The cybersecurity industry's testing standard community

Sandbox Evaluation Framework

Lead Author:
Jan Miller
CTO, Threat Analysis,
OPSWAT Inc.

Version: 0.1
Date: 2024-09-24
TLP: Amber

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.

Table of Contents

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 first line of defense, analyzing potentially malicious software in a controlled environment before it 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.

Factor	Real-Time Dynamic Analysis	Emulation-Based Sandbox	QEMU-Based Sandbox	VM-Based Traditional Sandbox
Execution Speed	Near real-time (milliseconds to seconds)	Extremely fast (milliseconds)	Slower (seconds to minutes)	Slow (minutes)
Execution Environment	Lightweight real-world simulation	High-level emulation of specific components	Full system emulation, including hardware and OS	Full virtual machine with complete OS stack
Resource Consumption	Medium—optimized but runs more of the code	Very low—emulates only critical parts	High—requires full emulation of hardware and OS	Very high—requires full OS, application, and hardware stack
Depth of Analysis	Detailed but optimized for speed	Focuses on critical malware behaviors	Comprehensive—includes full system behavior	Full behavioral and system interaction analysis
Use Case	Environments requiring fast decision-making (e.g., gateways)	High-throughput malware detection with low overhead	Malware analysis requiring detailed behavioral analysis	In-depth forensic analysis, complex malware detection

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:

Key Performance Indicator	**Score (0..10)**	**Weight**
Detection Capability	s1	w1
Anti-Evasion Technology	s2	w2
Compute Cost	s3	w3
Speed/Throughput	s4	w4
Deployment and Scalability	s5	w5
Reporting and Threat Hunting	s6	w6
Integrations and Automation	s7	w7
Maintenance and Security	s8	w8

Each of these indicators address 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 the 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 a forensic 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 supported
3	Below average support
5	Partially supported
7	Above average support
10	Fully supported

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: Detection 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	Vendor Score	Comment
Support Windows	File Type Support	TBD	PE, DLL, Powershell, VBS, JScript, Office (all flavors, including .DOC, .DOCX, XLM 4.0, .XLS, .PPT, .PUB, etc.), PDF
Support Linux	File Type Support	TBD	ELF, Bash, Lua, Python
Support Android	File Type Support	TBD	APK
Support OSX	File Type Support	TBD	MACH-O
Support Very Large Files	File Support	TBD	Very Large Files – Bigger than 1 GB
WMI Query Capture	Behavioral Analysis	TBD	
Memory Dumps	Behavioral Analysis	TBD	
Screenshots	Behavioral Analysis	TBD	
Injection Detection	Behavioral Analysis	TBD	e.g. APC, Process Hollowing, Atom Bombing
Interactivity	Behavioral Analysis	TBD	e.g. to bypass installers
BIOS / Reboot analysis	Behavioral Analysis	TBD	Bootkits, Supply Chain
Network Capture	Network and Communication Analysis	TBD	
SSL Decrypt via TLS key interception / MITM	Network and Communication Analysis	TBD	e.g. C&C protocol analysis
DNS Spoofing	Network and Communication Analysis	TBD	Increase extraction of potential C&C network IOCs
Config Extraction	Content and Configuration Analysis	TBD	
Generic Unpacking / Dynamic Payload Extraction	Content and Configuration Analysis	TBD	
Binary disassembly	Behavioral Analysis	TBD	
Fuzzy hashes	Content and Configuration Analysis	TBD	

Certificate validation	Content and Configuration Analysis	TBD	
Recursive processing of extracted files	Behavioral Analysis	TBD	
Compiler/RICH Header Parsing	Content and Configuration Analysis	TBD	

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	Vendor Score	Comment
Sleep Reduction	Evasion Technique	TBD	Avoid long sleeps, loops
MAC address spoofing	Evasion Technique	TBD	VMWare, VirtualBox, Qemu have default MAC address values
CPUID spoofing	Evasion Technique	TBD	Instruction level VM detection
RDTSC / GetTickCount spoofing	Evasion Technique	TBD	Performance counter used for execution time measurement
Mouse/Keyboard simulation	Evasion Technique	TBD	Human simulation, execution trigger (e.g. via dialog box interaction)
Registry Key Spoofing	Evasion Technique	TBD	Hide registry artefacts that reveal presence of a VM / agent
Advanced Anti-Evasion	Evasion Technique	TBD	E.g. Thermal temperature, Firmware tables
Wear-and-tear fuzzy images	Custom Images	TBD	Avoid off-the-shelf vanilla execution environment
Configurable Application Stack	Custom Images	TBD	Enable mimicking a golden execution environment (e.g. for exploit trigger)
Customizable system environment (e.g. System locale)	Custom Images	TBD	Enable mimicking a golden execution environment
Network simulation	Simulation and Manipulation	TBD	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	TBD	Usage of OS binaries to delay execution

KPI: Compute Cost

Given the importance of resource-efficient cybersecurity solutions, this indicator measures the computational resources, such as CPU and memory usage, required to execute and maintain a sandbox system during the analysis of potentially malicious files.

Feature Set

Feature	Category	Vendor Score	Comment
Total Memory Usage	Resource Consumption	TBD	
Total vCPU Hours	Resource Consumption	TBD	
Total Disc Usage	Resource Consumption	TBD	

We recommend running our sample set on an Amazon EC2 instance (m5a.xlarge, c5a.2xlarge, or c5a.4xlarge) and measuring the total memory, vCPU and disc usage. For memory and disc I/O metrics, a service such as CloudWatch needs to be configured.

Emulation-based sandbox systems typically require 10x fewer resources than traditional VM-based sandboxes due to their ability to focus on critical malware behavior without fully simulating the entire OS stack. Additionally, real-time dynamic analysis technologies, designed to minimize resource usage while providing immediate results, can use 100x fewer resources than traditional VM-based sandboxes, with analysis times measured in milliseconds. Therefore, in the absence of a feature set benchmark, we recommend using the following scoring: '10' for real-time dynamic analysis and emulation-based sandboxes, '7' for hybrid sandboxes with emulation-based dynamic analysis, '5' for QEMU Linux-based sandboxes (due to KVM integration), and '3' for VM-based sandboxes.

KPI: Speed/Throughput

This indicator focuses on the throughput and response time of a sandbox solution when analyzing potentially malicious files. It assesses how quickly a sandbox can process incoming samples without compromising analysis accuracy.

Feature Set

Feature	Category	Vendor Score	Comment
Average Processing Time for Small Size Sample Set	Processing Time Metrics	TBD	
Average Processing Time for Large Size Sample Set	Processing Time Metrics	TBD	
Total Processing Time for Document Set (N=1000)	Processing Time Metrics	TBD	
Total Processing Time for Executable Set (N=1000)	Processing Time Metrics	TBD	
Max Throughput per Virtual Machine (Analysis Node)	Throughput and Parallel Processing	TBD	
Max Parallel Processing Tasks	Throughput and Parallel Processing	TBD	

KPI: Deployment and Scalability

As organizations grow, the ability of a sandbox solution to scale seamlessly becomes critical. This indicator evaluates the system's scalability, ensuring it can handle increased workloads and adapt to changing operational requirements.

Feature Set

Feature	Category	Vendor Score	Comment
Cloud native	Deployment and Infrastructure	TBD	Not, if nested virtualization is required
Deployable as a container	Deployment and Infrastructure	TBD	E.g. Kubernetes Cluster
Can run in air-gapped environments	Deployment and Infrastructure	TBD	
Ensures full privacy	Deployment and Infrastructure	TBD	i.e., no data is sent to the vendor or any third-party
Auto-Scaling Mechanisms	Scalability and Availability	TBD	Dynamic workload (scaling actions, trigger metrics)
High availability	Scalability and Availability	TBD	Single point of failure / Ability to maintain service even during failures, Uptime monitors

KPI: Reporting and Threat Research

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	Vendor Support	Comment
Single-file PDF	File Formats	TBD	PDF-A support is a bonus
Single-file HTML	File Formats	TBD	
MAEC	Security Standards and Frameworks	TBD	
STIX	Security Standards and Frameworks	TBD	
MITRE ATT&CK mapping	Security Standards and Frameworks	TBD	
JSON/XML Export	Data Export and Integration	TBD	
Automated E-Mail Notifications	Data Export and Integration	TBD	
Advanced Report Search	Threat Hunting	TBD	e.g. Find reports sharing similar threat indicators or characteristics
Threat Prevalence Data	Threat Hunting	TBD	
Fuzzy Hashes	Threat Hunting	TBD	Similar sample correlation / Unknown threat detection

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	Vendor Score	Comment
Web API with automated documentation (e.g. OpenAPI)	Developer Tools for APIs and SDKs	TBD	
SDK with CLI	Developer Tools for APIs and SDKs	TBD	e.g. Python PIP package
SOAR plugins	Security Automation and Integration	TBD	e.g. Splunk SOAR, Palo Alto Cortex
SIEM system integration	Security Automation and Integration	TBD	e.g. via CEF syslog
MISP integration	Threat Intelligence Sharing and Management	TBD	
YARA with customizable ruleset	Threat Intelligence Sharing and Management	TBD	
MISP Galaxy / Automated tagging	Threat Intelligence Sharing and Management	TBD	
Threat Intelligence Reputation Lookup	Threat Intelligence Sharing and Management	TBD	
Automated E-Mail Notification	Security Automation and Integration	TBD	

KPI: Security and Maintenance

The ease of deploying and maintaining a sandbox solution significantly impacts operational efficiency. This indicator assesses the simplicity and efficiency of deploying the solution and the resources required for ongoing maintenance.

Feature Set

Feature	Category	Vendor Score	Comment
Network segregation by design	Network Security	TBD	Proper isolation of the detonation environment from internal networks / DMZ support
System Hardening & Continuous Updates	System Security	TBD	E.g. CIS compliance, automated patch management
Access Control Lists	System Security	TBD	Principle of Least Privilege (POLP)
Audit Logs	Security Monitoring and Logging	TBD	Audit trails
Certifications (ISO 27001, GDPR, NIST)	Compliance and Certification	TBD	
Data redundancy / Backup mechanisms	Data Management and Security	TBD	Mitigate data loss in case of hardware/software failures

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:

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

Proposed Weights:

Key Performance Indicator	**Score (0..10)**	**Weight**
Detection Capability	S1	10
Anti-Evasion Technology	S2	5
Compute Cost	S3	7
Speed/Throughput	S4	10
Deployment and Scalability	S5	10
Reporting and Threat Hunting	S6	3
Integrations and Automation	S7	3
Maintenance and Security	S8	7

Use Case #2: Small-Scale Processing focused on Forensic Analysis

Proposed Weights:

Key Performance Indicator	**Score (0..10)**	**Weight**
Detection Capability	S1	7
Anti-Evasion Technology	S2	10
Compute Cost	S3	3
Speed/Throughput	S4	3
Deployment and Scalability	S5	3
Reporting and Threat Hunting	S6	10
Integrations and Automation	S7	7
Maintenance and Security	S8	7

Use Case #3: Focus on Zero-Day Detection

Proposed Weights:

Key Performance Indicator	**Score (0..10)**	**Weight**
Detection Capability	S1	10
Anti-Evasion Technology	S2	10
Compute Cost	S3	3
Speed/Throughput	S4	5
Deployment and Scalability	S5	5
Reporting and Threat Hunting	S6	10
Integrations and Automation	S7	7
Maintenance and Security	S8	10

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

Open Source Benchmark Tools

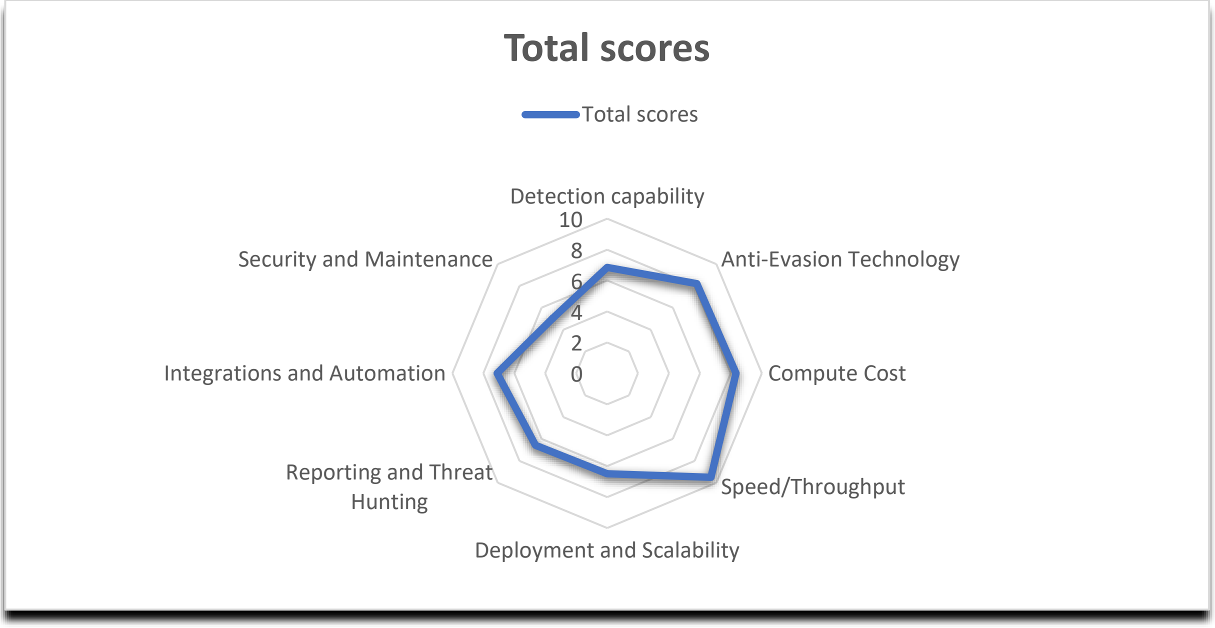
Please find a list of open-source sandbox benchmark tools that may be used for additional sandbox assessments below:

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

Example Evaluation: OPSWAT Filescan Sandbox

	Detection capability	Anti-Evasion Technology	Compute Cost	Speed/ Throughput	Deployment and Scalability	Reporting and Threat Hunting	Integrations and Automation	Security and Maintenance
Total score	6.85	8.2	8.33	9.5	6.5	6.56	7.11	5
Weight	10	10	7	10	10	3	3	5
Weighted score	68.5	82	58.31	95	65	19.68	21.33	25
Max score	100	100	70	100	100	30	30	50

Final score	Grading	
75	80	Very good
	65	Good
	50	Average
	35	Poor
	20	Very poor



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.