by hand, once per framework
Tag	Every detection the platform raises already carries its ATT&CK technique identifier, attached by Detection Factory at the point of authoring, not added afterward for compliance purposes.	Coverage claimed without a technical basis
Attribute	A firing detection is automatically attributed to every framework whose control set references its technique, continuously, without manual tagging.	Evidence assembled once a year under deadline pressure

STAGE	WHAT HAPPENS	THE FAILURE MODE IT ANSWERS
Score	Each framework carries a relevance score indicating how directly its technical monitoring obligations align to the platform's detection coverage.	Coverage claimed without a technical basis
Report	A per-framework, per-region evidence export is produced from the same underlying detections, current as of the moment it is generated.	A global footprint invisible per jurisdiction

A single detection can satisfy several frameworks at once precisely because Decompose and Tag share the same technique language. Attribute is the stage where that shared language turns into evidence for every applicable regime simultaneously.

3.2 How a mapping is judged

Quality is measured rather than assumed, on definitions a provider can report and a customer can audit. The measures below are reported natively and per tenant. They are defined here because the same metric name is measured differently by different vendors, and target values are agreed per engagement rather than asserted.

MEASURE	THE DEFINITION USED	WHAT A POOR VALUE TRIGGERS
Mapping completeness	A framework's technique-based obligations with at least one live, validated detection attached, as a share of the total.	A review of the gap against the Detection Factory coverage matrix.
Attribution latency	Elapsed time between a detection firing and its evidence being attributed to every referencing framework.	Investigation of the specific stage introducing the delay.
Citation currency	Supported frameworks with citation and version confirmed current within the defined review period, as a share of the catalog.	Prioritized legal and compliance review of the stale citation.
Frameworks evidenced per detection	The average number of distinct frameworks a single detection's evidence satisfies at once.	A review of whether the decompositions in place are too narrow to share techniques.
Time to onboard a new jurisdiction	Elapsed time from adding a framework's technique decomposition to its appearance in reporting.	A review of the decomposition backlog against the regions a tenant actually operates in.

MEASURE	THE DEFINITION USED	WHAT A POOR VALUE TRIGGERS

Mapping is not compliance. ***A framework this function evidences at a high relevance score is a technical mapping, not a legal determination. A specific detection firing is a fact the platform can produce. Whether an organization is compliant with a given regulation is a judgment the organization and its counsel make, informed by that evidence rather than replaced by it.***

SECTION 4

Regulatory Frameworks in the Platform

4.1 What it draws on, what it writes back, and what it does not do

What it draws on. The ATT&CK technique identifier Detection Factory attaches at authoring time, and the decomposition of each supported framework's legal or standards text into the technique set that satisfies it, reviewed on a defined cadence. Each is an input to a specific stage rather than a capability this function generates.

What it writes back. Per-framework evidence records and relevance scores, published into the shared model, and the board-ready reports generated from them. KPIs and SLAs may draw on the reporting cadence for its own audience-matched delivery. The relationship is one way: nothing returned to this function changes a mapping already published, and Decompose changes only when a framework's underlying text changes. An internal compliance review and an external audit draw on the same evidence records rather than on separate copies.

What it does not do. It does not attach a technique identifier to a detection, which is Detection Factory's authoring stage. It does not compute a risk score or decide a response. And it does not determine or assure legal compliance, only that a specific, technique-mapped control fired. Residency is a platform-level architecture property: this function evidences framework alignment whatever configuration is in place.

4.2 A worked example

The chain below is illustrative and technically representative rather than drawn from a named engagement. A detection fires for credential abuse consistent with valid-account techniques on an internet-facing system. Tag confirms the detection already carries its technique identifiers, attached when Detection Factory authored the rule. Attribute recognizes that both techniques are referenced by the control sets of three supported frameworks operating in the tenant's region, a data-protection regulation, a financial operational-resilience regime and an international security standard, and links the single detection to all three. Score reflects a highly relevant control area for two of the three and a

supporting one for the third. Report includes the detection, its timestamp and its technique mapping in the next scheduled export for each framework, and the same detection remains available to any framework added later that references either technique.

4.3 Delivery across many tenants

A framework's technique decomposition is maintained once and applies to every tenant subject to it, so onboarding a customer into a new jurisdiction means applying an existing decomposition rather than building a new mapping. What travels between tenants is that decomposition; what never travels is one tenant's detections, evidence records or reports.

Because the function is switched on with the platform rather than deployed separately, time to value is a matter of enabling the relevant frameworks rather than an onboarding project. In the first week, a tenant's existing detections are attributed against every decomposition already enabled for its region, so a first evidence export is available immediately. Through the first month, mapping completeness becomes visible against each enabled framework's obligations, surfacing any gap for Detection Factory to prioritize. At steady state, citation currency and attribution latency are tracked as ongoing measures.

SECTION 5

MITRE ATT&CK as the Shared Technique Language

MITRE ATT&CK is what makes cross-framework evidencing possible. Every supported framework is expressed as the set of techniques that represent the adversary behavior it requires an organization to defend against, and every detection already carries the technique identifier Detection Factory attached. Because the same language underlies both sides, mapping is a lookup rather than a translation exercise performed by hand.

Mapping is not coverage. A technique referenced in a framework's control set and matched to a live detection is claimed evidence. Whether that specific detection has been exercised in a purple-team run or validated against MITRE Engenuity ATT&CK Evaluations is the claimed-versus-demonstrated distinction the Detection Factory brief owns. This function inherits that distinction and states which of the two applies, rather than presenting every mapped technique as equally proven.

A representative set of supported frameworks appears below rather than the complete catalog, which is maintained live and available through platform reporting.

REGION	REPRESENTATIVE FRAMEWORKS	WHAT THEY REQUIRE DETECTION COVERAGE FOR
European Union and Europe	GDPR (EU 2016/679, Art. 32), NIS2 (EU 2022/2555), DORA (EU 2022/2554)	Security of processing, network and information security, and financial operational resilience, spanning both a general data-protection regulation and two sector-specific regimes.
Middle East	SAMA cybersecurity framework, UAE NESA, Qatar National Information Assurance	Financial-sector and national-assurance security monitoring obligations, each issued by a national regulator rather than a shared regional body.
International standards	ISO/IEC 27001 and 27002, PCI DSS, CMMC 2.0	Information security management, payment-card data protection and defense-sector maturity, none of them tied to a single jurisdiction.
North America	HIPAA Security Rule, NIST Cybersecurity Framework 2.0	Health-data safeguards and general cybersecurity outcome reporting, the latter widely referenced outside North America as well.

Relevance scores are illustrative of the model and dependent on the methodology behind them, so a specific figure is confirmed before use with a customer and reviewed alongside the citation currency check in section 3.2. A framework still in draft or pending is marked as such rather than presented as settled law, and adding one to the catalog reuses an existing decomposition wherever the same behavior is already regulated elsewhere.

SECTION 6

Questions Raised in Evaluation

The questions below recur in buyer evaluations, each answered directly against the mechanism described in section 3.

QUESTION, AS A BUYER ASKS IT	ANSWER
Does mapping to a framework mean the organization is compliant with it?	No. It produces evidence that specific technique-mapped controls fired. A compliance determination is a legal and organizational judgment this function does not make.

QUESTION, AS A BUYER ASKS IT	ANSWER
How current are the framework citations?	Reviewed on a defined cadence, and citation currency is one of the measures in section 3.2. A framework still in draft or pending is marked as such rather than presented as settled law.
What does the relevance score mean?	It indicates how directly detection coverage aligns to a framework's technical monitoring obligations. It is illustrative of the model rather than a certified audit score.
Does this function decide which detections fire?	No. Detection Factory authors and tags detections. This function consumes the technique identifier already attached and never adds one of its own.
Does compliance evidencing depend on where data is stored?	No. Data residency and sovereignty are platform-level architecture properties held by ClearSkies iISOC Data Sovereignty. This function evidences framework alignment whatever residency configuration is in place.
Does reporting depth vary by what a customer pays for?	Commercial packaging is a decision made elsewhere. This function evidences whatever frameworks are enabled for a tenant, at the same depth in every case.
What happens when a regulation is amended?	Decompose is rerun against the amended text, and only the affected technique set changes. Detections already attributed under the unchanged obligations are unaffected.

SECTION 7

The Benefits

Compliance evidencing is where a security operation either produces the record or asks to be taken at its word, and the outcomes are countable: how many regimes one detection evidences, how much of a framework's obligations have a live detection behind them, and whether the evidence an auditor reads is current or reconstructed.

7.1 At a glance

BENEFIT	WHAT PRODUCES IT
One detection evidences every regime that references it	Decompose and Tag share the same technique language, so Attribute links a firing detection to every framework at once.

BENEFIT	WHAT PRODUCES IT
A coverage claim has a technical basis behind it	Every attributed detection carries the technique identifier Detection Factory attached at authoring.
Evidence is current rather than reconstructed	Attribute runs continuously as detections fire, so Report exports what is true at the moment it is generated.
A gap is visible before an auditor finds it	Mapping completeness, defined in section 3.2, measures obligations with no live detection attached.
A new jurisdiction costs a decomposition, not a project	An existing technique decomposition is reused wherever the same adversary behavior is regulated elsewhere.
The limit of the claim is stated, not implied	Evidence of a control firing is distinguished from a compliance determination throughout, in sections 3.2 and 5.

7.2 What changes, and for whom

A continuous, technique-linked evidence base changes what each role spends time on, and what each role can defend under audit.

ROLE	WHAT CHANGES
Compliance analyst	Works from a live, technique-linked evidence report rather than reconstructing one from console exports before an audit.
SOC manager	Mapping completeness and citation currency become visible, manageable numbers rather than an assumption checked once a year.
CISO and risk	A per-framework, board-ready report replaces a compliance tracker's checkbox, with a technical basis behind every claim.
Service provider practice lead	A customer entering a new jurisdiction is mapped once against an existing technique decomposition rather than requiring bespoke work per regulation.

7.3 What sets it apart

▸ **Mapped once, evidenced everywhere.** A single detection's technique attribution satisfies every framework that references it, so the cost of adding a regime is a decomposition exercise rather than a parallel mapping project.

▸ **Coverage measured, not asserted.** A relevance score and a mapping-completeness figure replace a tracker's checkbox, so a gap is visible before an auditor finds it rather than after.

▸ **Continuous, not seasonal.** Evidence accumulates as detections fire rather than being assembled under deadline pressure, so a report is never older than the detection posture it describes.

> **THE VALUE IN ONE LINE.**
>
> Manual mapping produces a document that says a control exists. A shared technique language produces the detection that fired, the technique it carried and every regime that technique answers, from the same record, on the day the auditor asks.

SECTION 8

Summary

Regulatory Frameworks is how the platform turns its own detections into regulatory evidence. It is one core function among twelve, scoped to a five-stage mechanism that runs continuously inside the TDIR core rather than as an annual mapping project performed by hand.

▸ Decompose, tag, attribute, score and report replace framework-by-framework manual mapping.

▸ The technique identifier this function consumes is attached by Detection Factory, which authors and tags every detection.

▸ A single detection is attributed to every framework whose control set references its technique, continuously rather than seasonally.

▸ A mapped, firing control is a fact the platform produces. Whether that satisfies a given regulation stays a judgment the organization and its counsel make.

▸ The function maps and reports, and leaves detection authoring, scoring, response and commercial structure to the functions and decisions built for them.

Forward-looking statement: any statement about future capability is forward-looking and non-binding, and capability described in the present tense is capability available at the publication date, subject to the deployment configuration in force. This document describes evidencing and reporting capability and does not constitute legal advice on any customer's regulatory obligations.

ClearSkies™

CORE FUNCTION BRIEF

Third-Party Add-ons Marketplace

A core function of the ClearSkies iISOC platform

A governed catalog of third-party connectors, monitored live and flagged before a feed goes dark.

Contents

SECTION 1

Executive Summary

IN ONE SENTENCE.

The Marketplace is the core function that catalogs, connects, monitors and flags the health of every third-party integration, so visibility grows through a governed, repeatable framework rather than one bespoke connector at a time.

A detection is only as good as the data behind it. Bringing a new tool into a security operation conventionally means bespoke engineering, an ad hoc script or a professional-services project for each source, so onboarding takes weeks and produces something only the person who built it can maintain. A feed can degrade or a credential can expire silently, and nobody notices until an incident review asks why a source went dark. And a wide catalog of connected-but-unused sources is integration sprawl, while a narrow, deep view misses whole categories of attacker path.

The Marketplace answers this with a five-stage mechanism. A catalog of more than 600 integrations is organized across nine categories, so an architect can see what is connectable and what it would add before committing engineering time. A connection is provisioned as a governed link, standardized to a common schema, authenticated with least-privilege credentials, and versioned, replacing one-off integration work. Every active connector carries a live health signal, and a failure-risk model flags a degrading connector before it fails rather than after.

Breadth and currency are always assessed as a pair, because a view that is wide but stale, or current but narrow, is still a blind spot. A Resource Center of guides, runbooks and reference material lets an administrator configure and troubleshoot a connector without depending on whoever originally set it up.

THE FUNCTION IS PRECISELY SCOPED.

The Marketplace catalogs, connects and monitors third-party integrations. It does not decide how urgent a security alert is, which is the Risk Scoring Formula. It does not execute a remediation such as renewing a credential or throttling traffic, which is SOAR or an administrator acting on a published flag. And it defines no commercial integration tier or price. Each of those is a separate decision or a separate core function.

Five stages
Discover, connect, monitor, predict, publish

600+ integrations
Cataloged across nine categories

Coverage and freshness
Assessed as a pair, never as a single number

Representative of the function's design. Which connectors are enabled, the remediation configured for a flag and the report cadence are set per tenant.

SECTION 2

Why Ad Hoc Integration Falls Short

Integration is underinvested relative to the rest of the security operation, and the consequences surface as a blind spot rather than as a single visible failure. Four failure modes explain why, and each is answered by a named stage of the mechanism in section 3.

2.1 Integration built one connector at a time

Each new source conventionally means bespoke engineering, an ad hoc script or a professional-services project, so bringing on a new tool takes weeks and produces something only its builder can maintain.

2.2 Outages discovered after the blind spot opens

A feed can degrade in volume or freshness, or a credential can silently expire, and nobody notices until an incident review asks why a source had no data for days.

2.3 Breadth without depth, or depth without breadth

A wide catalog of connected-but-unused sources is integration sprawl, cost and maintenance without security return, while a narrow, deep view of only a few tool categories leaves whole categories of attacker path unmonitored.

2.4 A flagged risk with nowhere to go

A connector approaching failure is only useful information if it reaches someone who can act on it. A risk noticed but not routed to an owner or a runbook is no better than no warning at all.

2.5 The cost of the workaround

The usual response is a dedicated integration engineer or a standing professional-services engagement whose job is to build and firefight connectors one at a time. Cost scales with the number of sources and the pace of vendor API changes, while the underlying problem, connections that are not standardized,

monitored or predicted, is untouched. The workaround also caps how far visibility can grow: adding a source means adding integration capacity in step, so the breadth of what a security operation can see is bounded by how fast it can build and maintain connectors rather than by what would be useful to watch.

Industry research on security operations consistently ties tool sprawl and integration complexity to operational cost, and separately finds a meaningful share of connected sources contributing nothing to detection, though the specific figures are not yet sourced for this brief.

SECTION 3

How the Marketplace Works

The Marketplace is one continuously operating mechanism native to the TDIR core, not a drawer of one-off integrations. A vendor API or telemetry source enters at one end, and a governed, monitored, self-documenting connection leaves at the other.

> *Figure 1. The five-stage catalog and connector mechanism.*

3.1 The five stages

Each stage is named once here and referenced by name throughout the rest of this brief, paired with the failure mode from section 2 that it answers.

STAGE	WHAT HAPPENS	THE FAILURE MODE IT ANSWERS
Discover	The catalog is organized by category, network and firewall, endpoint and EDR, cloud and SaaS, identity, email, threat intelligence, posture, ticketing and notification, so an architect sees what is connectable before committing engineering time. That structure is what makes a gap visible as a gap rather than as an absence nobody thought to look for.	Breadth without depth, or depth without breadth
Connect	A connector is provisioned as a governed link: standardized to a common schema, authenticated with least-privilege credentials and key rotation, and versioned, replacing bespoke integration engineering.	Integration built one connector at a time
Monitor	Every active connector carries a live health signal, continuously comparing expected against actual data flow and volume.	Outages discovered after the blind spot opens

STAGE	WHAT HAPPENS	THE FAILURE MODE IT ANSWERS
Predict	A model scores each active connector for failure risk from flow metadata, credential state, capacity headroom and recent change context, flagging a connector before it degrades rather than after.	Outages discovered after the blind spot opens
Publish	Health, coverage, freshness and any at-risk flag are published into the shared model. This function does not itself renew a credential, throttle traffic or reconfigure a connector.	A flagged risk with nowhere to go

Predict hands off at the flag. What happens next, an automated renewal, a throttled poll or a routed troubleshooting task, is a decision SOAR or an administrator makes using the flag the Marketplace published.

3.2 How a connector is judged

Quality is measured rather than assumed, on definitions a provider can report and a customer can audit. The measures below are reported natively and per tenant, defined here because the same metric name is measured differently by different vendors. Target values are agreed per engagement rather than asserted.

MEASURE	THE DEFINITION USED	WHAT A POOR VALUE TRIGGERS
Coverage	Relevant tools and assets actively connected and feeding the platform, as a share of the in-scope catalog.	A prioritized onboarding review for the uncovered category.
Freshness	Average latency between an event occurring in a source and its normalized arrival.	A review of transport, moving a source from scheduled pull to streaming push.
Adoption rate	Connected sources actively contributing to detections, enrichment or response over a period, as a share of all connected sources.	A review of whether an unused source should be retired rather than maintained.
Prediction accuracy	Failure-risk predictions confirmed correct against the actual outcome, as a share of all predictions.	A retraining review of the model against recent misses.
Prediction lead time	Elapsed time between an at-risk flag and the connector's actual failure.	A review of whether the flag threshold fires early enough to be acted on.

MEASURE	THE DEFINITION USED	WHAT A POOR VALUE TRIGGERS

Wide but stale is still a blind spot. ***Coverage and freshness are never read alone. A source that is connected but arrives with significant latency gives a false sense of visibility, and a current picture of two categories says nothing about the seven it does not cover.***

SECTION 4

The Marketplace in the Platform

4.1 What it draws on, what it writes back, and what it does not do

What it draws on. The third-party vendor's API or telemetry stream as the external source; the normalized entity model the Centric-AI Fabric carries, into which a connector's data is standardized; and each tenant's catalog of enabled integrations. Each is an input to a specific stage rather than a capability this function generates.

What it writes back. A connector's health, coverage and freshness status, and any at-risk flag, published into the shared model. SOAR consumes a flag where an automated remediation exists; otherwise an administrator acts on it through the Resource Center. The telemetry a connector delivers flows through the Centric-AI Fabric like any other source: this function's own contribution is the status of the connection, not the content it carries.

What it does not do. It does not decide how urgent a security alert is or who investigates it, and it does not renew a credential, throttle traffic or reconfigure a connector on its own authority. SOAR or an administrator makes that decision from the flag it publishes. It defines no commercial integration tier or price.

> **THE ONE DOCUMENTED EXCEPTION TO ONE-WAY REPORTING.**
>
> Where a third-party vendor exposes a response API, the connector the Marketplace governs can accept an instruction back from SOAR as well as send telemetry forward, unlike a native add-on's strict one-way relationship. The capacity belongs to the connector; the decision to use it belongs to SOAR.

4.2 A worked example

The chain below is illustrative and technically representative rather than drawn from a named engagement. A cloud connector approaches its API rate limit during a traffic spike, and its access token is due to expire within days. Monitor shows volume nearing the expected ceiling, Predict combines the rate-limit headroom, the token's age and the recent traffic pattern into a high failure-likelihood score,

and Publish surfaces the flag. Because this tenant has an automated renewal remediation configured, SOAR reads the flag and invokes it, throttling a lower-priority polling job to protect headroom. Without that remediation, the same flag routes to an administrator through the Resource Center with the guide for that connector attached. Both branches begin with the same flag, and the function's work ends there either way.

4.3 Delivery across many tenants

The catalog, the connector framework and the Resource Center serve every tenant from the same governed structure, so onboarding a source is repeatable rather than a bespoke build per customer, and a vendor API change is absorbed once in the connector version rather than separately in every tenant using that source. What travels between tenants is the catalog and the framework; what never travels is one tenant's credentials, telemetry or connector configuration.

Because the function is switched on with the platform rather than deployed separately, time to value is a matter of connecting sources rather than an onboarding project. In the first week, the catalog is available for review and the first connectors in a tenant's priority categories begin feeding the platform. Through the first month, coverage becomes visible and Predict accumulates enough flow history to be meaningful for that tenant's own sources. At steady state, prediction accuracy and adoption rate are tracked as ongoing measures.

SECTION 5

Schema and Ingestion Framework Alignment

The Marketplace does not map detection content and does not compute technique relevance, so MITRE ATT&CK plays no direct role for this subject and is not given a row below. The frameworks that matter here are the ones governing how a third-party source is standardized and ingested, which is what the Connect stage does.

FRAMEWORK OR STANDARD	ITS ROLE IN THE MARKETPLACE
OCSF	The common schema every connector standardizes to at the Connect stage. The platform's normalized model aligns to it rather than being it, which is a distinction an architect will ask about.
Syslog, CEF, LEEF and REST API	General collection formats supported across the catalog, together with the position on custom and AI-assisted parsers.
OpenTelemetry	Application and infrastructure telemetry ingestion, for the subset of the catalog built on it.

FRAMEWORK OR STANDARD	ITS ROLE IN THE MARKETPLACE
STIX and TAXII	The connector format for the threat-intelligence category of the catalog. What happens to the content afterward, its curation and rating, belongs to the Threat Intelligence brief.

Support for a format is not compliance with an obligation. The platform states which schemas and transports a connector supports. It never states that supporting one makes an organization compliant with anything, and control-level framework mapping is held by the Regulatory Frameworks core function.

SECTION 6

Questions Raised in Evaluation

The questions below recur in buyer evaluations, each answered directly against the mechanism described in section 3.

QUESTION, AS A BUYER ASKS IT	ANSWER
Does the Marketplace decide how urgent a degraded connector is, or route it to an analyst?	No. Connector health carries its own status model. An at-risk flag is consumed by SOAR where an automated remediation exists, or by an administrator through the Resource Center.
Does the Marketplace execute remediation itself, such as renewing a credential or throttling traffic?	No. It publishes the flag. SOAR or an administrator decides and acts.
Can a third-party integration receive instructions from the platform, or only send data?	Both are possible, depending on the connector. Sending telemetry is universal, and where a vendor exposes a response API the connector can also accept an instruction back from SOAR. This is the one documented exception to the platform's native add-on architecture.
Does the Marketplace define what a customer pays for different levels of integration?	No. Catalog depth available at a given platform tier is a commercial decision made elsewhere. This function governs and measures whatever connectors are enabled.

QUESTION, AS A BUYER ASKS IT	ANSWER
How is a connected-but-unused source identified?	Adoption rate, defined in section 3.2, tracks the share of connected sources actively contributing to detections, enrichment or response, surfacing candidates for review or retirement.
What happens when a vendor changes its API?	The change is absorbed once in the connector version rather than separately in every tenant that uses that source, which is what the Connect stage's versioning exists for.
Can a tenant configure its own connectors, or must a provider do it?	The Resource Center's configuration guides support self-service setup. The same governed connector framework applies whether a tenant or a provider configures it.

SECTION 7

The Benefits

Integration is where a security operation either grows what it can see or spends its engineering budget keeping what it already has alive. The outcomes are countable: how much of the in-scope catalog is connected and current, how many degrading connectors were caught before a feed went dark, and how much of that took bespoke work.

7.1 At a glance

BENEFIT	WHAT PRODUCES IT
A new source is provisioned rather than engineered	Connect standardizes, authenticates and versions the link, replacing one-off integration work.
A blind spot is prevented rather than explained afterward	Predict scores a degrading connector and Publish surfaces the flag while there is still time to act.
A gap is visible as a gap	Discover organizes the catalog into nine categories, so an unmonitored category is a named absence.
A wide view cannot pass for a current one	Coverage and freshness, defined in section 3.2, are read as a pair and never as a single number.

BENEFIT	WHAT PRODUCES IT
A connector outlives the person who built it	The Resource Center carries the guides and runbooks that a bespoke integration kept in one engineer's head.
Visibility scales with the catalog, not with headcount	The same catalog and connector framework serve every tenant, and a vendor API change is absorbed once.

7.2 What changes, and for whom

A governed, monitored connector framework changes what each role spends time on, and what each role can rely on being current.

ROLE	WHAT CHANGES
Security architect	Sees what is connectable and what it would add before committing engineering time, rather than scoping a bespoke project to find out.
SOC analyst	Works from a live, current picture rather than discovering a blind spot in the middle of an incident.
Administrator	Self-serves setup and troubleshooting from the Resource Center rather than depending on whoever originally configured a connector.
Service provider practice lead	Repeatable onboarding and one console scale across every tenant, without bespoke integration engineering per customer.

7.3 What sets it apart

▸ **Governed, not bespoke.** Every connector is standardized, authenticated and versioned, so integration becomes a provisioning step rather than an engineering project with a single owner.

▸ **Failure flagged before it happens.** Predict surfaces a degrading connector while there is still time to fix it, so a blind spot is prevented rather than explained afterward in an incident review.

▸ **Breadth and depth measured together.** The two measures are read as a pair, so a wide-but-stale or narrow-but-current view is never mistaken for good visibility.

▸ **Self-service by design.** The Resource Center gives an administrator what a bespoke integration would otherwise need professional services for, so knowledge of a connector does not leave with the person who configured it.

The value in one line. Ad hoc integration buys visibility one connector at a time and pays for it forever. A governed catalog turns the same visibility into something that is provisioned, measured, and warned about before it fails.

SECTION 8

Summary

The Marketplace is how the platform turns a heterogeneous, third-party technology stack into governed, monitored visibility. It is one core function among twelve, scoped to a five-stage mechanism that runs continuously inside the TDIR core rather than as a drawer of bespoke, one-off connectors.

▸ Discover, connect, monitor, predict and publish replace ad hoc, one-connector-at-a-time integration.

▸ Every connector is standardized, authenticated and versioned, and its health is tracked live.

▸ Failure risk is flagged before a feed goes dark, and the function publishes the flag rather than acting on it unilaterally.

▸ Breadth and currency are assessed together, and a Resource Center supports self-service configuration and troubleshooting.

▸ The function catalogs, connects and monitors third-party integrations, and leaves security urgency, response and commercial structure to the functions and decisions built for them.

Forward-looking statement: any statement about future capability is forward-looking and non-binding, and capability described in the present tense is capability available at the publication date, subject to the deployment configuration in force. Catalog scale and parser coverage change as vendors and formats change.