The cybersecurity industry's testing standard community

Sandbox Evaluation Framework

Contributing Members:

- **Jan Miller (Lead Author)** – OPSWAT
- **Ralf Hund** – VMRay
- **Andrey Voitenko** – VMRay
- **Nima Bagheri** – Venak Security
- **Kagan Isildak** – Malwation

Version: 1.1 (Proposed Draft)
Date: 2026-06-18 (Draft for review)

Notice and Disclaimer of Liability Concerning the Use of AMTSO Documents

This document is published with the understanding that AMTSO members are supplying this information for general educational purposes only. No professional engineering or any other professional services or advice is being offered hereby. Therefore, you must use your own skill and judgment when reviewing this document and not solely rely on the information provided herein.

AMTSO believes that the information in this document is accurate as of the date of publication although it has not verified its accuracy or determined if there are any errors. Further, such information is subject to change without notice and AMTSO is under no obligation to provide any updates or corrections.

You understand and agree that this document is provided to you exclusively on an as-is basis without any representations or warranties of any kind whether express, implied or statutory. Without limiting the foregoing, AMTSO expressly disclaims all warranties of merchantability, non-infringement, continuous operation, completeness, quality, accuracy and fitness for a particular purpose.

In no event shall AMTSO be liable for any damages or losses of any kind (including, without limitation, any lost profits, lost data or business interruption) arising directly or indirectly out of any use of this document including, without limitation, any direct, indirect, special, incidental, consequential, exemplary and punitive damages regardless of whether any person or entity was advised of the possibility of such damages.

This document is protected by AMTSO's intellectual property rights and may be additionally protected by the intellectual property rights of others.

Version History

This section records substantive revisions to the Sandbox Evaluation Framework. Editorial fixes (typos, layout) are not separately tracked.

Version	Date	Summary of Changes
1.0	2025-03-26	Initial publication. Six KPIs covering Analysis Capability, Anti-Evasion Technology, Speed/Throughput/Scale, Reporting and Threat Intelligence, Integrations and Automation, and Security/Deployment/Maintenance. Five suggested weight profiles. Adopted by AMTSO on 2025-03-26.
1.1 (Proposed)	2026-06-18	Added a seventh KPI, “LLM-as-Sample Dynamic Analysis,” covering the sandbox’s ability to treat an LLM endpoint or model artifact as a sample, execute a structured prompt battery, and report observable behavior. Added a corresponding row to the KPI summary table in “Evaluation Framework.” Added a note to “Suggested Weight Profiles” explaining that default weights for the new KPI are deferred pending input from the AMTSO AI Working Group. No changes to the scoring formula, the 0/3/5/10 feature-score scale, or the grading bands. Author of v1.1 amendment: Jan Miller (OPSWAT).

Note: v1.1 is presented as a redline against v1.0. The substantive change is the addition of one new KPI; the framework structure, scoring methodology, and existing KPIs are unchanged.

Introduction

In the ever-evolving landscape of cybersecurity, the deployment of sandbox systems has become a crucial defense mechanism against emerging threats. These systems serve as a key line of defense in a processing pipeline, analyzing potentially malicious software, URL-based phishing attacks and web threats in a controlled environment before they can infiltrate an organization's network. With the ever-increasing sophistication of malware and evasion techniques, the need for robust and standardized testing frameworks to evaluate the effectiveness of sandbox solutions has never been greater.

The current scenario presents a fragmented landscape of open-source tools that individually address specific aspects of sandbox evaluation, such as anti-evasion techniques, speed, detection rates, cloud readiness, scalability, and compute cost. However, there is a notable absence of a comprehensive and standardized approach that integrates these crucial evaluation parameters into a unified framework. To address this gap, we propose the development of a versatile testing framework that offers a holistic assessment of sandbox systems.

Our motivation for this research is driven by the pressing need to establish a benchmark that not only evaluates sandbox solutions but also provides a means to compare their performance across key dimensions. This framework aims to streamline the evaluation process, offering clear insights into a sandbox's efficacy, resource efficiency, detection capabilities, and ability to counter evasion techniques.

Thus, the outcome of the evaluation framework will be the determination of a score per key performance indicator as well as an overall result. The weighting algorithms are part of this proposal.

Overview of Sandbox types and features

As part of the sandbox evaluation framework, it is essential to understand the different technologies used for dynamic malware analysis. Each type of sandbox—whether real-time dynamic, emulation-based, QEMU-based, or traditional VM-based—offers unique advantages and trade-offs in terms of speed, resource efficiency, and depth of analysis. The table below provides a comparative overview of these technologies, helping evaluators select the most suitable option based on their specific requirements, such as performance, scalability, or comprehensive behavioral insights. By clarifying these differences, organizations can better align their sandbox choice with their operational and security needs.

Key Features and Rationale

Inline Protection: Prioritizes low latency for seamless real-time protection.

- Suitable for scenarios requiring high-speed, basic behavioral profiling (e.g., email gateways).
- Note: Emulation-based sandboxes can also be configured for this use case when tailored for specific artifact types.

Dynamic Threat Triage: Balances speed with deeper analysis, supporting automated workflows.

- Facilitates IOC extraction and actionable insights for incident validation.
- Highlights the adaptability of emulation-based sandboxes in triage scenarios, where they can provide high-fidelity data without excessive latency.

Comprehensive Analysis: Focused on deep behavioral analysis for advanced threats, including evasive malware.

- Supports extended runtime for observing time-delayed or stealthy behaviors.
- Reflects the core strengths of emulation-based sandboxes in providing unparalleled depth and insight.

Threat Intelligence: Extracts and maps detailed IOCs to known TTPs, enhancing collaborative defenses.

- Emphasizes high scalability for bulk processing in TIPs while maintaining precision and accuracy.

Sandbox Type	Focus	Latency	Depth	Integration	Primary Use Cases
Inline Protection	Real-time threat interception	Very Low	Limited	Gateways, Proxies	Email/Web gateways, Web Application Firewalls, Inline malware filtering, Real-time attachment analysis
Dynamic Threat Triage	Balanced speed and depth	Low to Moderate	Moderate to High	SIEM, SOAR, EDR	EDR alert validation, Automated response workflows, User-reported phishing triage, Suspicious URL triage
Threat Intelligence	Intelligence generation	Moderate	High	TIPs, Collaborative Defense Tools	Threat Campaign Tracking, IOC extraction, Adversary attribution, MITRE ATT&CK mapping
Full attack chain analysis	In-depth behavioral analysis	High	Very High	Standalone, Forensic Tools	L3 Incident Response, Advanced Threat Research, Evasive malware detection, Complex attack chain mapping

Evaluation Framework

To accomplish a fair assessment, we introduce a structured evaluation framework that covers all key performance indicators (KPIs) needed to qualitatively assess sandbox solutions and allow their comparison. We propose the following high-level KPIs and scoring methodology:

KPI	Categories	Notes
Detection Capability	Content Analysis Depth, Behavioral Analysis Precision, Evasive Content Detection	Focus on outcomes (e.g., depth of IOC extraction, behavior classification) over mere support for file formats or artifacts.
Anti-Evasion Technology	Advanced Threat Detection Techniques, Outcome-Based Evasion Detection	Recognize alternative methods (e.g., instruction-level emulation) for achieving anti-evasion.
Analysis Depth	Comprehensive Reporting, Process & Network Visualization, Recursive Analysis	Includes features like process graphs, network dumps, and memory inspection, grouped to avoid over-segmentation.
Speed/Throughput/Scale	Sample Processing Time, Scalability, Resource Efficiency, SaaS Scalability, Multi-Platform Compatibility, Resource Efficiency, Cost Scalability	Ensure balance between speed and accuracy, avoiding overemphasis on throughput at the expense of detection quality. Ensure this is weighted appropriately and balanced against analysis depth and detection capability, avoiding bias toward lightweight architectures.
Deployment	Air-Gapped Deployment, CIS Compliance, GDPR Compliance, RedHat Support, SSO / Active Directory Integration, Regional Flexibility for SaaS, Retention Policies, Automated Backups	Retain features that showcase versatility across deployment scenarios.
Reporting and Threat Intelligence	Actionable Reporting, Threat Insights, ATT&CK Mapping	Retain a focus on the quality and precision of actionable reports, not just output formats.
Automation and Integration	SOAR Integration, API Flexibility, Threat Intelligence Sharing	Emphasize broad integration capabilities, including TIPs, APIs, and email-based submissions.
LLM-as-Sample Dynamic Analysis	Sample Intake, Execution Control, Stimulus, Behavioral Analysis, Reporting	Assesses whether the sandbox can ingest an LLM endpoint or model artifact as a sample, drive it under reproducible conditions with a structured prompt corpus, and report observable behavior in the same operational format as traditional sample analysis. Prompt-corpus content and response-taxonomy

KPI	Categories	Notes
		governance are deferred to the AMTSO AI Working Group.

Each of these indicators addresses a critical aspect of sandbox efficacy, allowing organizations to make informed decisions about which solution best fits their security needs.

For example, an organization focusing on a prevention use case may favor detection capability, speed, and scalability. An email security gateway vendor that needs to process a massive amount of files may favor detection capability, compute cost, and ease of deployment/maintenance, or a research lab might be interested in deep-diving memory dumps and dissecting a file from an incident response perspective.

Evaluation Score Formula

Based on the Weight configuration (see KPI table above), the final score of an evaluation can be calculated using the following formula:

Let S = {s1, s2, s3, ..., sn} be the set of scores, and W = {w1, w2, w3, ..., wn} be the corresponding weights.

1. Calculate the weighted sum (WS) as follows:

*WS = (s1 * w1) + (s2 * w2) + (s3 * w3) + ... + (sn * wn)*

2. Find the minimum and maximum values of WS within your dataset.
3. Normalize WS into the 0–100 range using the following formula:

*NormalizedValue = ((WS - MinWS) / (MaxWS - MinWS)) * 100*

Where:

- WS is the calculated weighted sum.
- MinWS is the minimum value of WS in your dataset.
- MaxWS is the maximum value of WS in your dataset.
- NormalizedValue is the final result, which will be in the 0–100 range.

Feature Set Scores

We propose that each KPI will come with a distinguished "feature set" and (optionally) a sample set / testing tools for validating the coverage. We recommend a score between 0 and 10 with the following meaning:

Score	Meaning
0	Not available
3	Limited Support
5	Comprehensive Support
10	Exceptional Capability

Please note that each feature set is intended to cover the most common features that we believe are critical to a variety of sandbox use cases: prevention, detection of targeted/zero-day malware, and forensic analysis.

KPI: Analysis Capability

This indicator assesses the sandbox's ability to accurately identify and classify malicious behavior. It evaluates the effectiveness of the system in detecting a wide range of threats, including known and unknown malware variants.

Feature Set

Feature	Category	Comment
Support Windows	File Type Support	PE, DLL, Powershell, VBS, JScript, Office (all flavors, including .DOC, .DOCX, XLM 4.0, .XLS, .PPT, .PUB, etc.), PDF
Support Linux	File Type Support	ELF, Bash, Lua, Python
Support Android	File Type Support	APK
Support OSX	File Type Support	MACH-O
Support URLs	File Type Support	URLs and HTMLs
Support Emails	File Type Support	EML, MSG, etc.
Support Archives	File Type Support	ZIP, ISO, etc.
Support Very Large Files	File Support	Very Large Files – Bigger than 1 GB
Process Spawn Capturing	Behavioral Analysis	E.g. via WMI
Memory Dumps	Behavioral Analysis	
Screenshots	Behavioral Analysis	
Injection Detection	Behavioral Analysis	e.g. APC, Process Hollowing, Atom Bombing
Live Interaction	Behavioral Analysis	e.g. to solve a captcha on a phishing redirect page
Automated Interaction	Behavioral Analysis	e.g. to click through an installer, move the mouse
Bootkit / Rootkit analysis	Behavioral Analysis	
Persistence capability analysis	Behavioral Analysis	Analyze autorun processes, user logon
Recursive processing of extracted files	Behavioral Analysis	
Binary disassembly	Behavioral Analysis	
Network Capture	Network and Communication Analysis	

Feature	Category	Comment
SSL Decrypt via TLS key interception / MITM	Network and Communication Analysis	e.g. C&C protocol analysis
DNS Spoofing	Network and Communication Analysis	Increase extraction of potential C&C network IOCs
Malware family detection	Content and Configuration Analysis	Detect malware families with YARAs, clustering, etc.
Config Extraction	Content and Configuration Analysis	
Generic Unpacking / Dynamic Payload Extraction	Content and Configuration Analysis	
Fuzzy hashes	Content and Configuration Analysis	
Certificate validation	Content and Configuration Analysis	
Compiler/RICH Header Parsing	Content and Configuration Analysis	
Phishing detection	Content and Configuration Analysis	Detect phishing login pages, identify affected service
IOC generation	Content and Configuration Analysis	File hashes, registry keys, domains, etc.
File extraction	Content and Configuration Analysis	Extraction of dropped and modified files
Function call logs	Content and Configuration Analysis	Function calls, syscalls, extracted strings, etc.
AV and reputation lookups	Configuration Analysis	Lookup known benign or malicious artefacts with reputation databases and/or AVs

KPI: Anti-Evasion Technology

In an era of sophisticated evasion techniques employed by cyber adversaries, this indicator evaluates a sandbox's ability to detect and counteract evasion methods, ensuring that threats cannot evade detection.

Feature Set

Feature	Category	Comment
Sleep Reduction	Evasion Technique	Avoid long sleeps, loops, time bombs

Feature	Category	Comment
MAC address spoofing / Hiding virtualized devices and UUIDs	Evasion Technique	VMWare, VirtualBox, Qemu have default MAC address values
CPUID spoofing	Evasion Technique	Instruction level VM detection
RDTSC / GetTickCount spoofing	Evasion Technique	Performance counter used for execution time measurement
Mouse/Keyboard simulation	Evasion Technique	Human simulation, execution trigger (e.g. via dialog box interaction)
Registry Key Spoofing	Evasion Technique	Hide registry artifacts that reveal presence of a VM / agent
Advanced Anti-Evasion	Evasion Technique	E.g. Thermal temperature, Firmware tables
Concealment of monitoring engine	Evasion Technique	Avoid direct detection of monitoring engine in analysis environment (e.g. agentless monitoring)
User environment randomization	Evasion Technique	Create random file artifacts on desktop before analysis
Anti API hammering	Evasion Technique	Avoid long loops with API calls
VPN support	Evasion Technique	Route dirty line traffic through different geolocations
Wear-and-tear fuzzy images	Custom Images	Avoid off-the-shelf vanilla execution environment
Configurable Application Stack	Custom Images	Enable mimicking a golden execution environment (e.g. for exploit trigger)
Customizable system environment (e.g. System locale)	Custom Images	Enable mimicking a golden execution environment
Network simulation	Simulation and Manipulation	Forensic use case and to further the attack chain analysis
Manipulate system tools (e.g. "ping -n" / ICMP echo delay, Task Scheduler)	Simulation and Manipulation	Usage of OS binaries to delay execution

KPI: Speed/Throughput/Scale

This Key Performance Indicator assesses the sandbox solution's ability to efficiently process potentially malicious files while maintaining detection quality. It encompasses:

- **Speed:** Measures the sample processing time to ensure rapid threat detection without compromising analytical depth.
- **Throughput:** Evaluates the volume of samples the system can handle concurrently, reflecting its capacity to support high-demand environments.

- **Scale:** Focuses on scalability across different environments, including SaaS deployments and multi-platform compatibility, while considering resource efficiency and cost scalability.

The goal is to maintain a balanced approach, ensuring that speed and throughput do not come at the expense of detection accuracy. This metric is weighted to reflect the importance of robust analysis depth, avoiding bias toward lightweight architectures that may compromise threat detection capabilities.

Feature Set

Feature	Category	Comment
Average Processing Time for Small Size Sample Set	Processing Time Metrics	
Average Processing Time for Large Size Sample Set	Processing Time Metrics	
Total Processing Time for Document Set (N=1000)	Processing Time Metrics	
Total Processing Time for Executable Set (N=1000)	Processing Time Metrics	
Max Throughput per Virtual Machine (Analysis Node)	Throughput and Parallel Processing	
Max Parallel Processing Tasks	Throughput and Parallel Processing	
Total Memory Usage	Resource Consumption	
Total vCPU Hours	Resource Consumption	
Total Disc Usage	Resource Consumption	
Cloud native	Deployment and Infrastructure	Not, if nested virtualization is required
Deployable as a container	Deployment and Infrastructure	E.g. Kubernetes Cluster
Can run in airgapped environments	Deployment and Infrastructure	
Ensures full privacy	Deployment and Infrastructure	i.e., no data is sent to the vendor or any third-party
Auto-Scaling Mechanisms	Scalability and Availability	Dynamic workload (scaling actions, trigger metrics)
High availability	Scalability and Availability	Single point of failure / Ability to maintain service even during failures, Uptime monitors
Sample Triage	Additional Capabilities	Ability to filter out samples with no active content, optimizing analysis resources

Feature	Category	Comment
		(primarily relevant for on-premises deployments)
Smart Caching	Additional Capabilities	Capability to effectively reuse results from past submissions for similar or identical samples (on-premises focus; cloud alternatives rely on SLA commitments)

Note: Sample Set may refer to a File or URL

KPI: Reporting and Threat Intelligence

Effective reporting is essential for incident response and decision-making. This indicator assesses the quality and comprehensiveness of reports generated by the sandbox solution, helping organizations gain actionable insights from analysis results.

Feature Set

Feature	Category	Comment
Single-file PDF	File Formats	PDF-A support is a bonus
MAEC	Security Standards and Frameworks	
STIX	Security Standards and Frameworks	
MITRE ATT&CK mapping	Security Standards and Frameworks	
JSON/XML Export	Data Export and Integration	
Automated E-Mail Notifications	Data Export and Integration	
Advanced Report Search	Threat Intelligence	e.g. Find reports sharing similar threat indicators or characteristics
Fuzzy Hashes	Threat Intelligence	Similar sample correlation / Unknown threat detection
IOC Scoring Capability	Data Quality	E.g. ability to determine the context/origin/prevalence of a potential IOC
Whitenoise Filtering	Data Quality	Avoid excessive reporting of behavior unrelated to the actual payload or attack chain
Threat Classification and Enrichment	Data Quality	

KPI: Integrations and Automation

Modern cybersecurity ecosystems rely on the integration of various tools and systems, as well as post-analysis automation. This indicator evaluates a sandbox's compatibility and ease of integration/automation with other security solutions, enhancing overall cybersecurity posture.

Feature Set

Feature	Category	Comment
Web API with automated documentation (e.g. OpenAPI)	Developer Tools for APIs and SDKs	
SDK with CLI	Developer Tools for APIs and SDKs	e.g. Python PIP package
Plugins for EDR/XDR platforms	Security Automation and Integration	
Data Ingestion via E-Mail	Security Automation and Integration	
SOAR plugins	Security Automation and Integration	e.g. Splunk SOAR, Palo Alto Cortex
SIEM system integration	Security Automation and Integration	e.g. via CEF syslog
TIP integration	Threat Intelligence Sharing and Management	e.g. MISP
YARA with customizable ruleset	Threat Intelligence Sharing and Management	
Threat Intelligence Reputation Lookup	Threat Intelligence Sharing and Management	
Automated E-Mail Notification	Security Automation and Integration	

KPI: Security, Deployment and Maintenance

This Key Performance Indicator evaluates the security posture, ease of deployment, and efficiency of maintenance for a sandbox solution. It assesses how securely the solution can be deployed, the simplicity of the setup process, and the operational resources required for ongoing updates, system hardening, and compliance with security best practices. The goal is to ensure minimal administrative overhead while maintaining strong security standards throughout the solution's lifecycle.

Feature Set

Feature	Category	Comment
Network segregation by design	Network Security	Proper isolation of the detonation environment from internal networks / DMZ support

Feature	Category	Comment
System Hardening & Continuous Updates	System Security	E.g. CIS compliance, automated patch management
RBAC (Role Based Access Control Lists)	System Security	Principle of Least Privilege (POLP)
Configurable Data Retention	System Security	Flexible data retention policies to meet regulatory and business requirement.
Custom Password Policies	System Security	Ability to enforce organization-specific password complexity, expiration, and rotation policies to enhance security
Audit Logs	Security Monitoring and Logging	Comprehensive audit trails for security monitoring and compliance
Certifications	Compliance and Certification	ISO 27001, SOC 2, NIST
AI Risk Assessment & Compliance	Compliance and Certification	Ensures data integrity and transparency in AI systems with mechanisms for explainability, auditability, and compliance with data minimization principles
Data redundancy / Backup mechanisms	Data Management and Security	Mitigate data loss in case of hardware or software failures
Regional Data Centers	Data Management and Security	Support for data localization and compliance with regional data residency requirements
Data Governance & Explainability for AI	Data Management and Security	

KPI: LLM-as-Sample Dynamic Analysis

This indicator assesses the sandbox's ability to ingest a Large Language Model (LLM) as a sample and produce a structured behavioral report. Where a traditional sandbox detonates a binary and observes syscalls, an LLM-aware sandbox detonates prompts and observes generations. The KPI evaluates whether the sandbox can drive the LLM under reproducible conditions, observe and classify responses, and present results in the same operational format as traditional sample analysis.

Scoring follows the framework's existing 0 / 3 / 5 / 10 scale (Not available / Limited / Comprehensive / Exceptional) and the same weighted-sum-and-normalize formula. No changes to the framework structure or grading bands are introduced by this KPI.

Scope

In Scope:

- **Submission of an LLM** (hosted endpoint, local model, or model file) to the sandbox as a sample.
- **Execution of a structured prompt battery** against the LLM under controlled, reproducible conditions.

- **Capture and classification** of the LLM's responses into observable behavior categories.
- **Reporting** of LLM-sample behavior alongside, and in the same format as, traditional sample analysis.

Out of Scope:

- **Measuring overall LLM safety, alignment, helpfulness, capability, or factual accuracy.** The AMTSO AI Working Group is the right home for those questions.
- **Adversarial robustness testing, jailbreak research, or red-team-grade evaluation.**
- **Prompt corpus authoring, content, or governance** — deferred to the AMTSO AI Working Group.
- **LLM-powered features inside the sandbox itself** (e.g., LLM-assisted triage). Separate question, separate amendment.

Feature Set (Indicative)

Feature	Category	Comment
LLM endpoint ingestion	Sample Intake	Accept hosted API endpoints (e.g., OpenAI-compatible, Anthropic-compatible) as a sample type, with credential handling and rate-limit awareness.
Local model ingestion	Sample Intake	Accept model files (GGUF, safetensors, ONNX, etc.) and execute them locally within the sandbox boundary, without phoning home.
System prompt and configuration capture	Sample Intake	Record the system prompt, sampling parameters (temperature, top-p), tool/function configuration, and safety filter state under which the LLM was tested.
Reproducible sampling	Execution Control	Submit each prompt N times (operator-configurable) and record full response distributions, not single-shot outputs.
Multi-turn detonation	Execution Control	Drive multi-turn conversations from a scripted scenario, including tool-use call/response loops where the LLM has tool access.
Prompt corpus support	Stimulus	Accept and execute a structured prompt corpus supplied by the operator. Corpus content and governance are out of scope for this KPI; the AI WG is the natural home.
Response classification	Behavioral Analysis	Apply a documented classification scheme to each response. Automated classifiers are permitted; the method should be disclosed.
Tool-call and side-effect capture	Behavioral Analysis	When the LLM has tool access (web, code execution, file I/O, shell), record all attempted tool calls as first-class behavioral events, comparable to syscall capture for binaries.

Feature	Category	Comment
Output content inspection	Behavioral Analysis	Inspect generated content even when wrapped in fictional, code-comment, verse, or hypothetical framing. Wrapper format alone should not gate the classification.
Run reproducibility metadata	Reporting	Record corpus version, model identifier and version, sampling parameters, system prompt hash, and run timestamp in every report.
Per-category reporting	Reporting	Surface results in a structured, per-category form rather than as a single composite score.

Executing the Framework

To execute the evaluation framework effectively, we propose the inclusion of a sample set of benchmark files that encompass a diverse range of evasion techniques and behaviors, ensuring a rigorous evaluation of sandbox capabilities across most key performance indicators. This will provide a single source of truth and standardized method for assessing sandbox solutions and offering a clear visualization of their performance (ideally, in a radar chart). This framework empowers organizations to make informed decisions when selecting and configuring sandbox systems.

Suggested Weight Profiles

We also propose standard weight configurations for distinguished use cases to ensure the evaluation is performed in alignment with the end user's needs. We believe, the following use cases are most distinguished:

Note (v1.1): The five profiles below were defined for the original six KPIs and are reproduced unchanged. The new KPI "LLM-as-Sample Dynamic Analysis" is deliberately omitted from these profiles. Operators applying the framework to LLM-relevant use cases may set their own weight for the seventh KPI; a reasonable interim default is to weight it only where it applies to the specific evaluation.

Use Case #1: Large-Scale Processing Focusing on Detection

Proposed Weights:

Key Performance Indicator	Score (0..10)	Weight
Analysis Capability	S1	High
Anti-Evasion Technology	S2	Medium
Speed/Throughput/Scale	S3	High
Reporting and Threat Research	S4	Low
Integrations and Automation	S5	Low

Key Performance Indicator	Score (0..10)	Weight
Security, Deployment and Maintenance	S6	Medium

Use Case #2: Malware/Phishing Triage and SOC Automation

Proposed Weights:

Key Performance Indicator	Score (0..10)	Weight
Detection Analysis Capability	S1	Medium
Anti-Evasion Technology	S2	High
Speed/Throughput/Scale	S3	Low
Reporting	S4	High
Integrations and Automation	S5	High
Security, Deployment and Maintenance	S6	Medium

Use Case #3: Focus on Zero-Day Detection

Proposed Weights:

Key Performance Indicator	Score (0..10)	Weight
Analysis Capability	S1	High
Anti-Evasion Technology	S2	High
Speed/Throughput/Scale	S3	Low
Reporting and Threat Research	S4	High
Integrations and Automation	S5	Medium
Security, Deployment and Maintenance	S6	Medium

Use Case #4: Air-Gapped Critical Infrastructure Security

Proposed Weights:

Key Performance Indicator	Score (0..10)	Weight
Analysis Capability	S1	Medium
Anti-Evasion Technology	S2	Medium
Speed/Throughput/Scale	S3	Medium
Reporting and Threat Research	S4	Medium
Integrations and Automation	S5	Low
Security, Deployment and Maintenance	S6	High

Use Case #5: Threat Intel Generation

Proposed Weights:

Key Performance Indicator	Score (0..10)	Weight
Detection Analysis Capability	S1	High
Anti-Evasion Technology	S2	High
Speed/Throughput/Scale	S3	Low
Reporting	S4	High
Integrations and Automation	S5	Medium
Security, Deployment and Maintenance	S6	Medium

To calculate the final score, please fill in the score of your sandbox solution and refer to Evaluation Score Formula.

Suggestions on Transparency

To enhance transparency across different use cases, consider the following recommendations:

- **Publish Detection Efficacy:** Regularly share performance metrics, including false positive/negative rates, across diverse scenarios like targeted attacks, evasive malware, and APT detection—not just mass malware detection.
- **Publish Weighted Scores and Feature Sets Used:** Clearly outline the criteria, scoring methodology, and feature sets applied in performance evaluations to provide stakeholders with full visibility into the assessment process.
- **Publish Mapping to the MITRE ATT&CK Framework:** Align detection results with the MITRE ATT&CK framework to illustrate coverage across various tactics and techniques beyond basic malware detection.
- **Showcase Real-World Use Cases:** Provide examples from incident response, SOC alert triaging, and forensic investigations to demonstrate the sandbox's efficacy in complex, real-world scenarios.
- **Transparency in Update and Versioning History:** Make information about sandbox updates, detection engine changes, and version improvements publicly available to build trust in continuous improvement efforts.
- **Document Known Limitations:** Clearly communicate known blind spots or limitations (e.g., specific evasion techniques, file types) to set realistic expectations for users.

Open Source Benchmark Tools

Please find a list of open-source sandbox benchmark tools that may be used for additional sandbox assessments below:

- https://github.com/a0rtega/pafish
- https://github.com/joesecurity/pafishmacro
- https://github.com/LordNoteworthy/al-khaser
- https://github.com/hfiref0x/VMDE

Disclaimer: While these tools are widely used for sandbox benchmarking, they are known to occasionally produce false detections. Therefore, the accuracy and reliability of the results

cannot be fully guaranteed. It's recommended to complement these tools with additional testing methodologies for comprehensive assessments.

Example Evaluation: Sandbox Vendor

	Analysis Capability	Anti-Evasion Technology	Speed/ Throughput/ Scale	Reporting and Threat Hunting	Integrations and Automation	Security, Deployment and Maintenance	Total score
Score	6.85	8.2	9.5	6.56	7.11	5	
Weight	10	10	10	3	3	5	
Weighted score	68.5	82	95	19.68	21.33	25	
Max score	100	100	100	30	30	50	
Final score							72%

Grading	Threshold
Very good	80
Good	65
Average	50
Poor	35
Very poor	20

Conclusion

In conclusion, this testing framework addresses the pressing need for a comprehensive, standardized approach to evaluating sandbox systems on a use-case basis. By assessing key performance indicators such as speed, compute cost, detection, and anti-evasion, organizations can confidently select the sandbox solution that aligns with their security requirements, ultimately bolstering their defense against evolving cyber threats.

With this guideline, we hope to encourage both sandbox vendors and end users to conclude that "not every sandbox is the same" and different sandboxes serve different use cases.

This document was adopted by AMTSO on 2025-03-26. v1.1, adding the LLM-as-Sample Dynamic Analysis KPI, is proposed as of 2026-06-18; the adoption date will be filled in once the amendment is approved by the membership.