Open Science
BRAZIL

DOI: https://doi.org/10.37497/opsbrazil.53

PREPRINT

MACHINE-SCALE ATTACKS: THE RISKS OF AGENTIC AI AND LESSONS FROM AN INCIDENT RESPONSE FOR OPERATIONAL DEFENSE

Ataques em escala de máquina: os riscos da IA agêntica e as lições de uma resposta a incidentes para a defesa operacional

RAFAEL ANDRADE ROSADO
Security Operations Center, Safetyware Tecnologia da Informação Ltda, Ribeirão Preto, SP, Brazil
E-mail: rafael.rosado@safetyware.com.br

ABSTRACT | Artificial intelligence (AI) expands the ability of malicious actors to reconnoiter exposed surfaces, chain tests, exploit vulnerabilities, and adapt their actions with less human intervention. This increases the pressure on controls designed for slower investigation and remediation cycles. This article examines the problem through a critical review of academic literature, primary threat intelligence sources, and an anonymized operational case study of incidents at a large Brazilian organization between October and November 2026. The investigation involved direct analysis in the CDN/WAF security dashboard, correlation of server and SIEM logs, assessment of exposures, and monitoring of the containment measures. Traffic labeled as AI bots made requests to sensitive paths, but the available data do not allow the authorship or the cause of the intrusions to be attributed to those bots. As a result, a four-front defense model is proposed—surface reduction, contextual detection, containment with compensating controls, and governance—supported by escalation criteria, an implementation roadmap, and indicators. The main contribution is to establish auditable operational decisions that depend neither on definitive identification of the attacker nor on the immediate acquisition of advanced platforms. The article concludes that AI intensifies preexisting risks and demands faster decisions, but defensive performance remains conditioned on the quality of the architecture, identities, telemetry, and response processes.

Keywords | Agentic Artificial Intelligence, Automated Attacks, Bots And Crawlers, Attack Surface, SOC, Incident Response, Compensating Controls, Security Governance

Submetido: 23-11-2026
Postado: 23-11-2026
e-ISSN: 2764-9822

Endereço de correspondência:
Rafael Andrade Rosado – E-mail: rafael.rosado@safetyware.com.br

CC BY

RESUMO | A inteligência artificial (IA) amplia a capacidade de agentes maliciosos de reconhecer superfícies expostas, encadear testes, explorar vulnerabilidades e adaptar suas ações com menor intervenção humana. Isso aumenta a pressão sobre controles projetados para ciclos mais lentos de investigação e remediação. Este artigo examina o problema por meio de uma revisão crítica da literatura acadêmica, de fontes primárias de inteligência de ameaças e de um estudo de caso operacional anonimizado sobre incidentes ocorridos em uma organização brasileira de grande porte entre outubro e novembro de 2026. A investigação envolveu análise direta no painel de segurança CDN/WAF, correlação de registros de servidores e SIEM, avaliação de exposições e acompanhamento das medidas de contenção. O tráfego identificado como proveniente de bots de IA realizou solicitações a caminhos sensíveis, mas os dados disponíveis não permitem atribuir a autoria ou a causa das intrusões a esses bots. Como resultado, propõe-se um modelo de defesa em quatro frentes: redução da superfície, detecção contextual, contenção com controles compensatórios e governança, sustentado por critérios de escalonamento, roteiro de implementação e indicadores. A principal contribuição consiste em estabelecer decisões operacionais auditáveis que não dependam da identificação definitiva do atacante nem da aquisição imediata de plataformas avançadas. O artigo conclui que a IA intensifica riscos preexistentes e exige decisões mais rápidas, mas o desempenho defensivo permanece condicionado à qualidade da arquitetura, das identidades, da telemetria e dos processos de resposta.

Palavras-chave | Inteligência artificial agêntica; ataques automatizados; superfície de ataque; resposta a incidentes; governança de segurança.

I. INTRODUCTION

The change facing cybersecurity operations comes less from new vulnerabilities than from the speed with which human actors and automated systems locate targets, try access paths, and persist when they find a misconfiguration. An organization may publish dozens of portals, APIs, development environments, and administrative services. For defenders, each published service requires inventory, an owner, testing, monitoring, remediation, and review. For the attacker, it is enough that one of those paths remains unprotected. The asymmetry is particularly serious because the marginal cost of repeating tests against many targets falls with the use of programmable infrastructure and, more recently, of agents based on language models.

In this context, the expression *machine-scale attack* describes the possibility of executing many reconnaissance, authentication, and exploitation actions in parallel, without waiting for a human operator to manually analyze each response. Not every bot uses artificial intelligence, and not every automated attack is autonomous: scanning scripts, vulnerability scanners, botnets, and brute-force campaigns have existed for decades. The relevant novelty is the combination of models capable of interpreting unstructured results with execution tools, browsers, test environments, code repositories, and mechanisms that make it possible to decide the next step in a sequence. Automation comes to cover part of the selection, adaptation, and chaining of procedures.

Experimental evidence predates the recent reports. Fang et al. [1], [2] demonstrated, in studies from 2024, that agents supported by language models could carry out web application exploitation tasks and, under bounded conditions, exploit known vulnerabilities after receiving descriptions of the flaws. These findings say little about the success rate

of real-world attacks. What they show is that some tasks that used to require technical expertise have become partially reproducible by systems with access to tools and sufficient context. Performance depends on the model, the information available, the environment, and the defensive barriers. For that reason, neither the reading that AI compromises any system nor the reading that nothing has changed relative to the old scripts holds up.

In October 2026, the discussion gained concrete operational examples. Microsoft documented the use of the agentic framework ARTEX in campaigns attributed to the group Storm-0940, including scanning, authentication attempts, and abuse of applications with previously granted permissions. Reuters described an operation in which agents coordinated by an open platform, Cairn, were used to attack a large number of websites. These accounts point to an adversary capable of repeating procedures at lower cost and, in some stages, obtaining privileged access before the traditional human workflow of triage and response completes its cycle. Neither of the two sources states that this pattern already accounts for the majority of cyber incidents [3], [4].

Time pressure also appears in the Microsoft Digital Defense Report 2026, which points to a reduction in the median time between the identification of vulnerabilities in real-world environments and their becoming available for exploitation, while enterprise remediation may still take weeks. Coordinated disclosure, availability of exploits, traditional automation, and reuse of tools already explained part of that gap before AI. The argument of this work is that AI aggravates a preexisting asymmetry by making it possible to increase scale and to automate parts that previously depended on human time [5].

In the corporate environment, AI is also on the inside. Coding assistants, support agents, workflow orchestrators, and interfaces to knowledge bases introduce new non-human identities and new operational capabilities. If an agent can query secrets, execute commands, or modify cloud resources, the organization needs to govern it with the same rigor applied to privileged accounts and production integrations. In practice, it is common to treat the identification of a known tool or bot as a guarantee about its behavior and about the intent of whoever invoked it, which it does not provide.

This article originates from the experience of a managed security services team that investigated incidents related to exposed applications, credentials, and development flaws at a large Brazilian organization. The observation of requests identified as originating from AI bots broadened the discussion about WAF exceptions and about the reading of sensitive paths. The case data do not prove that the intrusions were carried out by AI: the chains observed had independent compromise factors, including direct access that bypassed the protected edge. The account serves as an empirical basis for testing the applicability of the proposed controls and for making their limitations explicit.

The research question is: **how can an organization adapt prevention, detection, and response to automated attacks potentially amplified by AI without depending exclusively on advanced technologies or on conclusive attribution of authorship?** To answer it, the work combines a review of the literature and of technical reports with a retrospective analysis of an operational case. The general objective is to propose a defensive model that

reduces uninspected access paths, distinguishes noise from relevant exploitation, speeds up containment, and brings discipline to exceptions. The specific objectives are: to analyze the relationship between agentic AI and compression of the attack cycle; to separate resource request, actual exposure, and compromise; to identify compensating controls usable by teams at different maturity levels; and to build verifiable performance indicators.

The contribution of this text is applied: an evidence taxonomy, a four-front operational model, graduated criteria for incident escalation, action milestones, a 90-day implementation roadmap, and a set of metrics that can be reproduced and evaluated in other environments. The work does not offer a new algorithm, a controlled experiment, or statistics representative of multiple companies. The following sections discuss the theoretical foundations, the method used, the results of the case, the practical implications, and the limits of generalization.

II. LITERATURE REVIEW AND TECHNICAL FRAMEWORK

A. From Traditional Automation to Model-Driven Autonomy

Automation and autonomy are not synonyms. In traditional automation, preprogrammed rules execute relatively stable paths: testing credentials, crawling URLs, looking for vulnerable versions, or comparing responses against signatures. Agentic systems can receive a broad objective, select tools, interpret outputs, keep a memory of findings, and choose subsequent actions. The risk potential increases because a sequence no longer requires that every condition has been anticipated in code. The language model functions as a mechanism for interpretation and planning, while external tools carry out the actions. The effective reach depends on permissions, environment restrictions, model reliability, and the ability to execute commands.

Fang et al. [1] investigated agents capable of exploiting web applications in experimental scenarios; Fang et al. [2] evaluated the exploitation of published vulnerabilities, known as *one-day vulnerabilities*. In the second study, an agent succeeded on 87% of a set of 15 vulnerabilities when it received the corresponding description, but showed much lower performance without that context. That percentage holds for the sample and the experimental setup of the study; taking it as a general success rate for real attackers would be a mistake. The results suggest that the availability of technical information, integration with tools, and the capacity for adaptation are decisive variables. From a defensive standpoint, this recommends treating vulnerability disclosure and service exposure as components of a single operational decision.

The literature and threat intelligence also indicate important limits. Models can produce incorrect commands, misinterpret responses, lose context, or depend on supervision. In addition, reports of “AI attacks” may range from the generation of phishing messages to agentic multistep execution. To make the analysis comparable, this article distinguishes four levels: AI as support for creating content and code; AI as an analytical assistant that suggests actions; AI as an executor with tools and supervision; and AI as an orchestrator of

chained tasks with limited human intervention. At any of these levels, operational risk tracks the privileges and the autonomy granted to the system.

B. Operational Evidence from Campaigns in 2026

The campaign described by Microsoft on October 21, 2026, is especially relevant because it brings together telemetry, analytical attribution, and indications of speed. According to the company, the actor Storm-0940 used ARTEX to automate reconnaissance, password spraying, and activity in cloud environments, including after compromising a third-party multitenant application. The investigation found authentication attempts directed at more than 2,250 tenants and subsequent operations in part of the environments reached. In some episodes associated with the campaign, the interval between the start of the attack and privileged access was under three hours. Microsoft cautions that the mere presence of authentication or of an involved application does not prove that data were read or exfiltrated. This caution will be reproduced in the analysis of the internal case [3].

The account published by Reuters on October 19, 2026, based on research by NorForge, describes the use of agents coordinated by the Cairn system in a campaign against more than one thousand web applications, with compromises reported at more than one hundred organizations. The article presented estimates of model usage cost and of operating time per target. These figures are attributed to the published investigation, not to an independent measurement made in this study. What matters in the example is the possibility of a single operator supervising processes that previously required serial technical work. The consequence is an increase in the number of economically viable targets, including small and little-known services [4].

Another example is the PHP2Shell episode, disclosed in October 2026. Cloudflare described two critical vulnerabilities in PHP and the early deployment of WAF protections, including for free plans, before public disclosure. The case shows both the benefit and the limit of the virtual patch: the rule reduces exposure to known exploitation patterns but does not replace updating the vulnerable component at the origin. The existence of controls at the edge does not, by itself, prevent direct access to the server when the published service is not properly closed off [6].

The Microsoft Digital Defense Report 2026 [5] also reports that agentic automation appears at various stages of the offensive cycle and that most of the complex intrusions observed still involve significant human direction. For defense, the implication is twofold: to assume that reconnaissance and experimentation occur continuously and, even so, to evaluate each alert by evidence, context, and consequence.

C. Legitimate Bots, Identity Spoofing, and Content Exposure

A search crawler, a model-training bot, and a user-triggered agent can generate very similar HTTP requests. Their purposes, however, are not equivalent. Indexing crawlers traverse public content to discover and update indexes; training crawlers collect material according

to their own policies and the preferences declared by publishers; user-triggered agents may request specific pages during interactive tasks. In all cases, the decision to publish a given piece of data still belongs to the operator of the application. Verifying the bot's identity does not change that decision.

In August 2026, GreyNoise documented a set of scanners that falsely presented themselves under the names of known crawlers. The study identified 824 addresses associated with a cluster of spoofed identities and confirmed that they did not belong to the ranges published by the operators of the impersonated bots. It also observed searches for .env files, private keys, and other sensitive paths. The authors themselves record a decisive limit: observing a request does not mean that a file was returned or that a compromise occurred. The correct conclusion is that the User-Agent header does not constitute proof of identity and that controls must consider origin, cryptographic authentication when available, the purpose of the resource, and access behavior [7].

Cloudflare distinguishes verified bots and agents identified by mechanisms such as IP ranges, DNS, and signatures, depending on the category. This improves confidence in the declared origin but does not eliminate the risks of malicious users employing legitimate agents, of improperly accessible content, or of mistaken policy decisions. The most robust defensive rule is to keep inspection, authentication, and authorization tied to the resources, whatever name the visitor declares. The robots.txt file communicates crawling preferences and must not be treated as an access control for confidential information [8].

This distinction leads to three states that are frequently confused in incident reports. **Request** means that a client requested a path. **Confirmed exposure** requires evidence that the response or another channel delivered sensitive data to the unauthorized client; HTTP status code, size, content type, cache, and application logs can contribute, but an isolated 200 code is not enough. **Compromise** requires additional evidence of unauthorized use, execution, modification, credential acquisition, persistence, or another security consequence. These states may overlap in a single event, but they must be documented separately.

D. Exposed Applications, Identity, and Defense in Depth

The MITRE ATT&CK matrix organizes adversary behaviors into tactics and techniques, among them active scanning (T1595), exploitation of public-facing applications (T1190), use of valid accounts (T1078), credential access, execution, lateral movement, command and control, and exfiltration. The matrix describes observable behaviors but does not prescribe a single product to block them. Defensive modeling needs to cross each technique with prevention, detection, response, and evidence of effectiveness. Reconnaissance, for example, cannot always be eliminated: the goal is to prevent reconnaissance from revealing secrets or directly reaching systems that should remain behind controls [9].

The OWASP API Security Top 10:2023 brings together risks particularly close to those observed in modern applications: broken authentication, insufficient authorization, security misconfiguration, unrestricted resource consumption, and improper inventory management.

These categories explain why deploying a WAF does not automatically fix logic flaws, credentials embedded in code, excessive service privileges, or unprotected integrations. Defense needs to combine application security, network architecture, identity management, and runtime protection on the server. The edge layer stops protecting an asset that can be reached by an alternative path [10].

The NIST Cybersecurity Framework 2.0 offers a cross-cutting structure of governance, identification, protection, detection, response, and recovery. Revision 3 of NIST SP 800-61, published in 2025, integrates incident response into risk management and into the six functions of CSF 2.0; it replaced Revision 2. The CIS Controls v8.1, in turn, organize priority measures for inventory, configuration, vulnerabilities, logs, network, applications, and response. These frameworks do not, by themselves, establish universal times of one or four hours for investigating a web request; such milestones are operational proposals that depend on criticality and on the service contract [11], [12], [13].

E. AI as a Corporate Attack Surface and as a Defensive Control

Internal AI tools introduce non-human identities capable of accessing information and acting on systems. An application that allows an agent to read repositories, trigger pipelines, administer resources, or query databases concentrates permissions that would normally be distributed among operators and service accounts. The risk involves prompt injection, sensitive information disclosure, improper output handling, and excessive agency, categories described in the OWASP Top 10 for LLM Applications:2025. The NIST AI RMF [14] and its generative AI profile (NIST AI 600-1) provide a voluntary structure for assessing risks across the life cycle, distinct from any legal obligations that may apply [15], [16].

The compromise disclosed by Binance in November 2026 illustrates a complementary possibility: according to the company and to news reports based on the initial account, attackers started from compromised engineers' workstations and abused an internal AI tool during their progression toward sensitive systems. The technical details of the tool have not been fully published; therefore, the episode does not allow the conclusion that the model made autonomous decisions about the attack. Its value for this work is to show that internal agents must be assessed as privilege paths. Because it is more closely tied to social engineering, workstations, and transaction infrastructure than to edge bots, the case is included here only as support for the governance discussion [17].

The same capability that expands offensive operations can also improve defense: alert classification, code review, assistance with log analysis, contextual prioritization of vulnerabilities, and simulation of attack chains. These uses require quality evaluation, access control, data protection, recording of decisions, and human validation. Automated triage that merely produces persuasive diagnoses, but does not point to observable evidence, can increase risk. In a SOC, it makes sense to automate low-ambiguity decisions and to reduce the time the analyst spends understanding the situations that require judgment.

III. METHODOLOGY

A. Design and Purpose

The study uses a qualitative and descriptive approach, combining a critical review of publications with a single case study of an operational nature. The case is instrumental: it serves to reveal exposure mechanisms, detection difficulties, response decisions, and requirements for a defensive model, with no claim to statistically represent Brazilian organizations. The analysis is retrospective and starts from evidence produced during a real security intervention. The main unit of analysis is the **chain linking exposure, observation, interpretation, and control**; the individual incidents are subunits used to compare access paths and consequences.

No attacks were created for this work, there was no independent reproduction of exploits, and no artifacts that could identify the client's systems are made available. The primary reference is a preliminary post-incident report, supported by operational records and by the experience of the team that administered the controls. The executive report does not replace the formal forensic report of the specialized response team, whose conclusion, should it differ, would take precedence. This hierarchy prevents operational observations from being presented as definitive forensic findings.

B. Sources of Evidence and Triangulation

Four classes of material were considered. The first covers alerts and events centralized in the SIEM, mainly logs from web servers, authentication, endpoint protection, and connectivity. The second consists of the observations and actions carried out **directly in the CDN/WAF administrative console**, including analysis of security events, traffic identified as bots, configuration and removal of exceptions, and creation of rules. The third comprises response documentation, change records, meeting notes, and the preliminary post-incident report. The fourth brings together independent public sources, vendor reports, academic literature, and best-practice frameworks used to interpret the phenomenon; none of them serves as proof of facts of the private incident.

In the initial phase, the organization did not have the Enterprise contract plan of the CDN/WAF platform required for the intended log integration with the SIEM. The license was acquired during the response. This limitation concerns the **systematic centralization, retention, and correlation of logs**, not the team's operational visibility over the edge. The MSSP had administrative permissions and investigated the traffic directly in the console, applying security changes in the client's environment. The analysis of events in the dashboard, however, must not be automatically equated with a complete and immutable forensic export of all HTTP requests. Records presented by the dashboard may have sampling characteristics, query scoping, and retention that depend on the plan or the product [18], [19].

The triangulation sought to compare edge events, origin logs, host alerts, and the chronology of changes, whenever available. Gaps were preserved as gaps. For example, the team could demonstrate that an exception rule existed and that traffic labeled as bot was requesting sensitive paths, but it did not have sufficient proof that this rule had been exploited in the initial stage of all the intrusions. It was also possible to record the drop in requests reaching the servers after simultaneous changes at the edge, without estimating the individual causal contribution of each change.

C. Evidence Classification Criteria

To prevent excessive inferences, the article adopts five categories: **observed event**, when there is a record of behavior; **plausible exposure**, when a configuration would permit the action but evidence of its use is lacking; **confirmed access**, when there is a consistent record of unauthorized authentication, reading, execution, or modification; **undetermined attribution**, when the identity or the link between actions remains unknown; and **associated operational effect**, when the change in an indicator coincides in time with defensive measures without a causal isolation experiment. The use of these terms is deliberately conservative. Classifying an address as malicious in a reputation database does not, by itself, prove that all connections from that address have a malicious purpose. These categories address the strength of the evidence. The three states in Section II-C describe what happened to the resource, and the levels in Section VI-A guide the response.

In HTTP traffic, a request to /.env or to a configuration file is a sign of probing or of an attempted read. 401, 403, and 404 responses normally indicate denied access, nonexistence, or another condition without the requested content, but they need to be evaluated together with the behavior of the application. A 200 status requires checking the actual size, the body or hash when authorized, content type, authentication, rewrite rules, cache, and correspondence with the origin logs; applications can return a default page, a login page, or an error with a 200 status. Partial responses, redirects, and resources served from cache, in turn, demand additional interpretation. Only after this correlation is it appropriate to state that a specific secret was actually exposed.

D. Time Frame and Unit of Measurement

The reference report covers activities in October and November 2026, with the operational response concentrated in late October and early November. The specific series of searches for sensitive paths on the Nginx servers uses observations between October 28 and November 9, queried afterward in the SIEM. The number of servers contributing these logs varied throughout the crisis; therefore, the absolute count does not constitute a standardized rate per server or per application. The analysis presents values as observed counts and recommends, for later investigations, rates normalized by monitored origin, total number of requests, period, and configuration changes.

Two equivalent populations with random assignment were not compared, no control group was collected, and several changes were applied on nearby dates. The conclusion that can be drawn is that the set of actions coincided with a sharp drop in sensitive searches reaching the monitored servers. A specific causal relationship between the removal of a bot rule and the total reduction cannot be quantified with the available data. The fact that total traffic remained substantial makes it possible to rule out the trivial explanation that all access to the environment simply ceased, but it does not eliminate changes in coverage and routing.

E. Confidentiality, Risks of Bias, and Reproducibility

The names of the organization, brands, domains, IP addresses, users, financial amounts, products used in identifying contexts, and exploitation artifacts were omitted or generalized. The text should not be understood as a publication of the confidential report received; it uses only the lessons and the aggregate data necessary for the argument. The author has a professional relationship with the provider responsible for the operation, a circumstance that offers contextual access but may also introduce an interpretive bias in favor of the measures taken. For that reason, the version for review makes explicit the limitations of attribution, the simultaneous changes, and the need for external validation.

The reproducibility proposed is **procedural**, not the disclosure of sensitive data. Another organization can repeat the steps of inventory reconciliation, cross-analysis of edge and origin, evidence classification, exception review, egress assessment, and metrics. Similar numerical results are not expected to follow automatically. External validity will depend on application in diverse environments, calibration of false positives, and longitudinal analysis.

IV. RESULTS OF THE CASE STUDY

A. Exposure Architecture and Multiplicity of Paths

The organization analyzed operated public web applications, APIs, identity services, and internal tools distributed across different infrastructure environments. The author's MSSP administered the relevant security controls and worked on monitoring and response in coordination with the client's team and investigation specialists. The inquiry documented five distinct fronts of compromise or abuse, with no proven link between their perpetrators. Four of them involved web systems or published tools, while the fifth was related to legacy authentication in an identity service. With fronts this different, attributing all the events to CDN/WAF bot traffic would oversimplify the available evidence. Table I summarizes the five fronts.

Table I. Observed Fronts and Defensive Implications (Anonymized Description)

Front	Predominant path or weakness	Observed result	Derived priority control
CMS-based institutional application	Authentication exploited directly at the origin service, outside the expected inspection	Remote execution and server compromise	Lock down the origin, MFA, patch components, and block anomalous execution
Service platform APIs	Load balancer that forwarded traffic through a non-inventoried path	Reading of configurations and abuse of application functionality	Default deny at the load balancer, API inventory, and segregation of secrets
User identities	Legacy authentication protocol without an effective second factor	Account abuse and improper sending of messages	Block legacy methods, require MFA, and monitor password spraying
Internal AI orchestration tool	Administration interface exposed with a default credential	Command execution and reading of secrets; further progress limited by privileges	Remove public exposure and default credentials, control execution and connectors
Portal with file upload	Inadequate upload validation and excessive application privileges	Improper execution and code changes with impact on the business process	Validate files, separate privileges, protect integrity, and recover from a trusted source

These paths show that the effectiveness of a WAF depends on the actual coverage of the traffic. An application that responds directly at the origin address, through another domain, through a misconfigured load balancer, or through an independent administrative access remains vulnerable even when the main domain uses a CDN. In the case studied, part of the systems came to be protected by the edge during the response; direct publications were restricted and load balancers received stricter rules. The conclusion applicable to other organizations is that **inventorying domains is not enough**: it is necessary to reconcile DNS, certificates, public addresses, cloud services, load-balancing routes, firewall rules, and the assets that actually respond to requests.

The investigation also revealed weaknesses in defense in depth: outdated components, credentials in code or in configuration files, excessive application privileges, servers with broad outbound communication, and gaps in command auditing. None of these conditions is specific to AI. The case matters to the debate because locating and chaining these weaknesses becomes easier for an adversary that has persistent scanning and adaptation mechanisms. Because these were distinct incidents, the proposed model treats the factors as independent risk classes.

B. What Was Actually Observed About AI Bots

During the response, the operators investigated directly in the CDN/WAF console and identified traffic classified or presented as associated with search bots and AI bots. On one day analyzed, six bots accounted for approximately 183,000 requests to one of the portals. The initial interpretation considered the possibility of legitimate crawling. The later review of the requested paths showed searches for environment and configuration files, behavior incompatible with the purpose of a broad authorization for indexing institutional content. The classification as an AI bot derives from the identification attributes available in the

platform; it is not equivalent to independent proof of the real identity of each sender, as shown by the public cases of spoofing documented by GreyNoise.

The edge layer had exception rules that allowed differentiated passage for categories of bots. The exact administrative origin of the business request for this configuration remained pending documentary confirmation in the preliminary version of the report. The MSSP, with authorization from the owner of the environment, removed exception rules, strengthened blocking mechanisms, and deployed restrictions on sensitive paths. The observation of requests to these paths **does not automatically confirm** that the response contained the file sought, that the bot read a secret, or that it originated any of the intrusions. Such a conclusion would require corresponding origin or edge logs with body, size, and other evidence, or forensic artifacts demonstrating actual access.

There is also an operational distinction: although the license initially contracted did not cover the planned log integration with the SIEM, the security team had administrative access to the CDN/WAF and carried out investigation, traffic analysis, policy review, and rule deployment through the native console. The client acquired the necessary license during the response, which enabled log export to evolve. The initial limitation was therefore one of centralization and retention: events were analyzed in the console, with no exported trail for retrospective correlation in the SIEM.

C. Response Applied and Behavior of the Indicators

The containment measures were carried out in layers: closing direct paths to origin servers, reviewing CDN/WAF exceptions, temporarily blocking sources that were probing for sensitive files, strengthening host protections, expanding log collection, isolating affected servers, and recovering systems from trusted sources. Identity controls were also modified, such as rotation of secrets and restriction of exposed credentials and authentication methods. The exact order varied according to criticality and team availability; the case does not correspond to a previously planned linear implementation of the model described in this article.

On the instrumented Nginx servers, requests to sensitive paths recorded in the SIEM decreased from **16,815 on November 4 to 7 on November 9, 2026**, an approximate arithmetic reduction of **99.96%** in the observed count. Total traffic on the monitored servers remained high at the end of the period, with approximately ten million daily requests on the final day of the comparison. This supports the interpretation that there was a selective reduction in the sensitive searches reaching the origin, and not a general halt in web service. However, the group of servers with available logs varied over the series, from approximately 63 to 117 per day; the reduction is not a rate normalized by asset or by request.

The report also recorded significant fluctuation in security alerts during the deployment of agents and the review of rules. Part of the detections was related to legitimate administrative changes and to the arrival, in batches, of new events from the protection systems. For this reason, the mere drop in the total number of alerts was not taken as conclusive evidence

that the attack had been eliminated. The indicator of sensitive paths reaching the origins is closer to the edge control applied, but it is still subject to the effects of coverage and routing changes. A longitudinal evaluation should preserve series of requests allowed and blocked at the CDN, counts per application, agent coverage, and confirmation that the changes did not merely shift traffic from one path to another.

D. The Specific Contribution of the Case to the Proposal

The case reinforces four practical findings. First, the real attack surface is made up of all reachable paths, including those outside the officially managed domains. Second, exceptions for bots, vendors, or legacy systems can create routes without the same inspection applied to other visitors. Third, the SOC needs to correlate heterogeneous signals, avoiding turning every suspicious request into an "intrusion" or every outbound alert into "confirmed C2." Fourth, relatively accessible measures—denying unknown paths, locking down the origin, applying edge rules, restricting privileges, and enabling blocking on the host—were associated with the reduction in exposure without waiting for complete application replacements.

The case also presents an important limit for AI governance. The compromise of an internal orchestration tool demonstrates that AI platforms must be part of the inventory of critical services, even when they do not use their own models or store data classified as highly sensitive. By bringing together connectors, flows, tokens, and permissions, these platforms can enable jumps to higher-impact services. The restriction of privileges observed in part of the architecture helped limit the advance, which shifts attention from the AI interface to everything it can reach.

V. DISCUSSION AND ANALYSIS: A FOUR-FRONT DEFENSE MODEL

A. Principles of the Model

The proposal organizes controls into four interdependent fronts: **F1—Reduce the reachable surface; F2—Detect and classify early; F3—Contain and compensate; F4—Govern exceptions and human and non-human identities**. The ordering follows practical return: reducing an exposed path usually yields an immediate result, but only the simultaneous operation of the fronts sustains it. The design principle is to prevent the attacking machine from converting cheap reconnaissance into privileged access before the SOC has time to act.

Two premises guide the model. The first is that the declared identity of the requester does not prove its legitimacy or intent. The second is that defensive automation needs to be proportional to the degree of confidence in the signal. Immediately blocking a request for a secret path that cannot possibly be legitimate may be appropriate; isolating a critical server merely because it made an HTTPS connection to a new address can cause avoidable unavailability. Governance must establish which decisions can be automatic, which require human confirmation, and which are subject to formal exception and review.

B. Front 1: Reducing the Effective Attack Surface

The first action is to reconcile the external inventory with the protection architecture. The team should compare DNS records, issued certificates, public addresses and load balancers, cloud resources, firewall rules, old published services, and entries recognized by the CDN/WAF. The expected result is a matrix of **service, origin, domain, owner, criticality, entry control, application identity, and administration path**. If a service responds on an address or port outside the approved path, that discrepancy requires containment or a justified exception. Administrative interfaces and AI platforms should use private access or a zero trust model; obscuring the URL does not count as a control.

Locking down the origin is a priority. In architectures with a CDN/WAF, application hosts should accept connections only from authorized proxies or, preferably, cryptographically authenticate the communication between edge and origin, when the platform supports that design. Blocking by IP range is useful, but it requires periodic updating and protection against misuse of shared source ranges; authenticating the connection reinforces the measure. Load balancers need to reject unknown hosts and avoid generic forwarding to internal services.

Secrets must remain outside the public root and outside versioned code. The development pipeline needs to scan for keys and passwords before deployment, keep replacement records, and prevent configuration files, .git directories, backups, and dumps from being served by the application. Blocking patterns at the edge are a complementary defense. If a secrets file was actually delivered to an unauthorized client, it is necessary to rotate credentials and examine subsequent use, even if the source of the request declares itself to be a known crawler. Hiding the path from a specific bot leaves the underlying exposure intact.

Vulnerability remediation must consider **exploitability and exposure**, in addition to the CVSS rating. For published services, actively exploited vulnerabilities deserve escalation and deadlines different from those assigned to isolated systems. When installing a patch requires a version change, acceptance testing, or prolonged unavailability, a virtual patch, configuration mitigation, administrative restriction, or temporary removal from the internet may be adopted, always with an owner and a deadline for the definitive fix. On systems with host protection, blocking mode should be validated before mass activation to avoid undue impact on legitimate processes.

C. Front 2: Contextual Detection and Useful Telemetry

The second axis favors detections that combine an edge signal, application behavior, and a consequence on the host. The highest-value events include: sensitive paths requested repeatedly; authentication attempts against many accounts; crawler identity inconsistent with the verified origin; file upload followed by a suspicious process; shell creation by the web process; changes in code directories; first-time use of a deployment token; unusual

egress from a server that should not freely contact the internet. Each rule must state its source, conditions, exceptions, action, and the evidence required for confirmation.

To integrate CDN/WAF and SIEM in environments with high ingestion costs, it is advisable to define retention classes: allowed or exempted events in critical windows; events with high risk scores; representative samples or aggregations of blocked traffic; and complete trails of policy changes. Indiscriminately discarding everything that was blocked reduces cost, but it can eliminate the ability to reconstruct the start of a campaign, calculate rates, or validate rule effectiveness. Retention must consider contractual, regulatory, and forensic requirements and the operation's own financial capacity.

It is essential to preserve the visitor's original address and the request identifier in the origin logs, using trusted headers only when the connection actually arrived through an authorized proxy. Without this care, a server sees only the proxy's address or, worse, accepts a header forged by the visitor. To correlate attack flows, the following are recommended: timestamp with a consistent time zone, domain, host, method, normalized and original URI, status, response bytes, connection identifier, WAF rule, authenticated user when appropriate, and the process associated with the event on the server. Unnecessary personal information should be minimized in accordance with the retention policy.

The cost of the alert must also be assessed. A rule that generates thousands of repeated events without context competes for the analyst's attention with execution and persistence events. The best candidates for containment automation are high-confidence indicators, such as access to a unique decoy path or a web process creating a shell associated with a suspicious request. Rules based only on country, ASN, isolated reputation, or absolute request frequency should support triage and still require validation. Legitimate services can share infrastructure with malicious traffic, and addresses can change quickly.

D. Front 3: Proportional Containment and Compensating Controls

The third front recognizes that many organizations do not have an advanced bot management solution, a complete SIEM, homogeneous EDR, or an in-house round-the-clock team. Even so, there are things that can be done. In simple applications, a CDN with basic managed rules or a reverse proxy with an open-source WAF provides an initial layer; on the host, process auditing and integrity control can produce useful indications; at the firewall, restricting outbound traffic to authorized destinations and services limits C2 and exfiltration. The effect comes from the combination of these controls. Table II lists common gaps and compensating alternatives.

Table II. Reference Controls and Compensating Alternatives

Operational gap	Reference control	Applicable compensating measure	Recognized limit
No advanced bot platform	Managed identity verification and behavioral analysis	Blocking of sensitive paths, rate limiting, challenge, and IP/signature verification	Legitimate crawlers and users may be affected; requires review
Advanced WAF unavailable	Managed WAF with up-to-date rules	Basic CDN or open-source WAF with recognized rule sets	Does not fix business logic or a directly reachable origin
Patch cannot be applied in time	Fix of the component	Virtual patch, administrative restriction, and temporary unpublishing	Requires a definitive fix and effectiveness testing
EDR/IPS does not cover all hosts	Server protection in blocking mode	Process auditing, hardening, no execution in upload directories, and integrity monitoring	May detect without preventing exploitation in real time
No granular segmentation	Segregation and per-application policies	Egress firewall and service accounts with minimal scope	Dependencies on legacy systems may make complete denial difficult
Immature secrets management	Vault and automated rotation	Removal of secrets from the web root, repository scanning, and scheduled rotation	Requires discipline and inventory to avoid forgotten credentials
SOC without 24x7 human coverage	Continuous monitoring and response	Automation of high-confidence blocks, expiration, and tiered notification	Ambiguous cases still require human review

Containment needs to include a rollback mechanism: rule duration, owner, condition for revocation, impact record, and exception route. Temporary blocks on a set of IPs can reduce noise, but they must not serve as a substitute for closing sensitive paths or fixing the application. Propagation across CDN, firewall, and host protection should be done with scope control and prevention of duplicates. Poorly calibrated automation can compromise availability as quickly as the attack it is trying to stop.

E. Front 4: Governance of Exceptions, AI, and the Chain of Trust

Every security exception is a risk decision. Each bypass rule, inspection exemption, or expanded permission must record the requester, business purpose, technical owner, affected resource, justification, residual risk, approval, validity period, and review. Exceptions expire by default, unless renewal is documented. Management needs to cover CDN/WAF, firewall, identities, third-party integration, EDR, applications, and AI tools. A legitimate need for indexing must not result in a complete waiver of inspection for content that was never authorized for publication.

A bot policy should separate categories, instead of operating with a single “good bots” group. For search crawlers, public institutional content can be allowed subject to verified identity and rate limits. For bots intended for model training, the decision depends on the value to the business and on content usage policies. User-triggered agents should be treated as visitors that may carry unknown intentions; authentication, authorization, WAF, and API protection remain valid. Unverified bots receive no privilege for declaring a vendor name. In all categories, what cannot be served to unauthenticated users must not be served to crawlers either.

The internal AI policy needs to inventory models, agents, connectors, credentials, knowledge sources, and execution capability. Permissions should be granted per task and identity, with isolation between users, short-lived tokens, and human approval for high-impact actions. Tools that change infrastructure, execute commands, or handle payments require an auditable trail. Actions generated from untrusted external content must go through validation, because malicious instructions can arrive as the text of a page, document, or message. Governing AI, in this sense, means establishing verifiable boundaries for what the tools can read, write, and execute.

F. Attack Chain and the Controls That Interrupt It

Fig. 1 represents a didactic chain that runs from reconnaissance to impact. It should not be interpreted as a mandatory sequence, nor as an exact reconstruction of the five incidents analyzed. Attackers may start from a compromised identity, direct exploitation, the software supply chain, or third-party credentials. The mapping to MITRE ATT&CK tactics serves to justify the families of controls and to standardize the language among operations, audit, and management [9]. Table III details the relationship between stages, techniques, and controls; in the figure, C2, exfiltration, and impact appear together in stage 6.

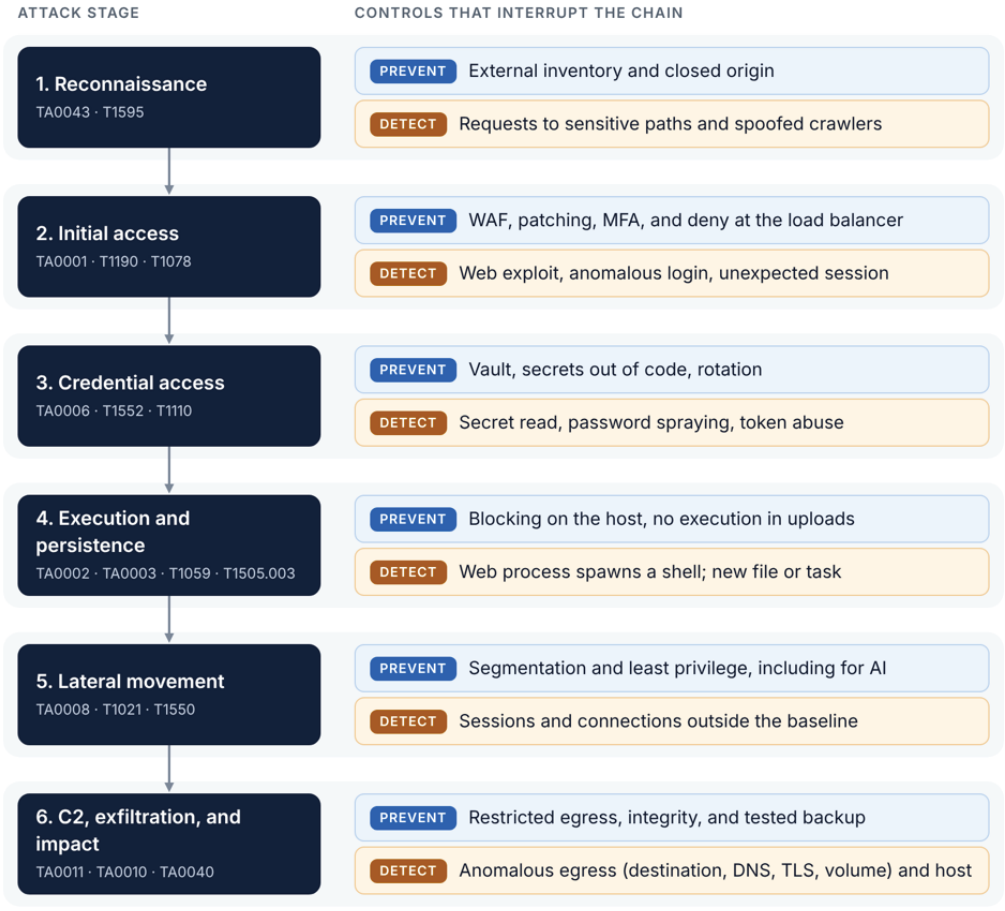

Fig. 1. Attack chain and interruption points (didactic model; prepared by the author).

Table III. Indicative Relationship Between Techniques and Operational Controls

Stage	ATT&CK examples	Prevention or reduction	Detection and response evidence
Reconnaissance	TA0043; T1595	Inventory, closed origin, minimal exposed content	Requests to sensitive paths and deviations from crawler patterns
Initial access	TA0001; T1190, T1078	WAF, patching, MFA, deny at the load balancer	Anomalous authentication, web exploit, unexpected session
Credential access	TA0006; T1552, T1110	Vault, removal of secrets, rotation, and password policy	Actual reading of a secret, password spraying, token abuse
Execution and persistence	TA0002, TA0003; T1059, T1505.003	Blocking on the host, immutable applications, upload control	Web process creating a shell, file, or persistent task
Lateral movement	TA0008; T1021, T1550	Segmentation and least privilege	Lateral sessions and connections outside the baseline
C2 and exfiltration	TA0011, TA0010; T1071, T1041	Restricted egress and per-service inspection	Periodic communication, suspicious destinations, anomalous transfer
Impact and recovery	TA0040	Separation of duties, integrity, tested backup	Improper modification of data or code, verifiable restoration

The relationship with NIST CSF 2.0 is cross-cutting: inventory corresponds mainly to Identify; preventive controls to Protect; telemetry to Detect; containment to Respond; safe return of service to Recover; and exception management to Govern. The CIS Controls v8.1 include inventory of assets and software, secure configuration, vulnerability management, logs, network defense, application security, and incident response. ISO/IEC 27001:2022 [20] offers relevant management and technology controls, while PCI DSS v4.0.1 [21] should be invoked as a mandatory requirement only when the environment and the contractual and regulatory scope fall under it. The mappings are indicative; they do not constitute an attestation of compliance, nor do they replace an applicability assessment.

VI. DETECTION, INVESTIGATION, AND RESPONSE PROTOCOL

A. From Anomaly to Incident: Decision Matrix

The expression “a scan only becomes an incident when something responds” is useful as a provocation, but technically insufficient. A public page can respond correctly to a scan without any violation; on the other hand, repeated attempts to exploit a critical application can justify opening an incident before compromise is confirmed, especially when an actively exploited flaw is involved. The proposed protocol separates **event**, **qualified suspicion**, **potential exposure**, **probable or confirmed compromise**, and **impact**. The degree of urgency combines potential consequence, quality of the evidence, criticality of the asset, and the ability to contain the event with low risk. Table IV brings together suggested triggers and responses, and Fig. 2 summarizes the flow, grouping these levels into three response bands.

Table IV. Triggers and Suggested Proportional Response

Situation	Mandatory checks	Initial classification	Proportional action
Requests to /.env, /.git/ config, and similar paths	Method, status, bytes, response type, source, volume, legitimacy of the scan	Probing/suspicion	Temporary block or challenge; correlation and logging
HTTP 2xx for a sensitive path	Content actually returned, fallback pages, cache, redirects, authentication, hash, and origin logs	Potential exposure until validated	Preserve evidence; treat the secret as compromised if the read is confirmed
Multiple login failures across many accounts	Time distribution, IP/ASN, fingerprint, subsequent successes, service accounts, known testing source	Attempted attack on identity	Restrict the source, protect accounts, investigate tokens and successful accesses
Web process creates a shell or modifies code	Parent process, command, user, binary, file created, associated request, change window	Probable/confirmed compromise	Isolate or restrict the host and preserve artifacts
New outbound connection from a web server	Destination, DNS, IP/URL, reputation, TLS, process, periodicity, volume, business purpose, and authorized change	Anomaly until correlated	Block a high-risk destination; expand hunting before isolating a critical service
Egress from a host with anomalous execution and a consistent IOC	Combined evidence from host, edge, DNS, EDR, and threat intelligence	High-confidence incident	Contain host, credentials, and communication route, with immediate notification

In HTTP responses, validation must observe the difference between **the path requested** and **the content returned**. Many applications respond with a 200 status and a generic HTML page even when the file does not exist; some proxies return cached content or error pages with an inconsistent code. A 206 status indicates partial delivery, which can also be sensitive. A 301 or a 302, in turn, does not prove a read, but its destination needs to be checked. The absence of an event at the origin does not prove the absence of delivery if the CDN responded directly from cache. When in doubt, the SOC preserves logs, reports the risk of exposure, and avoids asserting a “confirmed leak” before the artifacts have been compared.

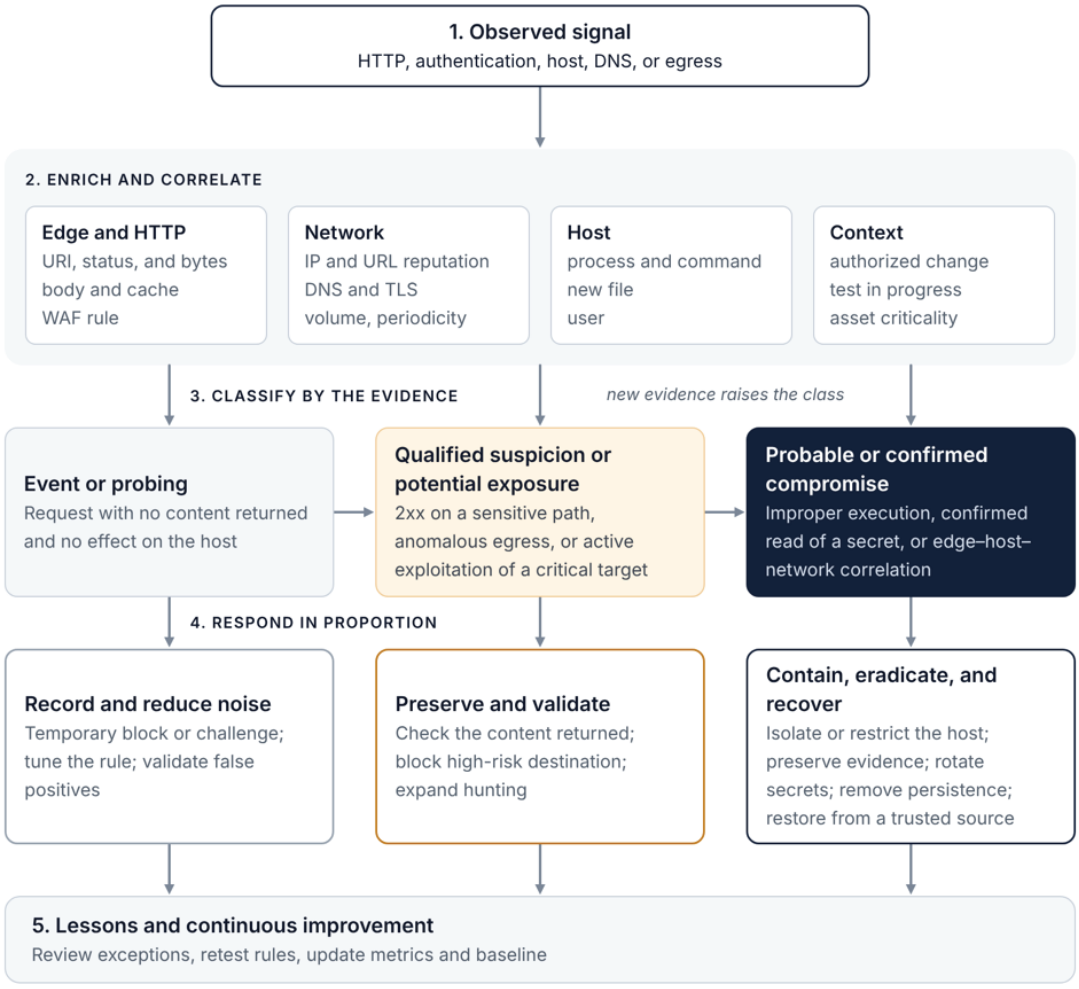

Fig. 2. Evidence-driven investigation and escalation flow (prepared by the author).

B. Investigation of Outbound HTTP/HTTPS Connections

In the case of an outbound connection initiated by a web server, the first criterion is to identify **which process** initiated the communication and which application need it serves. Update checks, business APIs, telemetry, and authentication can generate legitimate new connections. The classification should combine information from firewall, DNS, proxy, EDR, and operating system: IPv4/IPv6 destination, resolved domain, reputation and threat intelligence with observation date, domain age and history, ASN, geolocation only as context, certificate and TLS characteristics, port, HTTP method when visible, URI, volume transferred, frequency, and periodicity. A shared cloud provider address should not be automatically condemned on the basis of the aggregate reputation of the IP.

The signal becomes stronger when there is causal correlation: a suspicious request reaches the application, the web process creates a shell or a network tool, a new file appears in an upload or temporary directory, and then communication occurs with an unexpected destination. Other relevant signals are periodic connections at regular intervals, successive

attempts to resolve newly observed domains, transfer of a volume incompatible with the server's function, and unexpected use of protocols encapsulated in HTTPS. None of these elements in isolation constitutes universal proof of C2. Reputation should be treated as a contextual probability, subject to errors and rapid changes.

The initial response may involve blocking the destination, suspending the process with forensic preservation, segmentation, or complete isolation. The choice depends on the risk of the attack continuing and on the operational impact of the control. When there is strong evidence of unauthorized execution, preserving memory and disk before rebuilding—whenever feasible and safe—helps determine persistence and scope. If credentials or tokens were reached, the containment plan needs to include rotation, session revocation, and review of the associated privileges, in addition to blocking the network address.

C. Operational Milestones and Exit Criteria

The times below are **suggested targets** for operations with continuous service capacity. They are not deadlines set by NIST, ISO, or PCI DSS. They should be recalibrated according to severity, exposure, team availability, the need for forensic analysis, and contractual obligations. The order may be altered by incidents of greater criticality; isolation and evidence preservation can begin simultaneously. Table V consolidates the milestones.

Table V. Proposed Operational Protocol

Milestone	Objective	Minimum actions	Exit criterion
First 15 minutes, when confidence is high	Prevent immediate propagation	Reversible block, record of the decision, initial preservation, on-call activation	Emergency containment applied and owner engaged
Within 1 hour	Qualify and preserve	Check sources, impact, exposure, legitimate changes; collect edge and host logs	Initial hypothesis documented with evidence and uncertainties
Within 4 hours	Delimit access and consequences	Verify HTTP responses, identify credentials reached, start rotations, assess egress and sessions	List of assets/secrets at risk and containment plan
Within 24 hours	Investigate subsequent steps	Hunt for webshells, accounts, tasks, tokens, code changes, and lateral traffic; virtual patch when indicated	Preliminary scope per asset and evidence preserved
Within 72 hours	Eliminate the entry path	Lock down the origin, fix or restrict the service, remove persistence, rebuild when necessary	Service operating through an inspected path and critical credentials handled
Within 2 weeks	Learn and test	Review exceptions, run regression tests, validate alerts, and produce the post-incident report	Improvement plan approved, indicators and owners defined

Compatibility with NIST SP 800-61 Rev. 3 stems from the integration with Govern, Identify, Protect, Detect, Respond, and Recover; the flow does not intend to literally reproduce the normative diagram. The post-incident activity produces adjustments for continuous prevention and preparation. Compliance with standards such as ISO/IEC 27001 and PCI DSS requires a separate assessment of applicability, scope, and documentary evidence. The SOC

must maintain mechanisms for recording exceptions, preserving IOCs, communicating with the organization, and validating changes throughout the response.

D. Operational Coordination and Error Prevention

The investigation must classify facts, hypotheses, and discarded hypotheses. In crises involving multiple teams, contracted penetration tests, infrastructure changes, and mass deployment of agents can produce traffic and alerts similar to those of attacks. It is necessary to record the origin, window, authorization, and result of these activities, so as not to turn operational events into persistent false positives. At the same time, a legitimate test does not eliminate the risk revealed by the vulnerability found. Reducing noise depends on collaboration with the application and infrastructure teams, in addition to tuning detections in the SIEM.

When servers are rebuilt, service restoration must not be the only success criterion. Returning to production requires revalidating the protection agent, the logging agent, time synchronization, forwarding of the client's real address, outbound policies, code integrity, and application credentials. The reference report recorded lapses in instrumentation after rebuilds, reinforcing the need for this checklist. Crisis meetings can be split between a channel for technical decisions and another for communication with business areas and vendors, while keeping a single record of approved actions and their owners.

VII. IMPLEMENTATION, METRICS, AND VALIDATION OF THE MODEL

A. Incremental 90-Day Roadmap

Implementation can be organized into three thirty-day waves, without assuming that every service requires new tools. **First 30 days—close paths:** reconciled inventory of applications and origins, review of bypasses and bot permissions, blocking of sensitive paths, lockdown of the most critical origins, policy for load balancers, and restriction of legacy authentication. The deliverables are a list of exceptions with owners, a map of access paths, and evidence that priority applications do not respond through alternative routes.

Days 31 to 60 — see and contain: integration or systematic extraction of CDN/WAF events, preservation of a trusted source address in the logs, process auditing, rules for suspicious execution and password spraying, validation of shared blocklists, application of egress control on web servers, and host protection in an appropriate mode. Tests should combine expected attack scenarios with legitimate scenarios that could generate false positives, recording effectiveness and operational impact.

Days 61 to 90 — sustain and improve: approved publication policy, exception management, and AI agent governance; remediation deadlines differentiated by exposure; inventory of secrets and connectors; a response exercise simulating a chain from reconnaissance to egress; and the first cycle of indicators. Resources that are not fixed on time need a formal risk decision and a verifiable temporary control. The roadmap ends with measurable progress and a list of what is still missing.

B. Indicators and Measurement Quality

Table VI. Proposed Indicators for Monthly Tracking

Indicator	Measurement definition	Direction/initial target	Interpretation caveat
Edge coverage	Applications with all routes protected ÷ inventoried applications	Approach 100%	Verify origin addresses, not only DNS
Reachable direct origins	Assets that respond outside the approved path	Zero for assets without an exception	Test periodically from outside the network
Requests to sensitive paths reaching the origin	Count and rate per 1 million requests, per application	Sustained reduction	Normalize by coverage and volume
Confirmed exposure of a secret	Cases with evidence of improper delivery	Zero	Distinguish from a mere request
Time to qualification	First signal → documented decision	Downward trend	Separate high and low confidence
Time to containment	Qualified incident → effective block	Downward trend	Measure effectiveness, not just rule execution
Expired or ownerless exceptions	Rules without a valid record ÷ total rules	Zero	Include bot and AI configurations
Telemetry coverage	Hosts and applications with useful, intact logs ÷ inventory	Trend toward 100%	Measure events received, not just agent installed
Protection of accounts and agents	Privileged identities with adequate MFA, scope, and review	Trend toward 100%	Inventory non-human identities
Remediation time by exposure	Detection of an exploited critical flaw → verified patch or mitigation	Risk-based SLA	Do not count a virtual patch as a definitive fix

The “bots blocked” metric is insufficient: high numbers may result from aggressive rule behavior, higher traffic, or incorrect classification. It is recommended to create a baseline of false positives, service availability, legitimate visitor behavior, and results of controlled tests. Whenever possible, measure comparable periods and record rule versions, architecture changes, inventory variations, and deployment events. For the sensitive-paths indicator, keeping counts of 4xx/5xx, 2xx, and response size per application helps differentiate detection, blocking, and actual exposure. Table VI lists the proposed indicators.

C. Validation Plan in Other Organizations

Future validation can follow a pre- and post-implementation design with fixed asset inclusion criteria, three or more observation windows, and documentation of changes. Instead of asserting that the model reduces incidents by a universal percentage, the researcher should test specific hypotheses: does origin lockdown reduce access outside the WAF? Does exception review decrease permitted sensitive requests? Does integrating the CDN with the SIEM speed up the link between HTTP and execution? Does high-confidence automated blocking reduce containment time without increasing unavailability for legitimate use? Each hypothesis needs a metric, source, period, exclusion rule, and falsification criterion defined in advance.

The result must also be evaluated in terms of cost and operability. An organization without an internal SOC may prefer simple automations and support from an MSSP; another,

with multiple regulated environments, may require collection of raw events, long retention, and separation of environments. The choice of technology should not alter the principle of the model: **the ability to demonstrate that the exposed path is controlled and to reconstruct an event matters more than the number of products deployed**.

VIII. LIMITATIONS AND IMPLICATIONS FOR RESEARCH

This article starts from a single case and from an executive report that is still preliminary, prepared during a crisis. The author's access to the operational context makes it possible to describe decisions and limitations, but it does not constitute an independent audit. It was not possible to prove that AI bots caused the intrusions or that the different episodes had the same perpetrator. On several fronts, the initial traffic bypassed the edge; on another, the route of passage remained under verification. The crawler name presented in the events is not proof of the sender's identity or of the use of a specific model.

The defensive changes occurred together, the monitored population varied, and CDN/WAF telemetry only began to be centralized systematically during the response, after the licensing evolved. The drop in requests observed at the origin, although substantial, does not allow the individual effect of each control to be estimated. The proposed time targets have not been experimentally validated as universal SLAs. There are risks of selection bias in the choice of news reports, of limitations in vendor sources, and of change in the threat landscape after the documentary cutoff date.

Future research should apply the model in organizations of different sizes and sectors, compare normalized rates, investigate costs and false positives, and measure the impact of defensive agents on triage and remediation. Also deserving specific investigation are the effectiveness of bot signatures, the resistance of internal agents to prompt injection, and the governance of non-human identities with transitive permissions. Disclosure of additional case data requires formal authorization and a confidentiality assessment.

IX. CONCLUSION

The main result of the analysis is that attacks amplified by artificial intelligence do not necessarily require novel vulnerabilities to cause significant damage. Reconnaissance of external surfaces, exploitation of known flaws, credential attempts, and chaining of actions were already established practices. What AI adds, increasingly and in an empirically observable way in specific campaigns, is a capacity to interpret returned results, adapt procedures, and repeat operations at scale, with lower marginal cost and less need for continuous supervision. For defense, this reduces the time margin available between an exposure and its probable exploitation.

The literature review shows that there are academic grounds for treating offensive autonomy as a technical risk, but that its success rates depend on the test conditions and on the information available. Reports from 2026, especially ARTEX and Cairn, add examples of

real operations, even though most complex intrusions continue to involve people. Likewise, studies on crawler spoofing demonstrate that a request identified as an AI bot proves neither its legitimate origin nor the actual receipt of sensitive data. Methodological caution is part of defense: decisions based on undemonstrated attributions can direct teams to the wrong control.

The case study reinforced this distinction. The organization investigated faced access through distinct paths, including applications outside edge protection, weak credentials and protocols, application flaws, and an exposed internal AI tool. The MSSP analyzed the CDN/WAF directly during the response, even before log integration with the SIEM became feasible with the upgrade of the contracted plan. Traffic associated with bots requested sensitive resources, but no causal relationship was established between those requests and all of the compromises. What the case shows is an architecture with uninspected paths and poorly governed exceptions, vulnerable to automated exploitation whatever the identity of the attacker.

The operational response demonstrated the value of layered interventions. Closing direct origins, reviewing WAF policies, denying sensitive paths, improving server protection and auditing, and restricting privileges coincided with the drop in observed exposure. In the Nginx series, requests to sensitive paths reaching the servers fell from 16,815 to 7 in five days, with overall traffic still substantial. This result needs to be read as a temporal association with the set of measures: collection coverage varied and there was no control group.

The practical contribution of the article is a four-front model, supported by evidence criteria and a proportional response protocol. Reducing the surface involves inventorying all access routes, locking down origins, and preventing the exposure of secrets. Detecting early means combining edge, application, identity, and host, preventing a simple HTTP 200 or a low-reputation IP from being mistaken for compromise. Containing and compensating requires reversible high-confidence blocks, a fix or a validated virtual patch, segregation, and egress control. Governing means giving each exception an owner, a deadline, and a review, in addition to treating internal AI agents as identities with concrete, auditable privileges.

The proposal aims to be realistic for organizations with different resources. A basic CDN, careful configuration of load balancers, native auditing, and well-tested deterministic rules can reduce attack paths, although they do not replace higher-maturity protections or the remediation of vulnerabilities. The choice between commercial tools and open-source controls should respond to the risk of the asset and to operational capacity.

In machine-scale defense, the most sustainable gain comes from automating reliable checks, raising the quality of the evidence presented to the analyst, and reserving human judgment for changes in risk, uncertain causality, and business impact. Organizations do not need to know whether each attempt was conducted by an autonomous agent in order to prevent configuration files from being published or a server from responding through an unexpected route. They need, above all, to operate on the premise that these paths will be discovered quickly, and to be able to demonstrate, with records and tests, that their controls remain effective.

Declarations

Use of artificial intelligence. Generative AI tools (Claude) were used for grammar correction, writing review, formatting according to IEEE standards, organization of the literature, and generation of the diagrams in Figs. 1 and 2. The interpretation of the case, the conclusions, and the final responsibility for the content are exclusively the author's. The English (EN-US) version of this article was translated with the assistance of generative AI tools (Claude). The translated text was reviewed and validated by the author, who assumes full responsibility for its technical and linguistic accuracy.

Professional relationship and confidentiality. The author is part of the operation of the MSSP involved in the response to the case used as a reference. Data that would allow direct identification of the client and unnecessary details about the affected environments were omitted. The use of a preliminary internal report is not equivalent to disclosure or endorsement of its conclusions by all participating parties.

Nature of the model. The metrics, policies, deadlines, and normative mappings constitute recommendations of a technical and voluntary nature, to be adapted through risk analysis, contracts, and applicable legal requirements.

Internal documentary source, unpublished: Preliminary post-incident report of an MSSP, November 2026, and related operational observations. Confidential material used only as an aggregate methodological and factual reference, subject to validation of the final version and to institutional authorization for use in publication.

REFERENCES

[1] R. Fang, R. Bindu, A. Gupta, Q. Zhan, and D. Kang, "LLM agents can autonomously hack websites," arXiv:2402.06664, 2024. [Online]. Available: https://arxiv.org/abs/2402.06664

[2] R. Fang, R. Bindu, A. Gupta, and D. Kang, "LLM agents can autonomously exploit one-day vulnerabilities," arXiv:2404.08144, 2024. [Online]. Available: https://arxiv.org/abs/2404.08144

[3] Microsoft Security Research and Microsoft Threat Intelligence, "AI-driven, large-scale nation-state campaign using ARTEX enables supply chain compromise," *Microsoft Security Blog*, Oct. 21, 2026. [Online]. Available: https://www.microsoft.com/en-us/security/blog/2026/10/21/storm-0940-uses-artex-agentic-ai-connectwise-supply-chain-compromise/

[4] Reuters, "How one hacker used AI agents and a few thousand dollars to break into scores of sites," Oct. 19, 2026. [Online]. Available: https://www.reuters.com/legal/government/how-one-hacker-used-ai-agents-few-thousand-dollars-break-into-scores-sites-2026-10-19/

[5] Microsoft, *2026 Digital Defense Report: AI Is Changing the Physics of Cybersecurity*, Oct. 1, 2026. [Online]. Available: https://www.microsoft.com/en-us/security/security-insider/threat-landscape/2026-digital-defense-report

[6] Cloudflare, "Cloudflare deploys protections against php2shell vulnerability," *The Cloudflare Blog*, Oct. 21, 2026. [Online]. Available: https://blog.cloudflare.com/php-vulnerabilities-waf-protection

[7] GreyNoise, "Threat actors are posing as OpenAI, Anthropic and DeepSeek to target credentials and secrets," Aug. 28, 2026. [Online]. Available: https://www.greynoise.io/blog/threat-actors-posing-as-ai-crawlers

[8] Cloudflare, "Verified bots: Concepts and identity verification," *Cloudflare Docs*, updated Nov. 2, 2026. [Online]. Available: https://developers.cloudflare.com/bots/concepts/bot/verified-bots/

[9] MITRE, "MITRE ATT&CK: Enterprise matrix and techniques." Accessed: Nov. 13, 2026. [Online]. Available: https://attack.mitre.org/

[10] OWASP Foundation, "OWASP API Security Top 10: 2023." [Online]. Available: https://owasp.org/API-Security/editions/2023/en/0x11-t10/

[11] National Institute of Standards and Technology, *The NIST Cybersecurity Framework (CSF) 2.0*, NIST CSWP 29, 2024, doi: 10.6028/NIST.CSWP.29.

[12] National Institute of Standards and Technology, *Incident Response Recommendations and Considerations for Cybersecurity Risk Management: A CSF 2.0 Community Profile*, NIST SP 800-61 Rev. 3, 2025. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/61/r3/final

[13] Center for Internet Security, *CIS Critical Security Controls*, v8.1, 2024. [Online]. Available: https://www.cisecurity.org/controls/v8-1

[14] National Institute of Standards and Technology, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1, 2023, doi: 10.6028/NIST.AI.100-1.

[15] OWASP Foundation, "OWASP Top 10 for Large Language Model Applications: 2025." [Online]. Available: https://genai.owasp.org/llm-top-10/

[16] National Institute of Standards and Technology, *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*, NIST AI 600-1, 2024, doi: 10.6028/NIST.AI.600-1.

[17] Reuters, "Binance hackers used fake job interview in $6.3 billion attack," Nov. 3, 2026. [Online]. Available: https://www.reuters.com/world/binance-says-hackers-used-fake-job-interview-compromise-computer-before-63-2026-11-03/

[18] Cloudflare, "Security Events: Dashboard, visibility and availability," *Cloudflare Docs*, 2026. [Online]. Available: https://developers.cloudflare.com/waf/analytics/security-events/

[19] Cloudflare, "Logpush datasets and availability," *Cloudflare Docs*, 2026. [Online]. Available: https://developers.cloudflare.com/logs/logpush/logpush-job/datasets/

[20] *Information Security Management Systems — Requirements*, ISO/IEC 27001:2022, International Organization for Standardization, Geneva, Switzerland, 2022.

[21] PCI Security Standards Council, *Payment Card Industry Data Security Standard (PCI DSS)*, v4.0.1, 2024. [Online]. Available: https://www.pcisecuritystandards.org/document_library/

BIOGRAPHY

Rafael Andrade Rosado is a cybersecurity professional affiliated with the Security Operations Center (SOC) at Safetyware Tecnologia da Informação Ltda., working in security operations, cyber incident investigation and response, threat analysis, corporate environment protection, and information security governance. His research focuses on the risks associated with agentic artificial intelligence, automated cyberattacks, and operational defense strategies.

Rafael Andrade Rosado é profissional da área de cibersegurança, vinculado ao Security Operations Center (SOC) da Safetyware Tecnologia da Informação Ltda., com atuação em operações de segurança, investigação e resposta a incidentes cibernéticos, análise de ameaças, proteção de ambientes corporativos e governança de segurança da informação. Seus estudos abordam os riscos associados à inteligência artificial agêntica, à automação de ataques cibernéticos e às estratégias de defesa operacional.