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 A sequence that fails or drifts out of date is flagged for revision rather than left to fail again during the next incident.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.