SOC LEVEL 1 ESSENTIALS – PART G: AUTOMATION & SCRIPTING (POSTS 51–56)

Practical Automation for SOC Analysts: Python, Regex, CTI APIs, SOAR, Playbooks & Configs

SEPTEMBER 29, 2025

INAAM KABBARA
SECURITY ANALYST | CYBERSECURITY CONTENT CREATOR | EPITA MSC GRADUATE

Table of Contents

1- Why Python matters in a SOC

🧠 Automate repetitive tasks with simple scripts
💡 Speed up log analysis & CTI enrichment
❓ Have you ever built a script to save time in investigations?

🛑 What is Python's role in a SOC?
Python is one of the most popular languages in cybersecurity.
It's simple, powerful, and offers thousands of libraries that help analysts automate detection and response.

Common uses:
✓ Parsing and normalizing logs
✓ Automating CTI lookups (VirusTotal, AbuseIPDB)
✓ Repetitive triage actions
✓ Quick custom detection scripts

🔐 SOC Relevance
A Level 1 analyst handles huge alert volumes. Python can:
✓ Extract useful data from logs faster
✓ Enrich IOCs automatically with APIs
✓ Normalize alerts for faster triage

🔎 Practical Examples
✅ Extract IPs and usernames from SIEM logs
✅ Regex scripts to spot suspicious patterns
✅ Query CTI feeds and add context
✅ Simple anomaly detection in traffic captures

🧰 Helpful Tools
Built-in: os, re, json
pandas → log parsing
requests → APIs
scapy → network analysis
elasticsearch-py → SIEM queries

💡 Pro Tip
Start small: automate one task you repeat daily.
Even saving 5 minutes per task adds up to hours gained per week.

💬 Ask yourself:
🔎 "What SOC task can I automate today?"

Quel est le rôle de Python dans un SOC ?

Python est l'un des langages les plus utilisés en cybersécurité.
Simple et puissant, il permet d'automatiser la détection et la réponse.

Usages courants :
✓ Analyse et normalisation des logs
✓ Enrichissement CTI (VirusTotal, AbuseIPDB)
✓ Automatisation des tâches répétitives
✓ Scripts de détection personnalisés

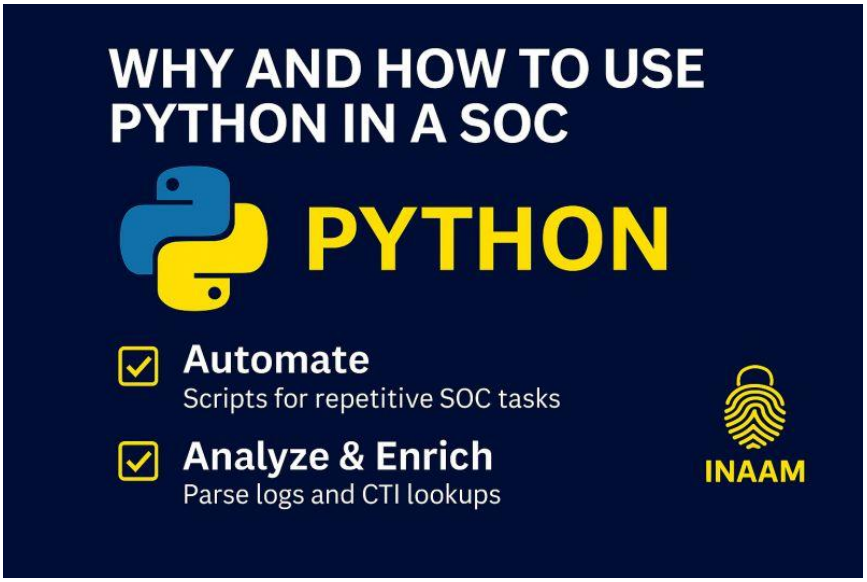
🔐 Utilité en SOC
Un analyste N1 gère un volume massif d'alertes. Python aide à :
✓ Extraire rapidement les données utiles
✓ Enrichir automatiquement les IOCs
✓ Normaliser les formats d'alertes

🔎 Exemples pratiques
✅ Extraction d'IPs et noms d'utilisateur dans les logs SIEM
✅ Regex pour détecter des motifs suspects
✅ Requêtes CTI automatiques
✅ Scripts simples pour anomalies réseau

🧰 Outils utiles
Standard : os, re, json
pandas pour logs
requests pour APIs
scapy pour trafic réseau
elasticsearch-py pour SIEM

💡 Astuce
Commencez petit : automatisez une tâche que vous répétez chaque jour.
Quelques minutes gagnées par tâche = des heures libérées par semaine.

💬 Posez-vous la question :
🔎 « Quelle tâche SOC puis-je automatiser aujourd'hui ? »


WHY AND HOW TO USE
PYTHON IN A SOC
PYTHON
Automate
Scripts for repetitive SOC tasks
Analyze & Enrich
Parse logs and CTI lookups
INAAM

2- Regex for SOC log parsing

🧠 Spot hidden patterns in logs automatically
💡 Extract IPs, usernames & error codes in seconds
❓ Have you ever used regex to filter massive log files?

🛑 What is Regex in a SOC?
Regex (regular expressions) is a pattern-matching language.
It lets analysts search, extract, and transform text in logs—ideal for quick triage.

Common uses:
✓ Extract IPs from SIEM logs
✓ Detect failed login attempts
✓ Pull usernames from auth logs
✓ Filter error codes in apps

🔐 SOC Relevance
SOC L1 analysts deal with millions of log entries. Regex helps by:
✓ Saving hours of manual searching
✓ Extracting IOCs (IPs, domains) fast
✓ Normalizing data before enrichment

🔎 Practical Examples
✅ Extract IPs → \b\d{1,3}(\.\d{1,3}){3}\b
✅ Detect “Failed password” → Failed password for (\w+)
✅ Match suspicious domains → [a-z0-9.-]+\.(ru|cn|xyz)

🧰 Helpful Tools
Python re → regex functions
pandas → CSV log parsing
Use in scripts before SIEM ingestion

💡 Pro Tip
Always test regex on sample logs.
A tiny mistake can create false positives—or miss threats.

💬 Ask yourself:
🔎 “Which SOC task could I automate with regex?”

Qu’est-ce que Regex dans un SOC ?

Regex (expressions régulières) est un langage de recherche de motifs.
Il permet d'extraire et transformer du texte dans les journaux—parfait pour le tri rapide.

Cas d'usage :
✓ Extraire des IPs dans SIEM
✓ Détecter connexions échouées
✓ Récupérer des noms d'utilisateur
✓ Filtrer des codes d'erreur

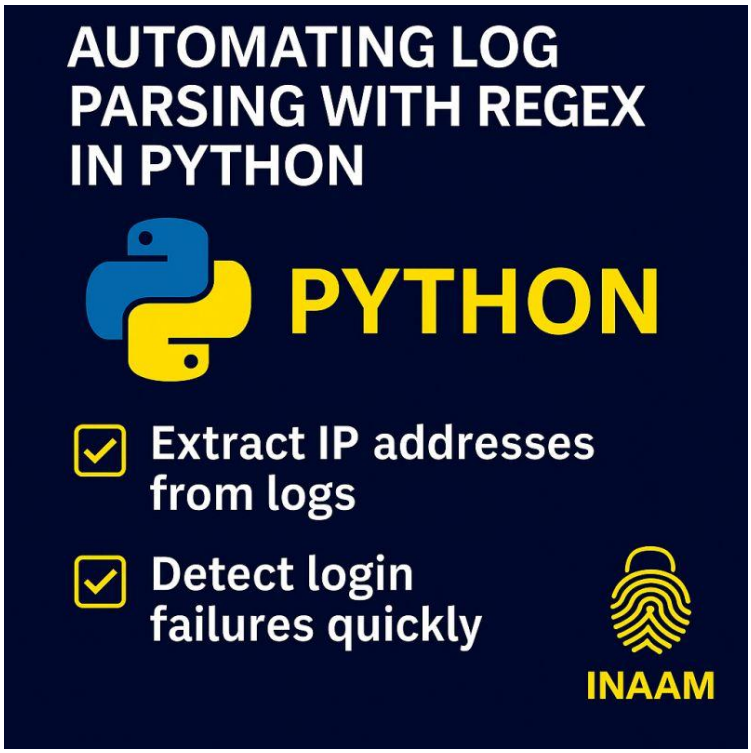
🔐 Utilité en SOC
Un analyste N1 gère des millions de logs. Regex permet :
✓ Gain de temps énorme
✓ Extraction rapide des IOCs
✓ Normalisation avant enrichissement

🔎 Exemples pratiques
✅ IPs → \b\d{1,3}(\.\d{1,3}){3}\b
✅ "Failed password" → Failed password for (\w+)
✅ Domaines suspects → [a-z0-9.-]+\.(ru|cn|xyz)

🧰 Outils utiles
Python re
pandas pour CSV
Scripts avant ingestion SIEM

💡 Astuce
Toujours tester vos regex sur échantillons.
Un caractère mal placé = faux positifs ou menaces ratées.

💬 Posez-vous la question :
🔎 « Quelle tâche SOC puis-je automatiser avec regex ? »


AUTOMATING LOG PARSING WITH REGEX IN PYTHON
PYTHON
Extract IP addresses from logs
Detect login failures quickly
INAAM

3- Enriching alerts with APIs

🧠 Add context from external CTI feeds
💡 Use VirusTotal, AbuseIPDB & OTX for IOCs
❓ Have you ever used APIs to speed up investigations?

🛑 What is Threat Enrichment via API?
Threat enrichment means adding intelligence to raw alerts.
Instead of just seeing an IP or hash, analysts query APIs like VirusTotal to check if it's malicious—saving time and improving accuracy.

Common uses:
✓ IP/domain reputation checks
✓ File hash lookups (malware detection)
✓ URL analysis for phishing
✓ Cross-referencing IOCs with CTI platforms

🔐 SOC Relevance
SOC L1 analysts often face bare alerts (e.g., "Suspicious IP detected"). APIs help by:
✓ Quickly classifying IOCs as safe or malicious
✓ Reducing false positives
✓ Adding external context for better triage

🔎 Practical Examples
✅ Query VirusTotal for a file hash → returns detection ratio
✅ Check AbuseIPDB for reported malicious IPs
✅ Use OTX pulses to identify if an IOC is part of known campaigns

🧰 Helpful Tools
requests library in Python for API calls
VirusTotal API (free & paid tiers)
AbuseIPDB, OTX, Hybrid Analysis
Integration with SIEM or SOAR for automation

💡 Pro Tip
Always respect API rate limits—use caching or local databases to avoid hitting daily quotas.

💬 Ask yourself:
🔎 "Which CTI APIs could enrich my daily SOC alerts?"

Qu’est-ce que l’enrichissement via API ?

L’enrichissement des menaces consiste à ajouter du renseignement aux alertes.
Au lieu de voir seulement une IP ou un hash, l’analyste interroge une API (ex. VirusTotal) pour vérifier si l’IOC est malveillant.

Cas d’usage :
✓ Vérifier la réputation d’IPs ou domaines
✓ Analyser des hachages de fichiers (malware)
✓ Contrôler des URLs suspectes (phishing)
✓ Croiser les IOCs avec des plateformes CTI

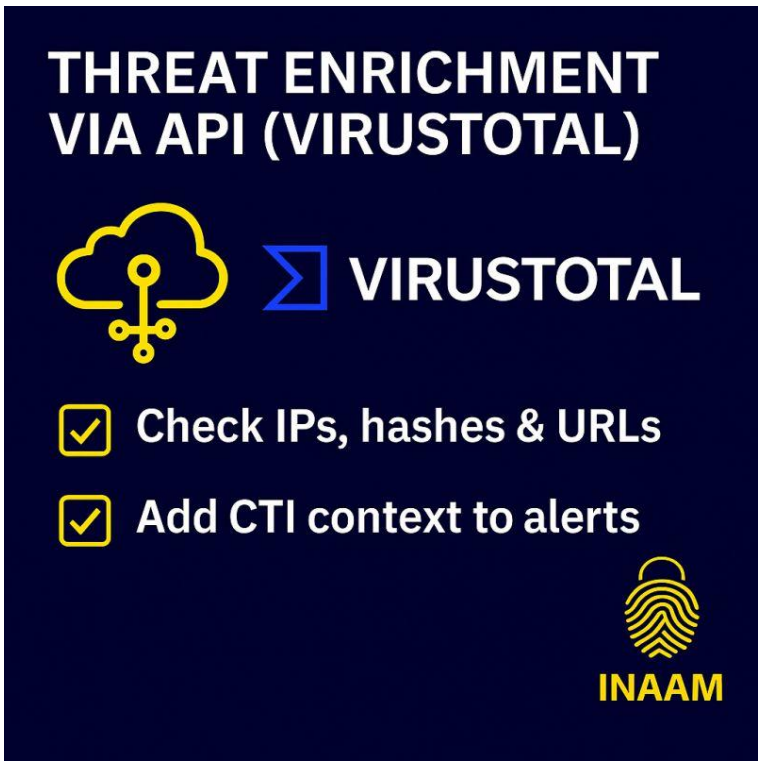
🔐 Utilité en SOC
Un analyste N1 reçoit souvent des alertes brutes. Les APIs permettent de :
✓ Classer rapidement les IOCs comme sûrs ou malveillants
✓ Réduire les faux positifs
✓ Ajouter du contexte externe pour mieux prioriser

🔎 Exemples pratiques
✅ VirusTotal pour un hash → score de détection
✅ AbuseIPDB pour vérifier une IP
✅ Pulses OTX pour identifier des campagnes connues

🧰 Outils utiles
Librairie Python requests
API VirusTotal, AbuseIPDB, OTX
Hybrid Analysis
Intégration SIEM / SOAR

💡 Astuce
Respectez toujours les limites d’API : utilisez du caching ou une base locale pour éviter les quotas.

💬 Posez-vous la question :
🔎 « Quelles APIs CTI pourraient enrichir mes alertes SOC ? »


THREAT ENRICHMENT
VIA API (VIRUSTOTAL)
VIRUSTOTAL
Check IPs, hashes & URLs
Add CTI context to alerts
INAAM

4- What is SOAR in a SOC?

🧠 Automate & orchestrate incident response
💡 Cortex, XSOAR & TheHive streamline workflows
❓ Have you used a SOAR platform before?

🛑 What is SOAR?
SOAR = Security Orchestration, Automation, and Response.
These platforms automate repetitive tasks, link tools, and standardize incident handling.

Key uses:
✓ Enrich IOCs (IPs, hashes, domains)
✓ Orchestrate actions across SIEM, EDR, firewalls
✓ Use playbooks for frequent incident types
✓ Manage cases in one place

🔐 SOC Relevance
For L1/L2 analysts, SOAR:
✓ Cuts down manual repetitive work
✓ Brings consistency to investigations
✓ Adds context for faster escalation
✓ Lets analysts focus on real analysis

🔎 Examples

✅ Cortex (TheHive Project) → open-source with analyzers
✅ Cortex XSOAR (Palo Alto Networks) → enterprise-grade automation with broad integrations
✅ TheHive → open-source case management & collaborative investigations

🧰 Helpful Tools
Cortex analyzers (VirusTotal, AbuseIPDB, etc.)
TheHive for investigations
XSOAR for automation & integrations

💡 Pro Tip
Start small—automate one repetitive task (like IP enrichment) before moving to full playbooks.

💬 Ask yourself:
🔎 “Which SOC task could I automate with a SOAR today?”

Qu’est-ce qu’un SOAR ?

SOAR = Security Orchestration, Automation and Response.
Il permet d’automatiser les tâches répétitives, d’orchestrer les outils et de standardiser la réponse aux incidents.

Usages clés :
✓ Enrichir des IOCs (IP, hash, domaine)
✓ Orchestrer des actions (SIEM, EDR, firewall)
✓ Playbooks pour incidents récurrents
✓ Gestion centralisée des cas

🔐 Utilité en SOC
Pour un analyste N1/N2, un SOAR :
✓ Réduit les tâches manuelles
✓ Améliore la cohérence des enquêtes
✓ Ajoute du contexte pour accélérer l’escalade
✓ Libère du temps pour l’analyse

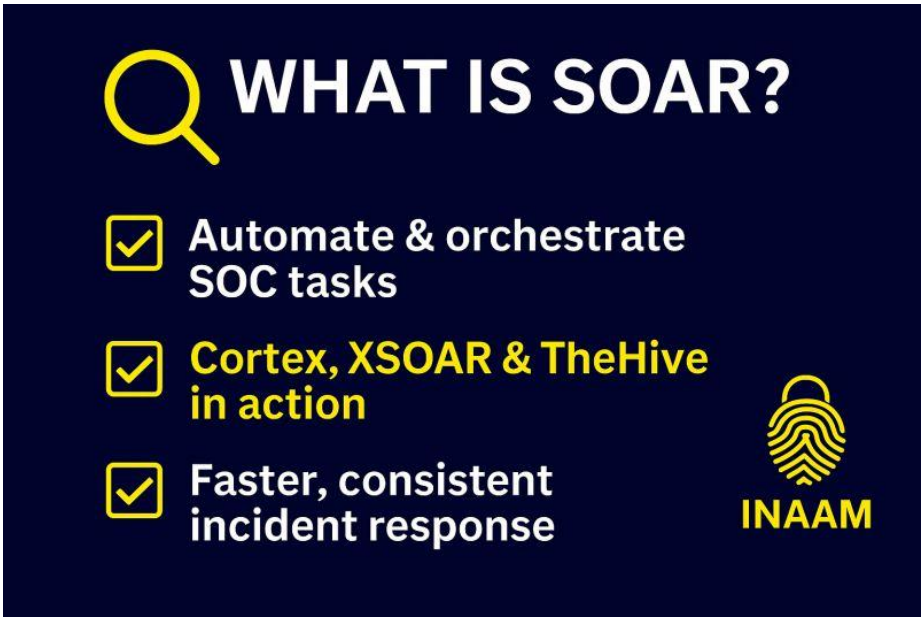
🎾 Exemples

✅ Cortex (TheHive Project) → open-source avec analyzers
✅ Cortex XSOAR (Palo Alto Networks) → solution entreprise avec nombreuses intégrations
✅ TheHive → gestion des incidents open-source et collaborative

🧰 Outils utiles
Cortex analyzers (VirusTotal, AbuseIPDB…)
TheHive pour enquêtes
XSOAR pour automatisation avancée

💡 Astuce
Commencez par automatiser une seule tâche (ex. enrichissement IP) avant de créer des playbooks complexes.

💬 Posez-vous la question :
🎾 « Quelle tâche SOC puis-je automatiser avec un SOAR ? »


WHAT IS SOAR?
Automate & orchestrate SOC tasks
Cortex, XSOAR & TheHive in action
Faster, consistent incident response
INAAM

5- What is a phishing playbook?

🧠 Standardized steps for incident response
💡 Helps SOC teams triage, contain & report
❓ Do you already use playbooks in your SOC work?

🛑 What is a SOC playbook?
A playbook is a step-by-step workflow for a specific incident type.
It ensures analysts respond quickly and consistently.
For phishing, it covers detection, verification, containment, and reporting.

🔐 SOC Relevance
Phishing alerts are among the most common for L1 analysts.
A playbook helps:
✓ Speed up triage
✓ Avoid missed steps
✓ Provide clear escalation rules
✓ Standardize across teams

🔎 Basic Phishing Playbook Steps
✅ Detection → Identify suspicious email/log
✅ Verification → Analyze headers, URLs, attachments
✅ Containment → Block sender/IP, quarantine
✅ User Awareness → Notify affected users
✅ Escalation → Escalate to L2 if compromise confirmed
✅ Reporting → Document in SIEM/ticket, update CTI

🧰 Helpful Tools
SIEM (ELK, Splunk, Sentinel)
Email header analyzers
VirusTotal / Hybrid Analysis
PhishTool, MISP

💡 Pro Tip
Validate phishing indicators before escalation.
Update playbooks with lessons from real incidents.
📌 Even a simple, well-documented playbook can save hours in repetitive investigations.

💬 Ask yourself:
🔎 “What step is missing in my phishing response process?”

Qu’est-ce qu’un playbook SOC ?

Un playbook est un processus structuré pour traiter un type d’incident.
Il garantit rapidité et cohérence.
Pour le phishing, il inclut détection, vérification, confinement et rapport.

🔐 Utilité en SOC
Le phishing est fréquent pour un analyste N1.
Un playbook permet :
✓ D’accélérer le tri
✓ D’éviter les oublis
✓ De définir des règles d’escalade
✓ D’harmoniser les pratiques

🔎 Playbook Phishing – Étapes
✅ Détection → email/alerte suspecte
✅ Vérification → en-têtes, URLs, pièces jointes
✅ Confinement → bloquer expéditeur/IP, quarantaine
✅ Sensibilisation → informer l’utilisateur impacté
✅ Escalade → passer au N2 si compromission
✅ Rapport → ticket/SIEM, enrichir CTI

🧰 Outils utiles
SIEM (ELK, Splunk, Sentinel)
Analyseurs d’en-têtes email
VirusTotal / Hybrid Analysis
PhishTool, MISP

💡 Astuce
Toujours valider les indicateurs avant escalade.
📌 Même un playbook simple et bien documenté peut éviter des heures perdues.

💬 Posez-vous la question :
🔎 « Quelle étape manque à mon playbook phishing ? »


PHISHING
PLAYBOOK
Standard steps for SOC response
Detect, verify, contain & report
Faster, consistent
investigations
INAAM

6- Why configs matter in SOC tools

🧠 YAML & JSON power automation and detection
💡 Analysts must know how to read/edit them
❓ Have you ever tuned a rule or parser with config files?

🛑 What are YAML, JSON & config files?
SOC tools (SIEM, SOAR, IDS/IPS) rely on configuration files to define rules, parsers, playbooks, and integrations.

The most common formats are:
YAML (YAML Ain't Markup Language) → human-readable, used in playbooks & pipelines
JSON (JavaScript Object Notation) → structured data, used in APIs & logs
Config files (.conf) → tool-specific parameters (e.g., Logstash, Zeek, Suricata)

🔐 SOC Relevance
For SOC L1/L2 analysts, configs are essential to:
✓ Create or tune detection rules (SIEM, Suricata)
✓ Define log parsing pipelines (Logstash, Filebeat)
✓ Configure SOAR playbooks (Cortex, XSOAR)
✓ Integrate external APIs (VirusTotal, AbuseIPDB)

🔎 Examples
✅ YAML: playbook in TheHive or XSOAR defining steps for phishing response
✅ JSON: SIEM alert with structured fields (IP, user, timestamp)
✅ Config file: Logstash pipeline with grok patterns for log parsing

🧰 Helpful Tools
Text editors (VS Code, nano, vim)
jq for JSON parsing in CLI
YAML linters & validators
Documentation for each SOC tool

💡 Pro Tip
Always validate syntax before applying configs—one misplaced space in YAML or a missing comma in JSON can break the pipeline.

💬 Ask yourself:
🔎 "Do I understand how my SOC tools use configs to detect and enrich alerts?"

Qu’est-ce que YAML, JSON & les fichiers config ?

Les outils SOC (SIEM, SOAR, IDS/IPS) utilisent des fichiers de configuration pour définir règles, parsers, playbooks et intégrations.

Formats courants :
YAML → lisible, utilisé pour playbooks et pipelines
JSON → structuré, utilisé dans APIs et logs
Config (.conf) → paramètres spécifiques (Logstash, Zeek, Suricata)

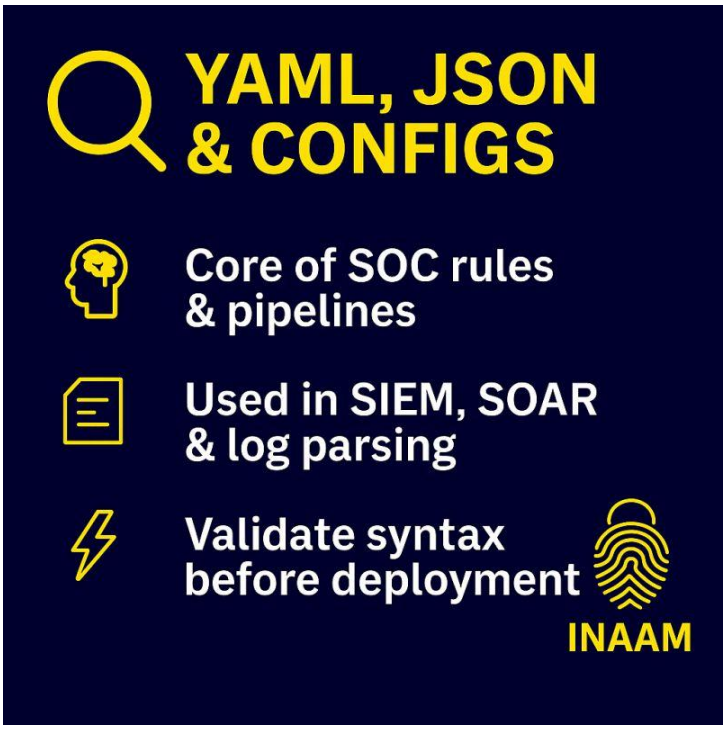
🔐 Utilité en SOC
Pour un analyste N1/N2, ces fichiers permettent :
✓ De créer/ajuster des règles de détection
✓ De définir des pipelines de parsing (Logstash, Filebeat)
✓ De configurer des playbooks SOAR (Cortex, XSOAR)
✓ D’intégrer des APIs externes (VirusTotal, AbuseIPDB)

🔎 Exemples
✅ YAML : playbook XSOAR pour phishing
✅ JSON : alerte SIEM avec champs IP, utilisateur, horodatage
✅ Config : pipeline Logstash avec motif grok

🧰 Outils utiles
Éditeurs (VS Code, nano, vim)
jq pour JSON en CLI
Validateurs YAML
Docs des outils SOC

💡 Astuce
Toujours valider la syntaxe avant application—un espace en trop dans YAML ou une virgule manquante en JSON peut casser tout le pipeline.

💬 Posez-vous la question :
🔎 « Est-ce que je comprends comment mes outils SOC utilisent les configs ? »


YAML, JSON
& CONFIGS
Core of SOC rules
& pipelines
Used in SIEM, SOAR
& log parsing
Validate syntax
before deployment
INAAM