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.