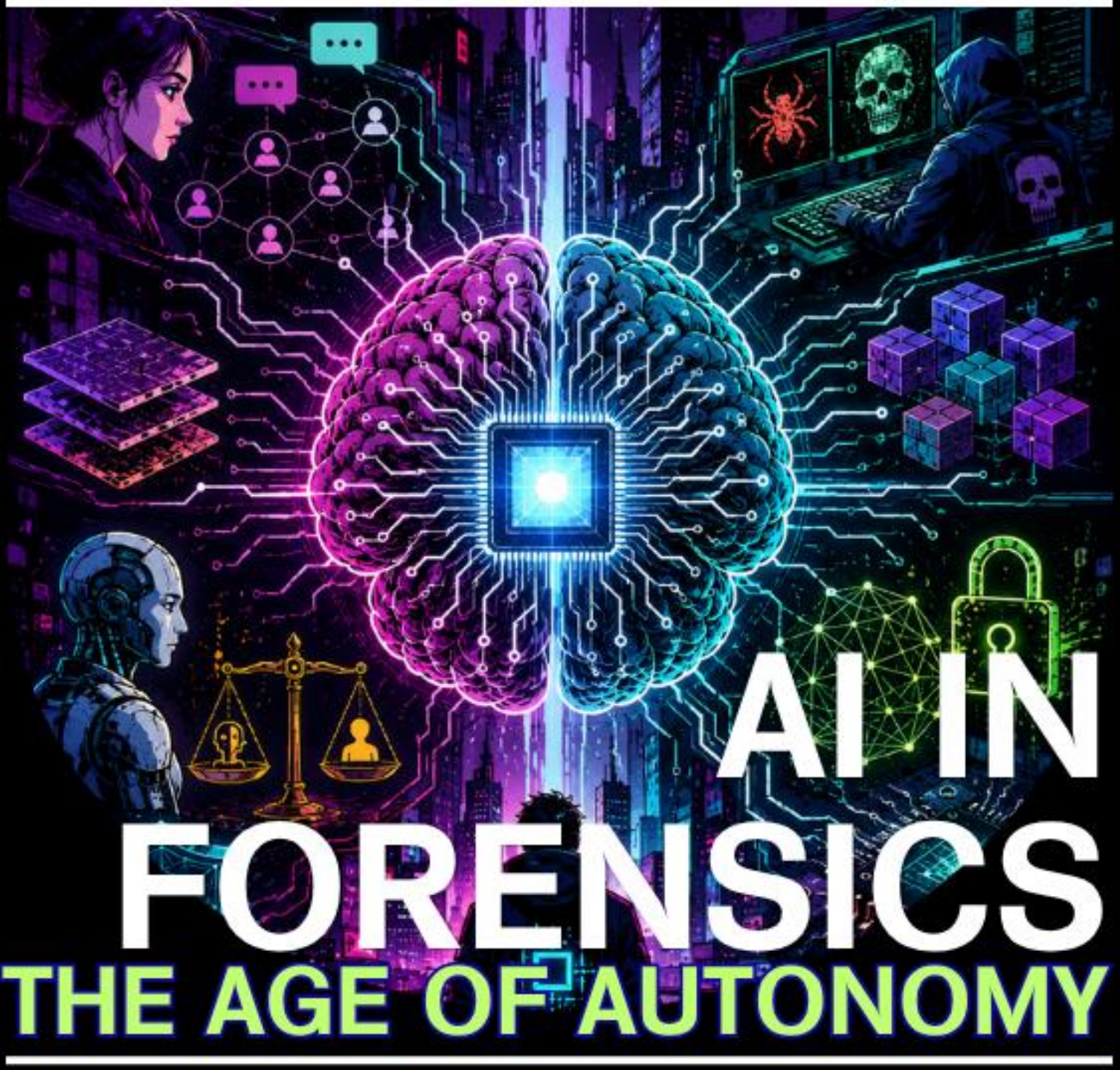
eForensics
VOL. 13
NO. 04
ISSN 2300-6986



AI IN FORENSICS
THE AGE OF AUTONOMY

Editor's Word

Dear Readers,

We live in an era where digital evidence surrounds us completely, and artificial intelligence is fundamentally redefining not only the modern threat landscape but the very fabric of how we conduct investigations. In this latest issue of eForensics, we bring you a highly curated collection of articles that cut straight through the ongoing AI hype to deliver practical insights from the front lines of digital forensics and incident response (DFIR).

The increasing use of artificial intelligence in forensic analysis is a worrying reality that could even create what Sammons, based on Saferstein's work, described in his classic book as "the CSI effect"—which creates the illusion that certain forensic science practices can solve any case and therefore develops unreasonable expectations that lead to erroneous verdicts. As we experience the challenges of AI in forensic investigations, the core issue shifts to the integrity of forensic evidence itself. The key ethical factor centers on how much evidence is a product of AI procedures rather than human analysis. Every verdict based only on AI is a null verdict.

Artificial intelligence has emerged as an indispensable force multiplier for lean security operations centers and overwhelmed forensic labs wrestling with massive data triage. However, this shift introduces complex operational and ethical dilemmas that until recently belonged strictly to the realm of science fiction. Can we maintain full procedural transparency when autonomous algorithms drive critical forensic decisions? Where exactly should we draw the line for AI agent permissions within an active SOC? Our contributors address these pressing questions head-on.

Inside this comprehensive issue, you will discover:

- **macOS Tahoe Binary Analysis via LLMs: Israel Torres walks through an exact, step-by-step methodology to coax ChatGPT into analyzing untrusted binaries and generating production-ready YARA rules, skillfully bypassing vendor-imposed platform guardrails along the way.**
- **Behavioral Identification of Chat-Based Abuse: Andreas Antonsen from STNDRDS AB shares a unique, multi-layered linguistic framework capable of identifying patterns of coercion, control, and psychological manipulation where traditional keyword-based filtering completely fails.**

- **AI Assistants as Targets and Weapons: Igor Korkin, Yuriy Tumanov, and Oksana Dokuchaeva dissect the rapidly expanding attack surfaces of Large Language Models, detailing the mechanics of context poisoning and corporate data exfiltration via active user browser sessions.**
- **The Boundaries of SOC Autonomy: Mayur Agnihotri staves off systemic risks by proposing an operational framework where an AI agent's level of permission is governed strictly by the reversibility of its actions, rather than subjective risk assessments.**
- **Post-Quantum Cryptography & Drone Warfare: Explore the field deployment of quantum-resistant ML-KEM-512 algorithms on resource-constrained ESP32 IoT microcontrollers, alongside a fascinating investigation into how Adversarial Machine Learning (AML) introduces a "forensic fog" during drone swarm crash analysis.**

We close this issue with an engaging, deep-dive debate into the ongoing reality of AI-Malware. Will on-the-fly, LLM-generated polymorphic code truly evade next-generation EDR solutions, or do these machine-generated threats leave behind highly distinct, structural footprints that make them inherently detectable to a trained eye?

Whether you are a seasoned DFIR practitioner, an academic researcher, or a cybersecurity enthusiast, this issue provides the strategic and technical insights necessary to successfully navigate the complex digital battlefield of 2026. AI is becoming a native member of the investigative team—not to replace human intuition, but to allow us to move faster, analyze deeper, and ultimately stay ahead of the curve.

Wishing you an inspiring and insightful read,

Ewa & Paulo & the eForensics Team

Beyond the Hype: Practical Considerations for AI-Assisted Malware Investigation *by Scott A. Macri, Founder & CEO, BITSnBYTES.io, LLC*

What you should know:

Readers should have a basic understanding of malware analysis concepts, including static analysis, dynamic analysis, sandbox reports, indicators of compromise, threat intelligence, and incident response workflows. Familiarity with common cybersecurity terms such as command-and-control, persistence, credential access, attribution, and confidence levels will be helpful, but deep reverse engineering experience is not required.

What you will learn:

Readers will learn how to evaluate AI-assisted malware investigation in practical operational terms. The article explains where AI can help analysts today, where it can mislead, why evidence and human review remain essential, how to protect sensitive investigation data, and how security teams can measure whether AI is improving investigation quality rather than simply producing faster or more polished output.

Introduction: Why Hype Is Not Enough

Artificial intelligence has quickly become one of the most promoted ideas in cybersecurity. In malware investigation, the promise is especially appealing: faster triage, clearer summaries, automated explanations, and help connecting technical artifacts to operational decisions. For overwhelmed analysts facing a steady flow of suspicious files, alerts, indicators, sandbox reports, and incident data, any technology that can reduce friction deserves serious attention.

The evidence problem

But malware investigation is not simply a speed problem, and it is not only a language problem. It is an evidence problem.

A useful investigation depends on what can be observed, what can be verified, what remains uncertain, and what conclusions the evidence can reasonably support. A suspicious executable may contain misleading strings. A sandbox report may show behavior that only appears under certain conditions. A network indicator may be shared across unrelated activity. A code similarity may suggest reuse, but not authorship. Even

experienced analysts must work carefully through ambiguity, incomplete data, and adversary deception.

This is where the hype around AI can become risky. AI-generated output can sound confident even when the underlying evidence is weak, incomplete, or misunderstood. A polished summary can make a tentative finding appear stronger than it is. A plausible explanation can obscure the fact that an analyst still needs to validate the original artifacts, review tool output, and consider alternate interpretations. In malware analysis, a wrong conclusion is not just an academic mistake. It can influence containment decisions, threat reporting, escalation, resource allocation, and leadership confidence.

That does not mean AI has no place in malware investigation. Used carefully, AI can help analysts summarize long outputs, organize observations, draft reviewable notes, translate technical findings for different audiences, and identify questions that still need answers. These are meaningful contributions, especially when teams are under pressure to move quickly. The problem begins when AI is treated as an authority rather than an assistant.

The practical test

The practical question is not whether AI can be added to malware analysis. It can. The more important question is whether AI improves the investigation without weakening analytic discipline. Does it help analysts reason from evidence? Does it preserve uncertainty? Does it make conclusions easier to review? Does it help teams communicate findings more clearly without overstating confidence? Does it reduce cognitive burden while keeping the human analyst accountable for the final judgment?

This article looks beyond the marketing promise of AI-assisted malware investigation and focuses on practical use. It examines where AI can help today, where it can mislead, how analysts should review AI-generated outputs, and what security teams should consider before trusting AI-supported findings in operational environments.

Malware Investigation Is an Evidence Problem Before It Is an AI Problem

Start with observable artifacts

Before evaluating what AI can do for malware investigation, it is worth returning to what malware investigation actually requires. At its core, the work begins with evidence: a file, a hash, a URL, a memory artifact, a process tree, a registry change, a packet capture, a sandbox trace, a YARA match, a suspicious command line, or an analyst observation. Each artifact may be useful, but none of them automatically tells the full story.

The analyst's role is to determine what the evidence supports, what it does not support, and what remains unknown. That distinction is critical. A file hash can identify a specific sample, but not necessarily its operator. A command-and-control domain can show

communication behavior, but not always intent. A matching string can suggest a capability, but it may also be dead code, copied code, junk data, or deliberate misdirection. A sandbox result can reveal behavior, but only under the conditions in which the sample executed.

Uncertainty remains

AI does not remove this uncertainty. In some cases, it can make uncertainty harder to see.

A model may summarize a sandbox report into a clean narrative: the sample establishes persistence, contacts a remote host, drops a secondary payload, and attempts credential access. That summary may be helpful, but the analyst still needs to confirm what happened. Was persistence observed directly, or inferred from a registry write? Was the remote host contacted successfully, or did the sample merely attempt resolution? Was a secondary payload recovered, or only referenced? Was credential access demonstrated by behavior, suggested by an API call, or assumed from a known malware family?

These differences matter. Malware investigations often influence operational decisions. An incident response team may isolate hosts, block infrastructure, escalate to leadership, notify partners, or change detection logic based on the conclusions analysts provide. If an AI system compresses uncertainty into confident language, it can cause teams to act on conclusions that were not fully supported.

This is why AI-generated analysis should be treated as interpretation, not evidence. The evidence remains the original artifact, tool output, observed behavior, and analyst-verified context. AI can help explain, organize, and summarize that material, but it cannot make weak evidence strong. It cannot turn a partial observation into a confirmed fact. It cannot replace the discipline of checking whether a claim is supported by the underlying data.

A practical use of AI begins with this boundary. Ask it to help clarify what is present in the evidence. Ask it to identify what still needs review. Ask it to separate observed facts from possible interpretations. Ask it to highlight assumptions. Those uses can improve the analyst's workflow without surrendering judgment to the model.

Better questions for AI

The best AI-assisted malware investigation does not start with the question, "What does the AI think this is?" It starts with better questions: "What do we know?" "How do we know it?" "What does this evidence actually support?" "What are we assuming?" "What would we need to confirm or reject this assessment?"

When AI helps analysts answer those questions more efficiently, it adds value. When it skips those questions and produces a confident conclusion, it becomes a risk.

Where AI Can Help Analysts Today

The strongest use cases for AI-assisted malware investigation are usually not the most dramatic ones. They are the practical, repetitive, cognitively expensive tasks that consume analyst time before a conclusion can be reached. In that role, AI can be valuable. It can help analysts move through large volumes of material, organize observations, and communicate findings more clearly. The key is to use AI where it supports reviewable work, not where it silently replaces judgment.

Summarization as navigation

One of the clearest near-term uses is summarization. Malware investigations often produce long and fragmented outputs: sandbox logs, process activity, file system events, registry changes, DNS lookups, HTTP requests, extracted strings, disassembly notes, antivirus labels, and threat intelligence references. Individually, these artifacts may be manageable. Together, they can become difficult to review quickly, especially during an active incident. AI can help condense that material into a first-pass summary that points analysts toward behaviors worth reviewing.

For example, an AI assistant might summarize that a sample attempted to modify startup locations, contacted multiple external domains, wrote files into a temporary directory, and spawned a child process with suspicious arguments. That summary can help an analyst prioritize review. It does not prove the sample established persistence, successfully communicated with command-and-control infrastructure, or completed a payload chain. Those details still require validation against the original tool output. Used properly, the summary is a navigation aid, not a conclusion.

Explanation and orientation

AI can also help explain unfamiliar technical details. Analysts frequently encounter APIs, command-line arguments, encoding patterns, packer artifacts, scripting behaviors, or operating system internals that require context. An AI assistant can provide a plain-language explanation of what a Windows API is commonly used for, what a PowerShell flag may indicate, or why a certain registry path matters. This can be especially useful for junior analysts or for experienced analysts working outside their usual specialty. The explanation should still be checked against authoritative references and the actual sample behavior, but it can reduce the time needed to orient the investigation.

Drafting and communication support

Another useful role is note drafting. Analysts often know what they observed but still need to convert fragmented observations into readable case notes or status updates. AI can help turn bullet points into a structured draft: observed behaviors, affected artifacts, possible significance, unresolved questions, and recommended next steps. This can improve communication between malware analysts, incident responders, SOC teams, and leadership. The draft must remain editable, and the analyst should remove unsupported wording, add caveats, and ensure that every claim reflects the evidence.

Gap identification

AI may also help identify gaps. A well-framed prompt can ask what additional evidence would strengthen or weaken a working assessment. For instance, if the current evidence suggests credential theft, the AI might suggest reviewing process access events, browser data access, LSASS interaction, command history, network exfiltration indicators, or related endpoint telemetry. The value is not that the model "knows" what happened. The value is that it can help generate a checklist of investigative questions the analyst may want to consider.

Comparison support

In some cases, AI can support comparison. If provided with prior case notes, known behaviors, or documented malware characteristics, it may help highlight similarities and differences between current observations and previous investigations. This can help analysts spot patterns, but it must be handled carefully. Similarity is not identity. Shared behavior does not prove shared authorship. Reused tools do not prove a common operator. AI can help surface possible relationships, but analysts must determine whether the evidence supports them.

Audience translation

AI can also help translate technical findings for different audiences. A reverse engineering note written for another malware analyst may be too detailed for an incident commander. A leadership update may need to explain risk, scope, and confidence without overwhelming the reader with raw artifacts. AI can help produce appropriate audience language, provided the analyst controls the message and preserves uncertainty. This is one of the safer and more useful applications because it improves communication after the technical review has already been performed.

These uses share a common pattern: AI helps organize, explain, draft, or prompt further inquiry. It does not decide. It does not replace the original evidence. It does not eliminate the need for analyst review. When used in this way, AI can reduce cognitive burden and help analysts spend more time on the parts of the investigation that require expertise: validation, interpretation, judgment, and communication.

Where AI Can Go Wrong

AI-assisted malware investigation becomes risky when the output sounds more certain than the evidence allows. Malware analysis often involves partial observations, conflicting signals, and adversary-controlled artifacts. AI systems are designed to produce coherent responses, but coherence is not the same as correctness. A clean explanation can still be wrong, incomplete, or unsupported.

One common problem is hallucinated technical details. An AI system may explain a function, behavior, string, or command in a way that sounds plausible but does not match the actual artifact. It may infer a capability from a weak indicator, describe behavior that was not observed, or fill gaps with patterns learned from similar-looking reports. In ordinary writing tasks, this may produce an inaccurate paragraph. In malware investigation, it can alter the direction of an incident response.

Another risk is misinterpreting tool output. Static analysis tools, sandboxes, disassemblers, memory tools, and detection engines often produce noisy or context-dependent results. A registry write may be suspicious in one context and benign in another. A failed network connection may be operationally different from a successful command-and-control session. A suspicious API import may indicate capability, but not execution. If AI compresses those distinctions into a simplified narrative, analysts may overstate what occurred.

Attribution is especially vulnerable to overconfidence. Malware labels, infrastructure overlaps, code similarities, and technique mappings can all suggest relationships, but they rarely prove authorship on their own. AI may combine weak signals into a strong-sounding conclusion, such as naming a malware family, campaign, or actor without sufficient evidence. This is dangerous because attribution claims often travel beyond the technical team. Once repeated in briefings or reports, they can be difficult to correct.

AI can also reinforce analyst bias. If an analyst already suspects a certain malware family or actor and prompts the system in that direction, the response may organize the evidence around that theory. The result can feel like confirmation even when the underlying evidence remains thin. This is not unique to AI. Analysts have always had to guard against confirmation bias. AI can accelerate the problem by producing polished support for a premature conclusion.

Another failure mode is the loss of uncertainty. Good malware reporting often depends on careful language: observed, attempted, likely, possible, suspected, unconfirmed, not observed, and unknown. AI-generated summaries may flatten these distinctions. A sample that “attempted to contact” an endpoint may become one that “communicated with” an endpoint. A behavior that “may indicate credential access” may become “credential theft.” A similarity to a known family may become identification as that family. Small wording changes can materially alter the meaning of a finding.

AI may also miss environmental context. Malware behavior depends on execution conditions, operating system version, privileges, network access, user interaction, geolocation checks, anti-analysis logic, and available dependencies. A sandbox result is not always a complete representation of real-world behavior. If AI treats one run as definitive, it may overlook conditions that prevented the sample from revealing additional behavior or caused it to behave differently than it would on a victim system.

There is also a communication risk. AI is good at producing readable narratives, and readability can create misplaced trust. A messy investigation may become a smooth story with a beginning, middle, and end. But real investigations often do not work that way. They contain unresolved questions, conflicting artifacts, and judgments made under uncertainty. When those rough edges disappear from the report, decision makers may believe the situation is clearer than it is.

The practical lesson is not that analysts should avoid AI entirely. It is that AI outputs must be treated as drafts, leads, or interpretations. They should be checked against original artifacts, tool output, and known context. Any statement that affects containment, attribution, reporting, or escalation deserves careful review. In malware investigation, the costliest AI mistake may not be a bizarre hallucination. It may be a reasonable-sounding answer that quietly exceeds the evidence.

The Attribution Trap

Few areas of malware investigation require more caution than attribution. Analysts are often asked to answer questions that sound simple: What malware family is this? Is this connected to a known campaign? Who is behind it? How confident are we? In practice, those questions are rarely simple. They require careful separation of observable facts from analytic judgments.

AI can make attribution riskier because it is good at producing fluent conclusions from incomplete signals. If a report includes a few recognizable behaviors, a familiar string, an infrastructure overlap, or a detection name from a security tool, an AI system may present a confident family or actor association. That answer may sound useful, especially under time pressure. But malware attribution is not a matching exercise based on one or two indicators. It is an analytic process that depends on the quality, uniqueness, and context of the evidence.

Reusable signals can mislead

Many artifacts used in attribution are reusable or misleading. Infrastructure can be shared, compromised, rented, abandoned, or intentionally copied. Malware code can be reused, leaked, purchased, modified, or borrowed from open-source projects. Techniques can be common across many actors because adversaries often adopt what works. Tool marks can be manipulated. Strings and metadata can be planted. Even behavioral similarity may only show that two samples solve the same operational problem in similar ways.

Labels are useful leads, not proof

Detection labels add another complication. Antivirus names and vendor classifications can be useful leads, but they are not final proof. Different vendors may use different names for the same family, the same name for related but distinct activity, or broad labels

that describe behavior rather than lineage. If AI treats labels as authoritative, it may convert a tentative classification into a firm statement. That can create a false sense of certainty.

The same problem applies to threat actors and campaign names. A model may recognize names from public reporting and connect them to observed behaviors, tools, or infrastructure. But public reporting varies in quality, age, confidence, and terminology. Some reports describe clusters of activity without naming an actor. Others use vendor-specific naming conventions. Some associations change over time as more evidence emerges. AI can compress this complexity into a neat answer that sounds more settled than intelligence actually is.

Good attribution requires disciplined language. Analysts should distinguish between what was observed and what is assessed. For example, “the sample attempted to contact this domain” is different from “the sample used known infrastructure associated with this actor.” “This behavior resembles prior reporting on a malware family” is different from “this sample belongs to that family.” “The evidence is consistent with” is different from “this was conducted by.” These distinctions are not academic. They tell decision makers how much weight to place on the conclusion.

AI-assisted workflows should preserve those distinctions, not erase them. When AI is used during attribution-related analysis, it should be asked to identify possible explanations, supporting evidence, contradicting evidence, and unresolved questions. It should not be asked to produce a final actor name from limited artifacts. Analysts should be especially skeptical of any output that gives a confident attribution without explaining the evidence and its limitations.

Treat attribution as a hypothesis

A safer approach is to treat attribution as a hypothesis. The analyst can ask: What evidence supports this association? What evidence contradicts it? Are the indicators unique enough to matter? Could the infrastructure be shared? Could the code have been reused? Are there alternative explanations? What confidence level is justified? What additional evidence would change the assessment?

This approach does not eliminate uncertainty, but it makes uncertainty visible. That is the point. In malware investigation, responsible attribution is not about reaching the most dramatic conclusion. It is about stating only what the evidence can support, explaining what remains unresolved, and being willing to revise the assessment when better information becomes available.

Human Review Must Be More Than a Rubber Stamp

The phrase “human-in-the-loop” appears frequently in discussions about AI and cybersecurity, but it can be too vague to be useful. In some systems, it means a person

clicks approve before an automated result is finalized. In malware investigation, that is not enough. Human review must mean that an analyst can inspect the evidence, challenge the interpretation, revise the conclusion, and document the reasoning behind the final assessment.

AI-generated findings should be treated like a draft from a junior assistant: useful, potentially insightful, but not authoritative. The analyst should ask where each statement came from, whether it is supported by the artifacts, and whether the wording overstates the evidence. A summary that says a sample "stole credentials" should be checked carefully. Did the sample access credential stores? Did it dump process memory? Did it invoke functions associated with credential access? Did it exfiltrate data? Or did the AI infer credential theft from a suspicious import, a detection label, or a behavior commonly seen in another malware family?

Separate facts, interpretations, and assumptions

A meaningful review process should preserve the difference between facts, interpretations, and assumptions. Facts are observations grounded in evidence: a hash value, a file path, a command line, a registry modification, a network request, a process relationship, or an extracted configuration item. Interpretations explain what those facts may mean. Assumptions fill gaps when evidence is incomplete. All three can be useful, but they should not be blended without distinction.

Look for alternate explanations

Analysts should also look for missing alternatives. If AI proposes that a sample is a downloader, what else could explain the same behavior? Could the observed network activity be a failed update check, an anti-analysis probe, or a decoy endpoint? If the system suggests persistence, was persistence achieved, or was the sample only attempting to write to a startup location? If a behavior resembles a known family, is the similarity distinctive, or is it common across many commodity malware samples?

This is especially important during active incidents, where time pressure encourages shortcuts. A polished AI summary can appear ready for reporting before the underlying work is complete. Analysts and team leads should resist that temptation. Review should include checking source artifacts, validating key tool outputs, preserving uncertainty, and identifying any unresolved questions that matter to containment or reporting.

Own the final judgment

Human review also has a communication role. Analysts are responsible not only for determining what happened, but for explaining how strongly they know it. A good final report should make clear which behaviors were directly observed, which conclusions are assessed, what confidence level is appropriate, and what evidence would be needed to

strengthen or revise the assessment. AI can help draft that language, but the analyst must own it.

The goal is not simply to keep a human somewhere in the process. The goal is human-led investigation. AI can assist with organization, explanation, and drafting, but the analyst remains accountable for the final judgment. In practical terms, that means AI should make review easier, not harder. It should help analysts see the evidence more clearly, not bury the evidence beneath fluent prose.

Protecting Sensitive Investigation Data

AI-assisted malware investigation raises a practical question that every security team should answer before using the technology: what information is being sent to the AI system, where is it processed, and who can access it afterward? Malware investigations often involve sensitive data. A sample may contain victim information, internal hostnames, usernames, credentials, proprietary documents, configuration details, source code fragments, law enforcement-sensitive information, or government data. Even a short prompt can disclose more than an analyst intends.

Know what leaves the environment

This matters because AI tools are often used through natural language interfaces. Analysts may paste sandbox summaries, endpoint logs, packet captures, filenames, file paths, command lines, decoded strings, or entire reports into a prompt. That material may seem routine during an investigation, but it can reveal internal infrastructure, business processes, affected systems, or details about an active incident. In some environments, it may also trigger legal, regulatory, contractual, or classification concerns.

Teams should avoid treating AI use as an individual analyst preference. It should be governed by policy. Analysts need clear guidance on what types of data may be submitted to external AI services, what must remain inside controlled environments, and what requires approval before use. The policy should account for the sensitivity of the investigation, the type of data involved, and the risk of exposing victim, customer, partner, or government information.

Retention and output handling

Retention is another concern. Before using an AI service, teams should understand whether prompts, uploaded files, generated responses, or metadata are stored; whether they may be used for model improvement; how long they are retained; and whether administrators or service providers can review them. These details affect whether the tool is appropriate for malware analysis, incident response, or threat intelligence work involving sensitive artifacts.

There is also a risk in AI-generated output. A summary can unintentionally include sensitive indicators, hostnames, user identifiers, internal IP addresses, or investigative assumptions that should not be broadly distributed. Analysts should review AI-assisted reports before sharing them outside the investigation team. They should remove unnecessary sensitive details, apply the correct handling markings, and ensure that the report is suitable for its intended audience.

Untrusted content and access control

Prompt injections and malicious content are additional concerns. Malware samples, scripts, documents, configuration files, and command-and-control responses can contain text designed to influence automated systems. If AI tools are asked to analyze untrusted content without safeguards, there is a possibility that malicious instructions embedded in the material could affect the model's response or the surrounding workflow. Analysts should treat untrusted text as evidence to be examined, not instructions to be followed.

Access control also matters. Not every analyst, responder, manager, or external partner should see every artifact or conclusion. Investigation data may need to be shared on a need-to-know basis. AI-assisted workflows should not become a shortcut around existing handling rules. If a team would not email a malware sample, credential dump, or victim-specific report to a broad distribution list, it should not casually paste the same content into an uncontrolled AI service.

Protecting sensitive data does not mean AI is unusable. It means teams need deliberate boundaries. They can sanitize prompts, remove identifiers, use approved internal tools, summarize only non-sensitive portions, restrict AI use to low-risk tasks, or require additional review for high-sensitivity investigations. The central principle is simple: AI should not weaken the controls that already exist around malware evidence, incident data, and intelligence reporting.

How to Use AI Without Weakening the Investigation

AI can be useful in malware investigation when analysts give it the right role. It should assist the investigation, not drive it. That starts with a simple rule: use AI to help examine and communicate evidence, not to invent conclusions beyond the evidence.

Prompt around observations, not verdicts

A practical way to apply this rule is to frame AI prompts around observations instead of verdicts. Rather than asking, "What malware family is this?" an analyst might ask, "Based on these observed behaviors, what possible capabilities should be reviewed, what evidence supports each possibility, and what additional artifacts would help confirm or reject them?" The second question keeps the work anchored in investigation. It invites the AI to support analytic thinking without pretending that a final answer is available.

Separate facts from interpretations

Analysts should also ask AI to separate facts from interpretations. A fact might be that a process created a file in a startup directory. An interpretation might be that the sample attempted persistence. A stronger conclusion, such as "the malware established persistence," should only be used if the evidence shows that the mechanism was successfully created and would execute as intended. This distinction is easy to lose in AI-generated summaries, so analysts should make it explicit.

Ask for uncertainty

Another useful practice is asking AI to identify uncertainty. For example: "What does this evidence not prove?" or "What alternate explanations should be considered?" These questions help counter the tendency of AI systems to produce smooth narratives. They also help analysts avoid premature closure, especially when early evidence appears to support a familiar malware family, behavior, or campaign.

Use checklists carefully

AI can also be used as a checklist generator. If the sample appears to perform discovery, credential access, command-and-control communication, or payload staging, the analyst can ask what additional evidence would normally be reviewed to support that assessment. The answer may remind the analyst to check process relationships, command-line arguments, registry changes, network timing, dropped files, memory artifacts, authentication logs, or endpoint telemetry. The checklist still needs expert review, but it can reduce the chance that important questions are missing during a busy investigation.

Treat generated language as editable

When drafting reports or notes, analysts should treat AI-generated language as editable working material. The final version should remove unsupported claims, add caveats, and preserve confidence levels. Words such as "confirmed," "likely," "possible," "attempted," "observed," and "not observed" should be used carefully. These terms are not filler. They communicate the strength of the evidence and help readers understand how much confidence they have in each finding.

Teams should also be cautious about using AI to merge multiple sources into one narrative. Combining sandbox output, static analysis notes, vendor labels, and threat intelligence references can be helpful, but only if the resulting summary keeps sources distinguishable. If the reader cannot tell whether a claim came from direct observation, a tool result, a third-party report, or AI interpretation, the output is not ready for operational use.

The same discipline applies to recommendations. AI may suggest blocking indicators, isolating hosts, escalating an incident, or hunting for related activity. Those suggestions can be useful starting points, but they should be evaluated against the organization's environment, risk tolerance, and operational constraints. A recommendation that makes sense in one network may be disruptive or incomplete in another.

Responsible use of AI should leave the investigation stronger than it was before. The analyst should have clearer notes, better questions, more organized evidence, and a more precise understanding of what remains unresolved. If AI produces a confident answer but makes it harder to see the evidence, challenge the reasoning, or explain uncertainty, it has weakened the investigation rather than improved it.

What Security Teams Should Evaluate Before Adopting AI-Assisted Malware Tools

Once a team understands where AI can help and where it can be misled, the next question is how to evaluate AI-assisted malware tools in practice. The answer should not start with the model's name, the interface, or the marketing claim. It should start with the investigation: does the tool help analysts produce better, faster, and more defensible work?

Start with evidence visibility

Evidence visibility should be the starting point. If an AI-assisted tool provides a summary, conclusion, or recommendation, analysts should be able to determine what information the output was based on. A statement about persistence should point back to relevant files, registry, service, scheduled task, or execution evidence. A statement about command-and-control behavior should be traceable to network activity, configuration data, or observed communication attempts. If the tool produces conclusions that cannot be checked against source material, it is creating trust without accountability.

Preserve uncertainty

Uncertainty is just as important. Malware analysis often produces partial answers. A useful tool should help preserve that uncertainty rather than hide it. It should make room for terms such as "observed," "attempted," "possible," "likely," and "unknown." It should not force analysts to binary conclusions when the evidence supports only a qualified assessment. Teams should be wary of systems that convert messy investigation data into confident labels without showing the reason behind them.

Keep analysts in control

Analyst control is another practical test. Analysts should be able to correct AI-generated summaries, reject unsupported interpretations, add context, and document why a conclusion changed. A tool that treats AI output as final is poorly suited for serious

investigation work. A tool that treats AI output as draft material for analyst review is more consistent with how malware analysis operates.

Fit the investigation workflow

A tool also must fit the way investigations happen. Malware analysis draws from static analysis, dynamic analysis, endpoint telemetry, sandbox results, network logs, memory artifacts, threat intelligence, and case notes. AI should help analysts work across those materials. It should not create a disconnected side channel where important reasoning lives outside the investigation record. If analysts must copy evidence into a separate chatbot, manually paste the response into a report, and then reconstruct where the claim came from later, the team may gain speed while losing traceability.

Protect sensitive data

Data protection cannot be treated as an afterthought. Before adopting a tool, teams should understand how samples, prompts, reports, attachments, and generated outputs are handled. They should know whether data leaves the organization, whether it is retained, whether it may be reviewed or used for training, and whether access can be restricted by role or sensitivity. If those questions cannot be answered clearly, the tool may not be appropriate for sensitive malware investigations.

Evaluate reporting quality

Reporting quality also deserves close attention. AI-assisted reporting should make findings clearer, not more dramatic. It should help analysts communicate what happened, what evidence supports the assessment, what remains unknown, and what actions are recommended.

Teams should review sample outputs carefully.

Do the reports distinguish facts from interpretations?

Do they preserve caveats?

Do they cite or reference supporting evidence?

Do they avoid unsupported attribution?

Do they help incident responders and decision makers act appropriately?

Measure operational value

Finally, teams should measure operational value, not demo appeal. During a pilot, they can ask whether the tool reduces triage time, improves consistency between analysts, helps identify missing evidence, improves handoffs, or produces clearer reports. They should also ask whether it introduces new review burdens, creates overconfidence, or causes analysts to spend more time correcting polished but unsupported text.

A useful AI-assisted malware tool should make the analyst stronger. It should improve visibility, organization, review, and communication. It should not ask the team to trust a black box because the output sounds convincing. In malware investigation, a tool earns trust by helping analysts stay close to the evidence.

Measuring Whether AI Is Actually Helping

AI-assisted malware investigation should be measured by operational value, not by how impressive the demonstration looks. A tool may generate a polished summary, produce a long report, or answer questions in natural language, but those outputs do not automatically mean the investigation improved. The better question is whether AI helps analysts reach more accurate, consistent, and defensible conclusions with less wasted effort.

Speed is not enough

Speed is useful, but speed alone is not enough. A faster summary has limited value if analysts spend the saved time correcting unsupported claims. A faster report can be harmful if it removes uncertainty or makes weak evidence appear stronger than it is. Teams should measure whether AI reduces friction without reducing analytic discipline.

Triage efficiency

One useful measure is triage efficiency. If AI helps analysts quickly understand the major behaviors in a sandbox report, identify which artifacts deserve closer review, or summarize repetitive tool output, it may reduce the time needed to reach an initial assessment. The important point is that the initial assessment should still be reviewed against the evidence. The goal is faster orientation, not faster overconfidence.

Consistency across analysts

Consistency is another important measure. Malware investigations often vary depending on analyst experience, time pressure, and available context. AI may help standardize note structure, remind analysts to consider common evidence categories, and produce clearer draft language. That can be valuable if it improves review quality across the team. It is less valuable if it simply makes every report sound similar while hiding differences in evidence quality.

Documentation and handoffs

Documentation quality should also improve. A good AI-assisted process should help analysts capture what was observed, what was inferred, what remains unknown, and what follow-up work is needed. If AI helps turn scattered notes into a clearer record, it can support handoffs between malware analysts, incident responders, threat intelligence teams, and leadership. But the final documentation must still reflect the analyst's judgment and should not include claims that were not verified.

Teams should also watch for reduction in duplicate work. During active incidents, multiple analysts may review overlapping artifacts, repeat similar searches, or recreate summaries that already exist elsewhere. AI can help by organizing prior observations and making existing context easier to find. This is useful only if the underlying information is accurate, current, and tied back to the investigation record.

Another measure is the quality of handoffs. Malware analysis rarely ends with the analyst. Findings may be passed on to SOC teams for detection engineering, incident responders for containment, threat intelligence teams for enrichment, legal or compliance teams for notification decisions, or executives for risk briefings. AI can help translate technical details into languages appropriate for each audience. The measure of success is not whether the text sounds polished. It is whether the recipient understands the finding, the confidence level, and the recommended action.

Measure negative effects

Teams should also measure negative effects. Does the tool encourage unsupported attribution? Does it make analysts less likely to inspect original artifacts? Does it generate summaries that require heavy correction? Does it expose sensitive data? Does it create another place where investigation notes must be managed? Does it increase review burden for senior analysts? These costs should be considered alongside any time savings.

The strongest evidence of value is improved decision quality. AI should help analysts ask better questions, find relevant evidence faster, document reasoning more clearly, and communicate uncertainty more accurately. If it does those things, it is contributing to the investigation. If it mainly produces confident prose, it may be improving appearance rather than analytic quality.

The Future of AI-Assisted Malware Investigation

AI-assisted malware investigation will likely become more capable, but its value will still depend on how well it supports disciplined analysis. The future is not likely to be a simple shift from human investigation to autonomous investigation. Malware analysis is too dependent on context, evidence quality, adversary behavior, and operational judgment for that to be a safe assumption. A more realistic future is one in which AI helps analysts move through information faster while humans remain responsible for interpretation and decisions.

Evidence summarization and case comparison

One likely area of improvement is evidence summarization. As tools become better at handling structured and unstructured information, analysts may be able to ask more useful questions across sandbox output, static analysis notes, endpoint telemetry, memory artifacts, and prior case records. Instead of manually searching through multiple reports, analysts may be able to ask what changed between two runs, which behaviors

were newly observed, or which artifacts remain unexplained. That kind of assistance could reduce time spent navigating data and increase time spent validating meaning.

AI may also become more useful for case comparison. Malware analysts often benefit from knowing whether a sample resembles prior investigations, whether similar indicators were seen before, or whether a behavior pattern has appeared in previous incidents. Future AI-assisted workflows may help surface those connections more quickly. Even then, similarity should remain a lead, not a conclusion. Analysts will still need to determine whether the comparison is meaningful, whether the evidence is current, and whether alternative explanations exist.

Investigation planning

Investigation planning is another promising area. AI could help analysts organize the next steps in an investigation based on what is known, what remains unresolved, and what operational decisions depend on the answer. For example, if the evidence suggests possible credential access but does not confirm it, AI could help identify what additional artifacts should be reviewed. This kind of support is valuable because it helps preserve analytic discipline under pressure. It helps analysts ask better questions rather than skip directly to confident answers.

Knowledge retrieval and governance

AI may also improve knowledge retrieval. Malware investigation often depends on remembering tool behavior, operating system internals, malware tradecraft, detection logic, and prior reporting. AI-assisted search and explanation may make that knowledge easier to access, especially for less experienced analysts. However, retrieved information must still be evaluated for accuracy, relevance, and date. Old reporting, vendor-specific labels, or unrelated examples can mislead if presented without context.

As AI becomes more integrated into security operations, governance will become more important, not less. Teams will need clearer policies for what data can be processed, how outputs are reviewed, how sensitive findings are handled, and how AI-assisted conclusions are documented. They will also need to decide when AI should be used, when it should be avoided, and when additional human review is required. The more capable the tool becomes, the more important it is to define its boundaries.

AI as teammate, not authority

The strongest future model is not AI as an independent malware analyst. It is AI as an analytic teammate that helps organize evidence, identify gaps, draft explanations, and support review. That model keeps the analyst in control while still taking advantage of useful automation. It also recognizes that the hard part of malware investigation is not merely producing an answer. The hard part is knowing whether the answer is supported, how confident the team should be, and what decisions can safely be made from it.

Conclusion: Practical Beats Magical

AI will continue to influence malware investigation, and that is not a bad thing. Analysts need help managing volume, organizing evidence, explaining technical behavior, and communicating findings under pressure. Used carefully, AI can support those tasks. It can help summarize long outputs, draft clearer notes, identify possible gaps, and make complex findings easier to understand across technical and non-technical audiences.

But AI does not change the fundamentals of malware investigation. The work still depends on evidence, context, validation, and judgment. Analysts still need to determine what was observed, what was inferred, what remains unknown, and how strongly the evidence supports a conclusion. A fluent AI-generated answer does not remove the need to inspect artifacts, review tool output, question assumptions, and preserve uncertainty.

The most practical approach is to treat AI as assistance, not authority. It can help analysts move faster, but speed should not come at the expense of accuracy or defensibility. It can help produce better language, but polished writing should not be confused with stronger evidence. It can suggest possibilities, but analysts must decide which possibilities are supported and which remain speculative.

Security teams should be cautious of claims that promise fully autonomous malware investigation, instant attribution, or final reports without meaningful review. Those promises are attractive because they reduce difficult discipline to a simple output. Real investigations are rarely that clean. They involve partial evidence, adversary deception, tool limitations, environmental conditions, and decisions made under uncertainty.

The better question is not, “Can AI analyze malware?” The better question is, “Can AI help analysts produce better, faster, and more defensible investigations?” If the answer is yes, the technology has value. If the answer depends on hiding uncertainty, trusting unsupported conclusions, or replacing analyst judgment with confident prose, the risk may outweigh the benefit.

Beyond the hype, the future of AI-assisted malware investigation should be practical, evidence-driven, and human-led. The goal is not magical automation. The goal is better analysis.

Good Uses and Risky Uses of AI in Malware Investigation

AI can be useful in malware investigation when the task is bounded, reviewable, and grounded in evidence. It becomes riskier when it is asked to make final judgments, infer intent, or produce conclusions that analysts cannot easily verify.

Table 1. Good uses and risky uses of AI in malware investigation

Good uses include	Risky uses include

Summarizing long tool outputs, such as sandbox reports, static analysis notes, endpoint logs, or extracted strings.	Assigning malware family names without strong supporting evidence.
Drafting investigation notes that analysts can review, correct, and refine.	Making threat actors or campaign attribution claims.
Explaining unfamiliar APIs, command-line options, file paths, registry keys, or operating system behaviors.	Producing confidence levels without explaining the evidence behind them.
Identifying follow-up questions or possible evidence gaps.	Generating final reports without analyst review.
Helping translate technical findings into clearer language for incident responders, SOC teams, leadership, or other stakeholders.	Combining weak signals into a polished but unsupported conclusion.

The safest pattern is to use AI to help analysts think, not to let AI decide what the evidence means.

Analyst Prompting Principle

When using AI during malware investigation, prompts should keep the model anchored to the evidence. A useful prompt does not ask the AI to guess the answer. It asks the AI to explain what the available evidence may indicate, what it does not prove, and what additional information would be needed to support a stronger assessment.

Table 2. Prompt framing for AI-assisted malware investigation

Weak prompt	Better prompt
“What malware family is this?”	“Based on the observed behaviors below, what possible capabilities should be reviewed? For each possibility, identify the supporting evidence, the limitations of that evidence, and what additional artifacts would help confirm or reject the assessment.”

This framing helps preserve analytic discipline. It encourages the AI to support the analyst’s reasoning process rather than produce a premature conclusion. It also reminds the analyst to look for uncertainty, alternative explanations, and missing evidence before reporting a finding.

In practice, the best prompts are specific, evidence-bound, and review-oriented. They ask the AI to organize observations, identify gaps, compare possibilities, and clarify wording. They do not ask the AI to replace the analyst's judgment.

Red Flags in AI Malware Investigation Claims

Security teams should be cautious when AI-assisted malware tools promise more certainty than malware investigation can usually support. Strong claims may sound appealing during a busy incident, but they should be evaluated carefully before they influence operational decisions.

Table 3. Common red-flag claims and why they deserve scrutiny

Claim	Why it deserves scrutiny
"fully autonomous malware attribution"	should raise concern. Attribution depends on evidence quality, context, source reliability, and uncertainty. It is not something that should be delegated entirely to an automated system.
"instant actor identification"	are also risky. Shared infrastructure, copied techniques, reused tools, and overlapping malware capabilities can all create misleading similarities. A fast answer is not necessarily a supported answer.
"no analyst review required"	should be treated as a major warning sign. Malware investigation requires judgment. Analysts need to inspect evidence, validate tool output, consider alternate explanations, and decide how strongly the evidence supports a conclusion.
"guaranteed detection"	should also be questioned. Malware behavior changes, environments vary, and adversaries adapt. No single tool or model can remove the need for layered analysis and continuous validation.
"one-click final report"	may sound efficient, but final reporting should not be reduced to automatic prose generation. A useful report should distinguish observed facts from

	interpretations, preserve confidence levels, identify unresolved questions, and support review.
“AI replaces reverse engineering”	oversimplify both AI and reverse engineering. AI may help explain code patterns, summarize observations, or assist with documentation, but it does not remove the need for expert analysis when the investigation requires deep technical understanding.

The more a claim suggests that AI eliminates uncertainty, analyst judgment, or evidence review, the more carefully it should be examined.

Key Terms

Table 4. Key terminology used in the article

Term	Definition
AI-assisted malware investigation	The use of artificial intelligence to support malware analysis tasks such as summarization, explanation, note drafting, evidence organization, and investigative planning. In this article, AI-assisted does not mean autonomous investigation.
Artifact	Any item of evidence reviewed during an investigation, such as a malware sample, hash, string, log entry, file path, registry key, packet capture, process event, memory artifact, or sandbox result.
Attribution	The analytic process of assessing whether malware, infrastructure, behavior, or activity may be associated with a malware family, campaign, threat cluster, or actor. Attribution should be treated as a judgment based on evidence, not as a simple label.
Command-and-control (C2)	Communication between malware and attacker-controlled or attacker-used

	infrastructure. C2 activity may support tasking, data transfer, payload delivery, or operational control.
Confidence level	A statement of how strongly the available evidence supports an analytic judgment. Confidence should reflect evidence quality, consistency, source reliability, and remaining uncertainty.
Dynamic analysis	Observing malware behavior during execution, often in a sandbox or controlled environment.
Indicator of compromise (IOC)	A technical indicator that may suggest malicious activity, such as an IP address, domain, URL, file hash, registry path, mutex, filename, or other observable.
Sandbox	A controlled environment used to execute and observe suspicious files or behaviors. Sandbox results are useful, but they may not reveal all behavior because malware can depend on timing, user interaction, environment checks, privileges, or network conditions.
Static analysis	Examining a file or artifact without executing it. This may include reviewing strings, headers, imports, metadata, embedded resources, packer indicators, or disassembly.
Threat intelligence	Contextual information about threats, including malware families, tactics, techniques, procedures, campaigns, infrastructure, and observed activity. Threat intelligence should be evaluated for relevance, age, confidence, and source quality.

References

National Institute of Standards and Technology (NIST). Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. July 2024.

Useful for understanding generative AI risk management considerations, including governance, validation, oversight, and organizational controls.

National Institute of Standards and Technology (NIST). Special Publication 800-61 Revision 3: Incident Response Recommendations and Considerations for Cybersecurity Risk Management.
Relevant to the incident response context in which malware investigation findings are often used for containment, escalation, coordination, and risk management decisions.

MITRE. MITRE ATT&CK®.
A globally accessible knowledge base of adversary tactics and techniques based on real-world observations, commonly used for threat modeling, detection, malware analysis, and cyber threat intelligence work.

Cybersecurity and Infrastructure Security Agency (CISA). Malware Next-Gen.
Relevant background for readers interested in operational malware analysis services and automated malware submission workflows.

Macri, Scott A. Malware on the Move: Rethinking Threat Analysis in the Age of AI and Adversarial Innovation. BITSnBYTES.io, LLC, April 2025.
Related prior work discussing malware volume, analyst fatigue, automation gaps, hybrid analysis, and the need for improved malware defense workflows.

About the Author

Scott A. Macri is the Founder and CEO of BITSnBYTES.io, LLC, a small business based in Ashburn, Virginia, specializing in secure software development, cybersecurity engineering, cloud-native systems, and federal technology solutions. He has extensive experience supporting government missions across software development, malware analysis engineering, DevSecOps, systems integration, and secure application delivery.

Scott has supported federal cybersecurity and national security programs, including work related to malware analysis platforms, threat investigation workflows, and mission-focused software modernization. He is also the creator of THRaXe, a deterministic cyber decision support platform designed to support structured malware analysis, intelligence correlation, and evidence-traceable analyst workflows.